The Report Committee for Arthur Go Ishiguro
Certifies that this is the approved version of the following report:

Scheduling and Resource Allocation for Mobile Broadband Networks

APPROVED BY

SUPERVISING COMMITTEE:

Jeffrey G. Andrews, Supervisor

Gustavo de Veciana, Co-Supervisor

Scheduling and Resource Allocation for Mobile Broadband Networks

by

Arthur Go Ishiguro, B.S.E.E.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2014

Scheduling and Resource Allocation for Mobile Broadband Networks

Arthur Go Ishiguro, M.S.E.
The University of Texas at Austin, 2014

Supervisors: Jeffrey G. Andrews
Gustavo de Veciana

Unlike traditional cellular networks, where voice calls dominate the network traffic, modern mobile traffic is created by of a mixture of both voice and broadband data services. The heterogeneous mixture of voice and data services in mobile broadband networks includes voice calls, web browsing, file transfers, video streaming, and social media applications. Consequently, network planning and radio resource management strategies must be aware of the quality of experience perceived by the users using various types of applications. In this report, we explore the traffic characteristics, scheduling and resource allocation strategies, and user experience models in mobile broadband networks.

Table of Contents

List of Figures

Chapter 1

Introduction

Recent technological advances have allowed mobile users to access various broadband services such as video streaming, web browsing, and social media. The amount of data traffic in mobile broadband (MBB) networks has exploded in the past few years, and is expected to continue to grow in the future. In the past year alone, Ericsson reports an 80% increase in aggregate data traffic volume [1]. A recent technical report by Sandvine [2] also shows that the majority of the mobile downlink traffic is consumed by video streaming and web browsing services. As mobile data traffic continues to grow, understanding characteristics of both voice and data mobile traffic is crucial for the design of future communication systems.

Because of the variety in the application types, mobile users will also require different quality of service (QoS) requirements. To meet these requirements, the 3rd Generation Partnership Project (3GPP) introduced the Long Term Evolution (LTE) architecture with aggressive performance requirements [3]. LTE is a packet-switched internet protocol (IP) architecture that allows mobile devices to access the internet through its core network. In particular, LTE specifies nine classes of mobile traffic. Each class is labeled as guaranteed

bit rate (GBR) or non-guaranteed bit rate (non-GBR), and is associated with a service priority, delay requirement, and packet loss rate requirements [4]. GBR traffic, such as real-time voice and gaming, are allocated specific bandwidth in order to meet their bit rate and latency requirements. Non-GBR traffic, like file transfers, have a more relaxed delay or bit rate requirements. Because MBB networks consist of both GBR and non-GBR traffic, network operators must understand appropriate measures of user experience that are application-specific.

Network planning and optimization for future MBB networks must consider these heterogenous application types used in mobile devices. Unlike cellular networks consisting of only voice traffic, optimization strategies for non-real-time data traffic are also not well-understood. The purpose of this report is to investigate traffic characteristics, optimization strategies, and user experience measures for the various traffic classes of MBB networks. Chapter 2 describes traffic models that have been considered for voice and data services. Chapter 3 discusses and compares multi-user scheduling and resource allocation optimization strategies. In Chapter 4, quality of experience models for MBB applications are introduced. Finally, in Chapter 5, we consider a context-aware scheduling strategy for non-real-time web browsing traffic.

Chapter 2

Traffic Models for MBB networks

In system-level simulations, the characteristics of incoming packet traffic must be carefully considered by implementing appropriate models. Traffic models can follow either the full-buffer or the finite-buffer model. In the full-buffer model, it is assumed that data is always available for transmission. In contrast, the finite-buffer model assumes users that arrive and depart from the network after a particular payload is delivered, and consequently are not always competing for resources. The performance of schedulers has been shown to be largely affected by the way mobile traffic is modeled, since scheduling decisions depend on the number of active users [5]. For MBB networks, it is important to consider realistic traffic models for each service type to accurately perform simulations.

Several implementations of traffic models have been introduced by standardization groups. For instance, 3GPP has defined both full-buffer and finite-buffer models of file transfer protocol (FTP) best-effort traffic [6]. 3GPP2 [7] defines detailed models of Voice-over-IP (VoIP), FTP, web browsing, and video streaming traffic for CDMA2000 evaluations. WiMAX in [8] outlines several traffic models including internet gaming, VoIP, video conferencing, push-to-

talk, audio/video streaming, mobile broadcast, instant messaging, web browsing, email, telemetry (machine-to-machine), FTP, peer-to-peer (P2P), and virtual peer networks. In [9], NGMN proposes traffic models for FTP, web browsing, VoIP, video streaming, and interactive gaming. In addition to the models presented by NGMN, video telephony and email traffic is presented in [10] for evaluation for IEEE 802.16m.

Recent trends of MBB traffic show that an increasing amount of non-real-time applications such as web browsing and streaming traffic are starting to dominate the network data traffic [2]. In this section, we focus on describing the downlink models of key service classes in current MBB networks: VoIP, file transfers, web browsing, and video streaming.

2.1 VoIP Traffic

A two-state ON/OFF Markov chain consisting of talking and silent states is used to model an ongoing VoIP conversation. This section highlights the main idea of the VoIP traffic model. A more detailed description of the model is available in [11].

2.1.1 VoIP Traffic Model

In a given time instant, a VoIP user is either in the talking or silent state. The user switches from the talking to the silent state with probability β, and from silent to talking state with probability α. The state transition is modeled by a two-state Markov chain as illustrated in Figure 2.1, and is

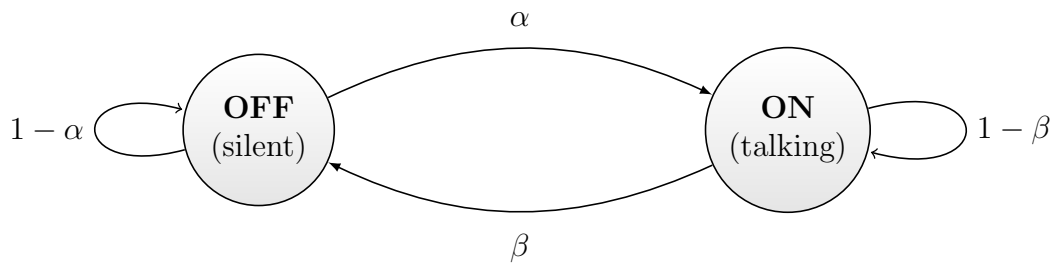


Figure 2.1: VoIP traffic model using a two-state Markov chain.

updated at the encoder rate.

In this Markov model, the mean talking time τ_T and silent times τ_S in the number of frames are given by

$$\mathbb{E}[\tau_T] = 1/\beta \qquad \mathbb{E}[\tau_S] = 1/\alpha. \tag{2.1}$$

The voice activity factor (VAF) v represents the fraction of users that are in the talking state. The parameters v, α, and β can be written in the form

$$v = \frac{\alpha}{\alpha + \beta} \qquad \alpha = \frac{v}{1-v}\beta. \tag{2.2}$$

Using the above expression, we can design the parameters α and β depending on the desired VAF, mean talking state, or the mean silent state.

2.2 File Transfers

File transfer over the internet is performed over a network protocol called the file transfer protocol (FTP). In general, the downlink file transfer traffic is modeled as a sequence of downloads modeled as Poisson arrivals. Each file transfer is associated with a random file size S and a reading time

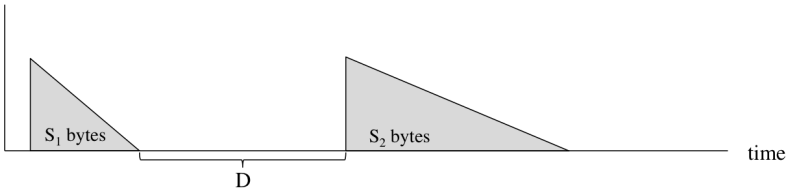


Figure 2.2: File transfer traffic model.

D. The reading time represents the time until the next file transfer occurs, as illustrated in Figure 2.2. In both 3GPP [6] and 3GPP2 methodologies, D is a random variable with an exponential distribution.

The underlying transport protocol is the transmission control protocol (TCP). In this model, the TCP connection begins with a call setup. During the subsequent file transfer, packets are divided and are downloaded through a series of conversations between the user and the server through a slow-start congestion control technique [12]. Once the file is downloaded, the connection is terminated with a call release. In this section, we discuss the details in the TCP connection using the slow-start congestion control model.

2.2.1 TCP Connection Model

The TCP connection is based on a sequence of handshake conversations between the user, base station router, and the file server. The following sequence of events occur in every file transfer download:

1. **Call Set-up**

   580 ms after its arrival, a user sends a SYNC signal of 47 bytes, and waits for an acknowledgement (ACK) signal from the server. This includes the

40 bytes of TCP/IP header and 7 bytes of point-to-point protocol (PPP) overhead.

2. **TCP File Transfer**

   Once the user receives the ACK signal in the call set-up, the TCP file transfer begins. The user sets the ACK flag in the first TCP packet.

3. **Call Release**

   In the last TCP segment, the user sets the FIN flag. The user waits for an ACK signal from the server, and the file transfer is completed.

2.2.2 TCP File Transfer Model

The data transfer between the user and the server is modeled by a sequence of conversations between the user and the server. The server sends the TCP data packet, and the user responds by sending an ACK signal. Once the ACK signal is received, the server sends the next two data packets simultaneously (or one if no more packets remain). Figure 2.3 illustrates the handshaking conversation in a file transfer. The server initially transmits one data packet, but the number of packets that are transmitted grows exponentially as the file transfer continues. This number of transmitted data packets is referred to as the TCP congestion window [12].

The total time it takes for the data to reach the user from the server is referred to as the round-trip time, given by $\tau_{RT} = \tau_C + \tau_L$. The first variable in the expression, τ_C, is the sum of the time for an ACK signal to reach the

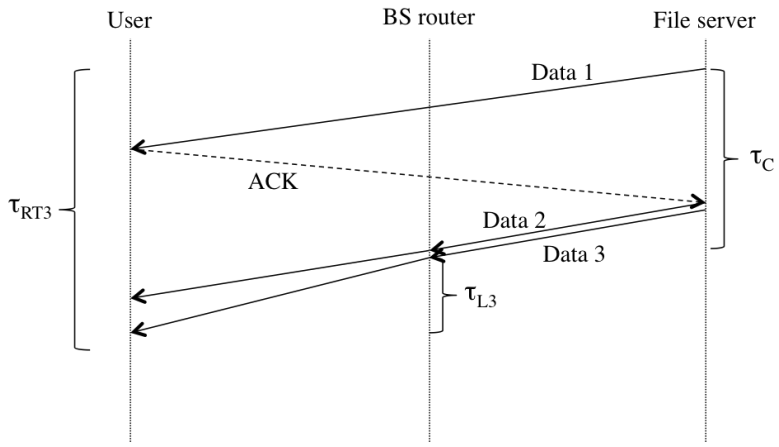


Figure 2.3: Handshake modeling of the slow-start TCP file transfer.

server from the user, and the data packet to reach the base station router from the server. The access link transmission time, τ_L, is the time it takes for the data packet to reach the user from the base station router. In the 3GPP2 model [7], τ_C is an exponential random variable with a mean of 50 ms, and τ_L is determined by the available access link throughput.

In a file transfer, the entire file is divided into packets of various sizes. In a single file transfer, the maximum transmission unit (MTU) of the each packet is chosen. According to [10], 76% of the packets have an MTU of 1500 bytes, and 576 bytes for the remaining 24%. The data packet includes the TCP/IP header of 40 bytes.

2.3 HTTP Web Browsing

One of the most accepted models of web browsing traffic was proposed by Choi and Limb based on a traffic study of a campus backbone network [13].

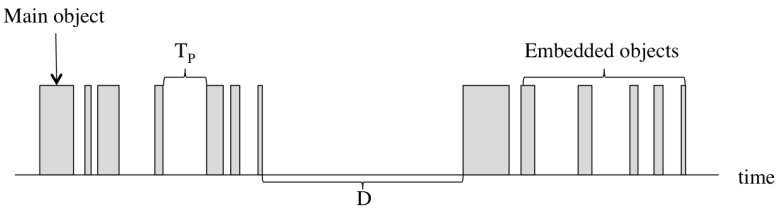


Figure 2.4: HTTP Web Browsing traffic model.

Their model reflects a behavioral characterization of interactions between a web server and a user through a series of download requests. The traffic characteristics and parameters in this model were obtained in 1999, however, and thus may be outdated to model modern wireless web browsing traffic. In this section, we discuss the basic model and recent studies on the accuracy of the parameter distributions.

2.3.1 Web Browsing Traffic Model

Web traffic consists of a sequence of web download requests referred to as packet calls, as illustrated in Figure 2.4. A packet call consists of a single main object of file size S_M and N_E embedded objects (such as images and audio files) of size S_E. Within a packet call, each object is separated by its parsing time T_P seconds. The last embedded object of the packet call is followed by a reading time of D seconds, where no data is transmitted until the next packet call. The reading time represents the user viewing the web content, until the next web download is requested.

2.3.2 HTTP/1.0 and HTTP/1.1 models

Because the underlying protocol for web browsing is TCP, a similar connection model to the FTP model is considered in the 3GPP2 traffic model [7]. This includes the TCP connection set-up, file transfer using slow-start, and connection release discussed in section 2.2.1. The same τ_C variable, used to model the sum of the ACK signal transmission and the data transfer from the server to the BS router, should be added to the parsing time, T_P.

In the 3GPP2 model of HTTP web browsing traffic [7], both HTTP/1.0 and HTTP/1.1 persistent versions of the TCP connection model are defined. In the persistent model, a single TCP connection is used to transfer objects within a web page [14]. HTTP/1.0 model, on the other hand, establishes a new TCP connection for each object. According to [15], about 50% of the internet traffic uses the HTTP/1.1 persistent model. A detailed description of the packet trace modeling can be found in [7].

2.4 Video Streaming

Delivery of video files over a wireless network go through several communication layers, and is difficult to accurately model through simulations. As video streaming protocols are still developing, most simulations for video streaming traffic are based on real video traces. Several video trace files of varying qualities are available in [16]. The main drawback of trace-based simulation models is that they are based on specific instances of video traffic. In this paper, we consider a more general model of streaming traffic characteris-

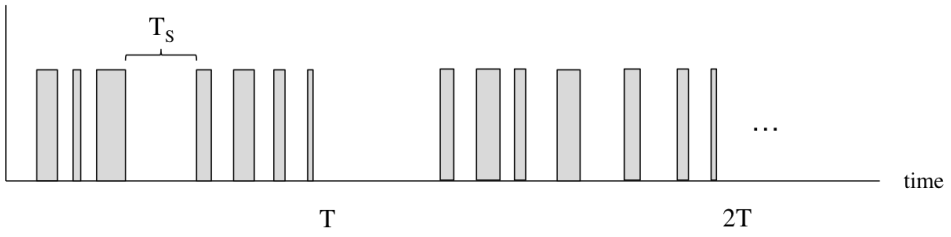


Figure 2.5: Video streaming model.

tics adopted by NGMN [9].

The NGMN video streaming traffic model [9] is based on a sequence of video frames that are transmitted at a constant rate of T seconds throughout the entire simulation time. This rate T is determined by the frame rate of the video. As illustrated in Figure 2.5, each video frame is composed of a constant N_S slices with packet size S_S. Each slice has an inter-arrival time of T_S seconds, which represents the encoding delay between each packet. This model also defines a de-jitter buffer window of T_B seconds, which is used to deliver continued display of the video to the user through the simulation time frame. The buffer is emptied at the source video data rate, and is filled as packets arrive to the user. If the buffer runs empty, then the user is no longer able to view the video. Although not directly applied to the traffic model, the de-jitter buffer can be used to measure the QoS of a streaming user.

2.5 Conclusion

In this section, we explored statistical traffic models for VoIP, FTP download, HTTP web download, and video streaming traffic. The traffic mod-

els described in this section have various characteristics in the time-domain. Real-time traffic such as VoIP may be appropriately modeled by a full-buffer model with continuous streams of data. However, non-real-time traffic such as FTP and HTTP downloads may be more bursty. For future designs for MBB networks, application-specific traffic models such as those described in this report is necessary to perform realistic simulations.

Chapter 3

Scheduling and Resource Allocation Algorithms

Radio resource management in LTE downlink is based on Orthogonal Frequency Division Multiple Access (OFDMA), where multiple users are assigned resource blocks (RBs) in both time and frequency. Under OFDMA, scheduling and resource allocation can be done flexibly on a per-RB granularity. Because wireless communication channels exhibit fluctuations in both time and frequency, users experience different channel conditions under a particular RB. Consequently, opportunistically assigning RBs to users with good channel conditions is one way to optimize network performance. Due to the stringent delay requirements of real-time applications, the buffer queue status is an indicator of user satisfaction. To properly serve users with heterogeneous QoS requirements in MBB networks, the scheduler must be both channel and queue-aware. When multiple users are present in the network, RBs can be assigned to a single user depending on the goal of the scheduling algorithm. In this report, we compare the performance of three types of scheduling algorithms: Maximum Throughput, Proportionally Fair, and Modified Largest Weighted Delay First (M-LWDF).

3.1 Channel and Queue-aware Scheduling Algorithms

In the scheduling algorithms considered in this report, a marginal utility $U_n(t)$ is defined and computed for each scheduling resource block. This utility is a measure of the value of assigning a particular resource block to a user for data transmission with respect to the performance goal of the scheduler. For each scheduling instance, the user with the highest utility is chosen to be assigned a resource block.

The *Maximum Throughput* (MT) scheduler is channel-aware scheduler that selects users with the best achievable throughput for a given resource block. In this algorithm, the marginal utility is given by

$$U_n(t) = R_n(t), \tag{3.1}$$

where $R_n(t)$ is the instantaneous achievable rate of user n. Although the MT scheduler maximizes the system throughput, fairness is sacrificed because users with poor channel conditions are never selected. Consequently, the resource assignment becomes skewed toward a group of users that experience good channels.

To combat potential unfairness in resource block assignment, the *Proportionally Fair* (PF) algorithm has been proposed [17]. The PF algorithm was originally proposed as solution for congestion control of elastic internet traffic. The PF algorithm incorporates the past throughput of the user $\bar{R}_n(t-1)$ in the marginal utility, given by

$$U_n(t) = \frac{R_n(t)}{\bar{R}_n(t-1)}. \tag{3.2}$$

In this way, users that have been less frequency been scheduled in the past are weighted more heavily than those with better channel conditions. This results in a more fair opportunity for resource block assignment across the users regardless of the fluctuation in the channel gains.

The aforementioned channel-aware algorithms are not applicable for users with real-time applications with additional QoS requirements. To properly serve such users, QoS-aware algorithms incorporate the status of the transmission queue in the scheduling rule. The Largest Weighted Delay First (LWDF) algorithm [18] considers the acceptable packet loss rate due to expired packet delivery deadlines as an additional scheduling weight. This method is particularly relevant for real-time services with specific packet delay and packet loss rate requirements. The *Modified Largest Weighted Delay First* (M-LWDF) algorithm [19] is an extension of the LWDF algorithm that incorporates the channel conditions, where the utility function is given by

$$U_n(t) = \alpha_n \frac{R_n(t)}{\bar{R}_n(t-1)} Q_n(t), \tag{3.3}$$

where $Q_n(t)$ is the transmit queue buffer length of user n, and α_n is a scheduling weight that can be defined for each user. In this algorithm, users with a large buffer in their queue are weighted more heavily. The M-LWDF algorithm thus incorporates the QoS demands of the users in the form of the queue buffer length.

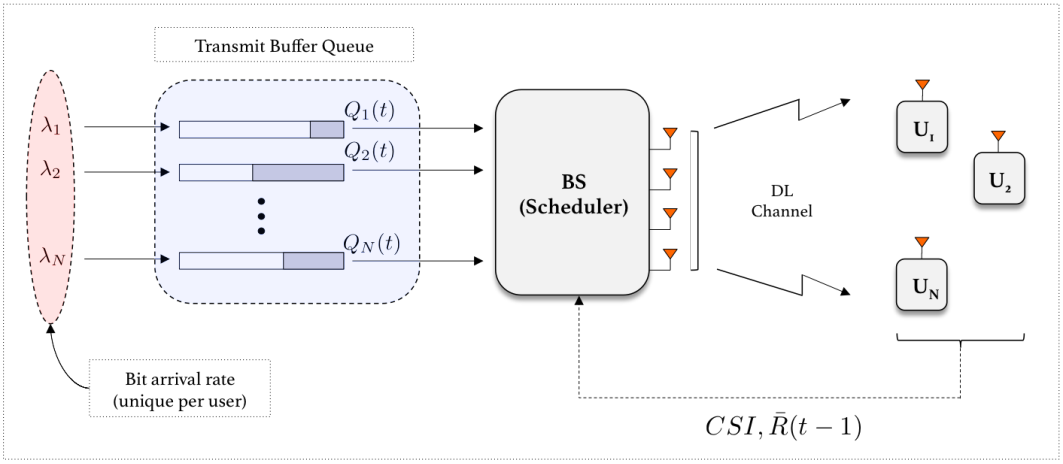


Figure 3.1: Downlink simulation setup.

3.2 Comparison of Utility-based Scheduling Algorithms

In this section, we compare the performance of the MT, PF, and M-LWDF scheduling algorithms in a MBB network. The purpose of the comparison is to study the tradeoffs in the scheduling algorithms in the context of MBB networks, where users pose a strict QoS requirement. In particular, we focus on the average delay for data transmission as a performance criteria.

3.2.1 Downlink Simulation Setup

We consider a downlink system with N users randomly distributed around a single base station as illustrated in Figure 3.1. Each user is either real-time (RT) or non-real-time (NRT). The traffic model is finite-buffer, where data for user n arrive in a transmit buffer queue at a constant bit arrival rate of λ_n bits per second. In each scheduling time instant, $\lambda_n \Delta T$ bits are pushed onto user n's queue, where ΔT is the scheduling time interval, which is set to

1 ms in this simulation. The base station uses the users' buffer queue status $Q_n(t)$, past throughput, as well as a known perfect channel state information (CSI) to make a scheduling decision. For each resource block, the appropriate utility metric is computed to schedule the users for data transmission. Data is transmitted according to a maximum ratio transmission (MRT) strategy [20] at the achievable rate through an i.i.d. Rayleigh fading MIMO channel with distant dependent path-loss with path-loss exponent of 3.8. The base station is equipped with 4 transmit antennas, and the users are all equipped with a single antenna. The base station's transmit power is 37 dBm, and the noise variance is -174 dBm/Hz. The total amount of bandwidth is 1.8 MHz, with each subband spanning 180 KHz each.

3.2.2 Evaluation Criteria

As discussed earlier, packet delay is a crucial QoS requirement for real-time applications. In this simulation, we consider the average delay of data transmission for each user. For a stable queuing system, Little's law [21] states that the average delay of user n is given by

$$\bar{D}_n = \bar{Q}_n/\lambda_n, \tag{3.4}$$

where $\bar{Q}_n$ is the average queue length of user n's transmit buffer. The average delay represents the amount of time the users' data have been waiting in the queue for transmission, and is a measure of packet delay performance.

Additionally, the distribution of the average delay is also a relevant performance metric. Even if the average delay across the users is low, some users

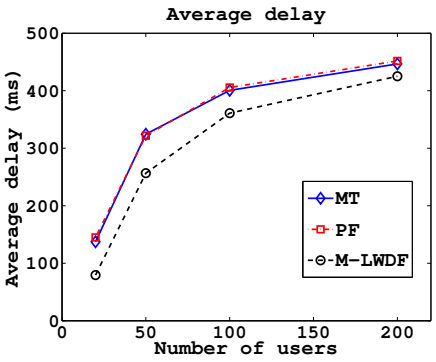
may suffer from a large delay. If users pose the same delay requirement, it is best to have a more equal distribution of the users' average delays. Jain's fairness index [22] has been used to evaluate the measure the fairness of resource allocation of communication systems. For the average delay, the fairness index is given by

$$J = \frac{(\sum_{n=1}^{N} \bar{D}_n)^2}{N(\sum_{n=1}^{N} \bar{D}_n^2)}. \tag{3.5}$$

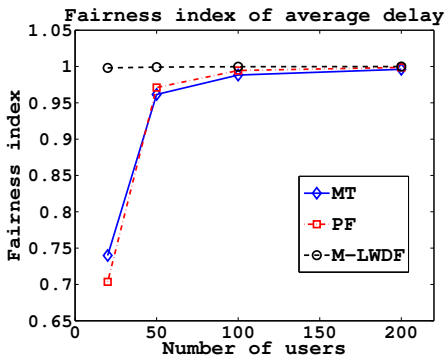
The index is bounded between 0 and 1, where a higher index represents a fair distribution of values.

3.2.3 Simulation Results

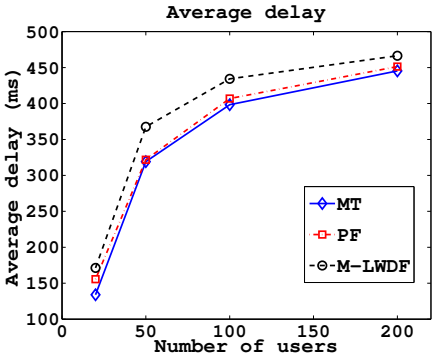
We assume that the network consists of 25% RT and 75% NRT users. For the M-LWDF algorithm, a higher priority weight α_n is used for RT users over the NRT users. Figures 3.2 (a)-(d) illustrate the average delay and its fairness index for the three scheduling algorithms (MT, PF, and M-LWDF) for the RT and NRT users, respectively. The simulation was performed for various values of N, the total number of users. Figures 3.2 (a) and (c) show that using the M-LWDF scheduling algorithm results in a lower average delay for RT users at the expense of that of the NRT users. This is because higher scheduling weights α_n are chosen for RT users over the NRT users, and thus RT users are more likely to be scheduled. In Figures 3.2 (b) and (d), we see that the fairness index of the average delay is far superior using the M-LWDF algorithm over the MT and PF algorithms for all values of N. This is because



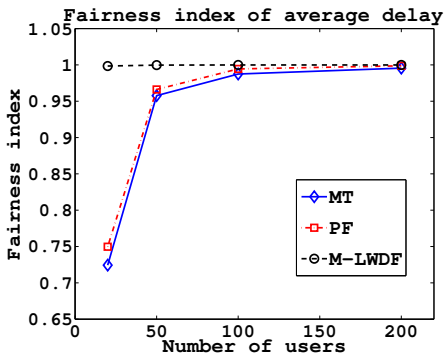
(a) Average Delay (RT)



(b) Average Delay Fairness (RT)



(c) Average Delay (NRT)



(d) Average Delay Fairness (NRT)

Figure 3.2: Average delay and its fairness index.

the M-LWDF algorithm is the only algorithm that incorporates the queueing dynamics of the network in the scheduling utility metric, which smoothes out the average delay across users. Although the PF algorithm may be fair in terms of resource allocation, neither the PF nor the MT algorithm results in a high fairness index of delays across the users. In contrast, the fairness index is almost 1 for all cases using the M-LWDF algorithm, which implies that all users have roughly the same average delay. This is extremely important for RT applications, where a tight delay budget must be met for each user.

3.3 Other Forms of Utility-based Algorithms

Besides the channel and queue-aware algorithms mentioned in this report, several other utility-based algorithms have been proposed. In this section, we describe the utility function and the characteristics of the exponential (EXP), logarithmic (LOG), and the maximum delay utility (MDU) algorithms.

The EXP algorithm [23] is similar to M-LWDF, but considers the network averaged delay in its scheduling metric. This is done by defining a normalized value of the total delay over the entire network in the utility function. In this algorithm, the utility function is defined as

$$U_n(t) = \alpha_n \exp(\frac{\alpha_n Q_n(t) - \bar{\chi}}{1 + \sqrt{\bar{\chi}}}) \frac{R_n(t)}{\bar{R}_n(t-1)} \tag{3.6}$$

where $\bar{\chi}$ is the average delay of N total users, given by

$$\bar{\chi} = \frac{1}{N} \sum_{n=1}^{N} \alpha_n Q_n(t). \tag{3.7}$$

A similar approach is done using the LOG rule [24], where the logarithmic function is used instead of the exponential. In this algorithm, the utility function is defined as

$$U_n(t) = \alpha_n \log(c + \alpha_n Q_n(t)) \frac{R_n(t)}{\bar{R}_n(t-1)} \tag{3.8}$$

where c is a constant. Because the logarithmic function saturates at large values, the EXP algorithm is considered more robust to combat long delays. Thus, the EXP algorithm aims to minimize the max-queue delay, while the LOG algorithm minimizes the sum-queue delay [25].

The MDU algorithm was developed to specifically consider QoS requirements for heterogeneous applications such as VoIP, streaming, and best-effort traffic [26]. In this algorithm, the utility function depends on the average delay in the transmission queue. The utility function is given by

$$U_n(t) = \begin{cases} (T_s \bar{D}_n(t)^{\gamma_{n,1}}) R_n(t)/\lambda_n & \text{if } \bar{D}_n(t) \leq \hat{D}_n \\ (T_s(\bar{D}_n(t)^{\gamma_{n,1}} - \hat{D}_n^{\gamma_{n,2}} + \hat{D}_n^{\gamma_{n,2}})) R_n(t)/\lambda_n & \text{if } \bar{D}_n(t) > \hat{D}_n \end{cases} \tag{3.9}$$

where $T_s, \lambda_n, \bar{D}_n, \hat{D}_n$ are the sampling period, average packet arrival rate, average queue delay, and the maximum allowed queue delay for user n. The variables $\gamma_{n,1}$ and $\gamma_{n,2}$ are constants and depend on the application type. Suggested values for these constants per flow can be found in [26].

3.4 Conclusion

This section discussed opportunistic OFDMA scheduling and resource allocation strategies for multi-user MIMO systems. Simulation results showed

that the M-LWDF scheduling algorithm yields higher fairness in the average delay distribution across the users. Moreover, priority metrics between RT and NRT users can be used to improve the average delay of certain traffic classes. The use of queue-aware scheduling algorithms is benefitial for MBB networks, where low latency is required for specific traffic classes.

Chapter 4

Quality of Experience for MBB Networks

Network optimization methods must efficiently control resources to provide service quality for mobile users in the network. For traditional voice services, objective network measurements such as throughput and packet loss rate were sufficient to quantify service quality for mobile users. However, with the increasing heterogeneity of broadband applications for MBB networks, it is necessary to characterize user satisfaction depending on the application in use. Considering the context of the application is an important way to quantify the quality of experience (QoE) that the user perceives. One way to quantify QoE is the mean opinion score (MOS), which has been used extensively in the research community. The MOS is a measure of user satisfaction that is typically measured through subjective studies, where users rate the quality of their service on a scale from 1 to 5. In this section, we explore proposed models of QoE using the MOS for VoIP calls, web browsing, and buffered video streaming services.

4.1 VoIP Calls

The primary contributions to the voice quality are the end-to-end delay and the packet loss rate. In [27], an intermediate quantity called the R-factor, was defined to measure the level of voice quality as a function of network performance metrics. The authors of [28] provide a simplified expression of the R-factor as

$$R = 94.2 - 0.24d - 0.11(d - 177.3)H(d - 177.3) - 11 - 40\ln(1 + 10e) \quad (4.1)$$

where d is the end-to-end delay in milliseconds, e is the loss rate, and $H()$ is the unit step function. Furthermore, the R-factor can be mapped to a MOS value given by

$$\mathrm{MOS}_{\mathrm{VoIP}} = \begin{cases} 1 & \text{if } R \leq 0 \\ 1 + 0.035R + (7 \times 10^{-6})R(R - 60)(100 - R) & \text{if } 0 < R < 100 \\ 4.5 & \text{if } R \geq 100. \end{cases} \quad (4.2)$$

4.2 Web Browsing

The primary indicator of service quality of web browsing traffic is the service response time. The service response time is defined to be the time period between when a user requests a web download and when the download has been complete. The authors in [29] conducted an experiment where participants were asked to rank the quality of a web browsing experience, as the service response time is manually controlled through the server. Through this

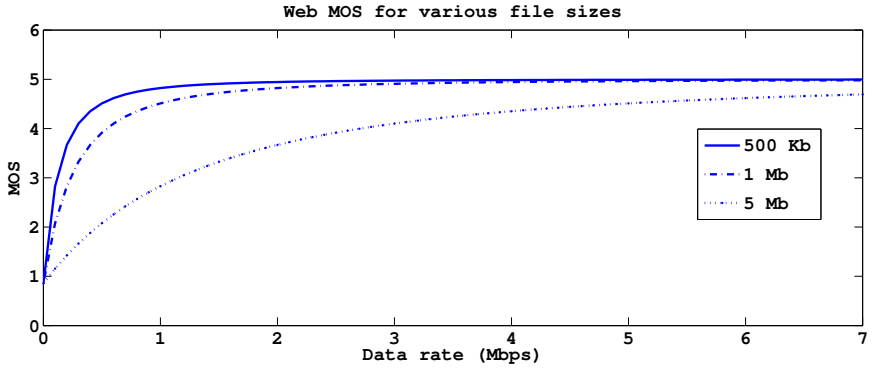


Figure 4.1: MOS metric for web browsing traffic for various file sizes.

experiment, the authors in [29] note that the QoE can be expressed as

$$\mathrm{MOS}_{\mathrm{web}} = 5 - \frac{578}{1 + (11.77 + \frac{22.6}{D})^2} \tag{4.3}$$

where D is the service response time in seconds. Moreover, the authors note that the service response time can be expressed as a function of the round-trip-time (RTT), the file size, and the supported data rate. However, because the LTE architecture is expected to support a relatively small RTT (less than 20 ms, ignoring wireline delays), the service response time can be approximated as $D \tilde{=} \frac{\mathrm{F}}{r}$, where F is the file size, and r is the supported data rate. It is important to note here that the QoE for web browsing traffic is not only governed by the data rate, but also the file size. Figure 4.1 illustrates the MOS for web browsing traffic as a function of the data rate for various file sizes. Clearly, more bandwidth must be allocated to users attempting to download larger websites to deliver the same level of user satisfaction. Another interesting phenomena that affects the QoE of web browsing users is the temporal variation in the downloading time through successive web browsing requests.

Referred to as the memory effect, the authors in [30] note that the QoE of the current web download is also a function of the previous one. Specifically, the MOS values tend to depend on the change in the page loading time of successive downloads. Thus, one way to enhace the web browsing experience may be to prevent sudden fluctuations in successive downloads in a web browsing session.

4.3 Buffered Video Streaming

The majority of video streaming on mobile devices operate on buffered HTTP video streaming (e.g. YouTube). In this method, video downloads are essentially similar to a large file transfer, where data are stored at the playback buffer. Rebuffering events occur if the buffer becomes empty as the video is being played at the user's terminal. Denoting T_{init} as the initial buffering time, T_{rebuf} as the average rebuffering time, and f_{rebuf} as the frequency of rebuffering events, the authors in [31] note that the MOS can be quantified as

$$\text{MOS}_{\text{video}} = 4 - 0.0672T_{init} - 0.742f_{rebuf} - 0.106T_{rebuf}. \tag{4.4}$$

From the above expression, we see that the QoE of video streaming services depends heavily on rebuffering events. Thus, the scheduler should ideally prioritize streaming users that are about to run out of video frames in its playback buffer. One way to achieve this would be to feed back remaining frames in the buffer, and quantize this value in the time domain as service deadline (i.e. determine how much time may pass until the next video frame must be transmitted).

4.4 Conclusion

In this section, we presented adopted ways to quantify the QoE for voice, web browsing, and streaming application types. While the exact computation and parameters vary depending on the application, the primary driver of QoE is the service delay. Because the traffic volume may differ depending on the application type, efficient scheduling and resource allocation strategies should also balance between opportunism and service delay appropriately for the traffic demands. For instance, because VoIP calls constitute a relatively low volume in size compared to data traffic, delivering satisfactory service for voice users could be accomplished by reserving bandwidth for real-time services for low latency and call drops. The remaining bandwidth should be carefully allocated between the non-real-time data traffic appropriately for the traffic volume and the required service delay.

Chapter 5

Scheduling For Non-Real-Time Flows

As noted in the previous section, the user experience of web browsing users depends on the download time. For real-time traffic, where data is always available to be transmitted, the data rate is an appropriate measure of QoE. However, the data rate does not capture the transaction's characteristics for non-real-time flows, such as the file size and the download time. In order to effectively schedule non-real-time flows for data transmission, it is important to identify appropriate performance metrics that reflects a context-aware QoE. The purpose of this section is to define a performance measure and to formulate a scheduling framework that captures the QoE of web browsing traffic.

5.1 Perceived Throughput

Consider a web browsing traffic requesting a download of file size F. Let D denote the amount of time it takes for the download to be completed. We define the perceived throughput T_p as

$$T_p = \frac{F}{D}. \tag{5.1}$$

It is important to note that the download time D depends on how multiple transactions are scheduled. For example, suppose that user 1 and user 2

request a download at the same time. If user 1 is scheduled before user 2, then the total download time for user 2 is the sum of the download times of the two users. In this way, the perceived throughput captures the file size and the entire download time, unlike the allocated data rate.

5.2 Suboptimality of Simultaneously Serving Multiple Transactions

In the previous section, we have defined the perceived throughput as a performance measure for web browsing traffic. Due to the time-varying nature of non-real-time traffic, the scheduler must allocate resources among multiple web browsing users competing for radio resources. In this section, we will show that any fractional resource allocation to schedule multiple transactions is suboptimal in terms of the perceived throughput. Consequently, scheduling of web browsing traffic should be done one transaction after another.

Suppose two download requests of size F_1 and F_2 bits arrive simultaneously. The network has a fixed network capacity of C in bits per second. Let α be the amount of resources allocated to transaction 1, such that αC and $(1-\alpha)C$ are the rates allocated to transactions 1 and 2. Without loss of generality, we assume that transaction 1 completes before transaction 2, such that its perceived throughput is exactly αC.

Let us now find an expression of the perceived throughput of transaction 2, which finishes after transaction 1. When transaction 1 is complete, the amount of bits remaining for transaction 2 is $\tilde{F_2} = F_2 - \frac{F_1}{\alpha C}(1-\alpha)C = F_2 -$

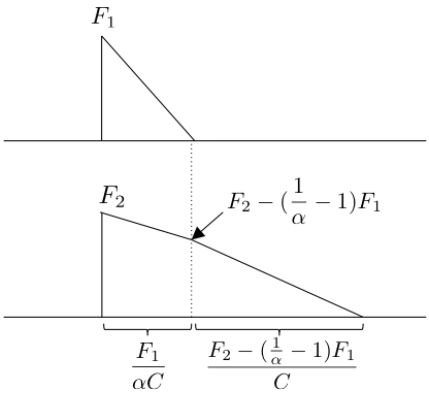


Figure 5.1: Download times when two transactions share resources.

$(\frac{1}{\alpha} - 1)F_1$. See figure 5.1 for an illustration of this scenario.

When transaction 1 is complete, we can allocate the entire capacity to transaction 2, such that its rate is C. Since the amount of time it takes to complete the transaction is now $\tilde{D}_2 = \frac{\tilde{F}_2}{C}$, we can compute the perceived throughput as

$$T_{p,2} = \frac{F_2}{D_2} \tag{5.2}$$

$$= \frac{F_2}{D_1 + \tilde{D}_2} \tag{5.3}$$

$$= \frac{F_2}{\frac{F_1}{\alpha C} + \frac{F_2}{C} - (\frac{1}{\alpha} - 1)\frac{F_1}{C}} \tag{5.4}$$

$$= \frac{F_2}{F_1 + F_2} C. \tag{5.5}$$

The above result shows that regardless of the value of α, the perceived throughput of the transaction that finishes later is the same. Since the capacity is fixed, the perceived throughput is the same as if both files were being downloaded at the same time. This means that in terms of the perceived throughput, the

optimal values of α is either 0 or 1, i.e. the scheduler should finish one transaction or the other without splitting resources. This observation motivates a scheduling algorithm that is somehow aware of the perceived throughput.

5.3 Gradient-based Active-average Scheduling

As noted in the previous section, we wish to design a scheduler that optimizes the network performance in terms of the perceived throughput T_p. Instead of using T_p, let us define the average active rate, $\bar{r}_n^a$, which is the rate of user n averaged only when the user is *active*. The average active rate can be seen as a proxy for the perceived throughput. We will now propose a scheduler that utilizes the average active rate in the scheduling decision.

Suppose that there are N users requesting a sequence of web download transactions, and the scheduler makes a scheduling decision whenever the number of active users changes. Let us define $U_n(\bar{r}_n^a(t))$ as the utility function that we wish to maximize, where $\bar{r}_n^a(t)$ is the average active rate of user n at time t. The goal of the scheduler is to find the optimal scheduling decision at time t that solve the problem

$$\begin{aligned} \text{maximize} \quad & \sum_{n \in N_a(t)} U_n(\bar{r}_n^a(t)) \\ \text{subject to} \quad & \sum_{n \in N_a(t)} d_n(t) \leq C \\ \text{over} \quad & d_n(t) \geq 0,\ n \in N_a(t) \end{aligned}$$

where $N_a(t)$ is the set of active users at time t, C is the network capacity, and $d_n(t)$ is the instantaneous bit rate allocated to user n at time t. As noted

in the earlier section, the optimal solution is to schedule one transaction at a time. If we assume that only one transaction is scheduled in each decision, the best that we can do is to move in a direction that improves the total utility. Using a gradient-based scheduling approach, we schedule the user that provides the greatest differential increase in the utility at time t. Assuming the proportionally fair utility function, defined by the logarithmic utility $U_n(\bar{r}_n^a(t)) = \log(\bar{r}_n^a(t))$, we schedule the user using the rule

$$n^* = \arg\max_n \{U'(\bar{r}_n^a(t))\, r_n(t)\} \tag{5.6}$$

$$= \arg\max_n \left\{ \frac{r_n(t)}{\bar{r}_n^a(t)} \right\}, \tag{5.7}$$

where $r_n(t)$ is the achievable rate for user n at time t. We assume that the active average rate is updated after every scheduling decision, only for *active* users, according to a time-averaging filter defined by

$$\bar{r}_n^a(t+1) = (1 - \frac{1}{\tau})\bar{r}_n^a(t) + \frac{1}{\tau} d_n(t), \tag{5.8}$$

where $d_n(t)$ is the bit rate allocated to user n at time t.

5.4 Simulation Performance Analysis

Traditionally, scheduling decisions are performed over each transmission time interval. In this section, we compare the active-average scheduling algorithm as described in section 5.3 with the traditional scheduling algorithm. We assume that the download transactions arrive as a poisson process, where the inter-arrival time is an exponential random variable with mean λ. The

file size of each transaction follows a lognormal distribution with mean μ and variance σ. The parameters of the random variables are the same for each user, and the scheduling decision is made every time the number of active transaction changes. For the purpose of simplicity, no fading is considered, and all users have thes same distance from the base station.

We compare the active-average scheduling algorithm with the traditional proportionally fair (PF) algorithm, which is performed in each scheduling time interval. In this algorithm, we assume that users are scheduled in each scheduling time interval, according to the rule

$$n^* = \arg\max_n \left\{ \frac{r_n(t)}{\bar{R}_n(t)} \right\}, \tag{5.9}$$

where $\bar{R}_n(t)$ is the average rate of user n at time t. The difference in this algorithm compared to the active-average approach is that both the scheduling decision is made in every scheduling interval, and the average rate is averaged regardless if the user is or is not active.

Let us define the *utility gain* as the ratio of the total utility summed across all users in the active-average scheduling algorithm over the tradtional PF algorithm. Specifically, the utility gain G_{util} is given by

$$G_{\text{util}} = \frac{\sum_{n=1}^{N} \log(\bar{T}_{p,n}^{\text{trans}})}{\sum_{n=1}^{N} \log(\bar{T}_{p,n}^{\text{PF}})}, \tag{5.10}$$

where $\bar{T}_{p,n}^{\text{trans}}$ and $\bar{T}_{p,n}^{\text{PF}}$ are the average perceived throughputs of user n (averaged over all transactions) using the active-average and traditional PF schedulers,

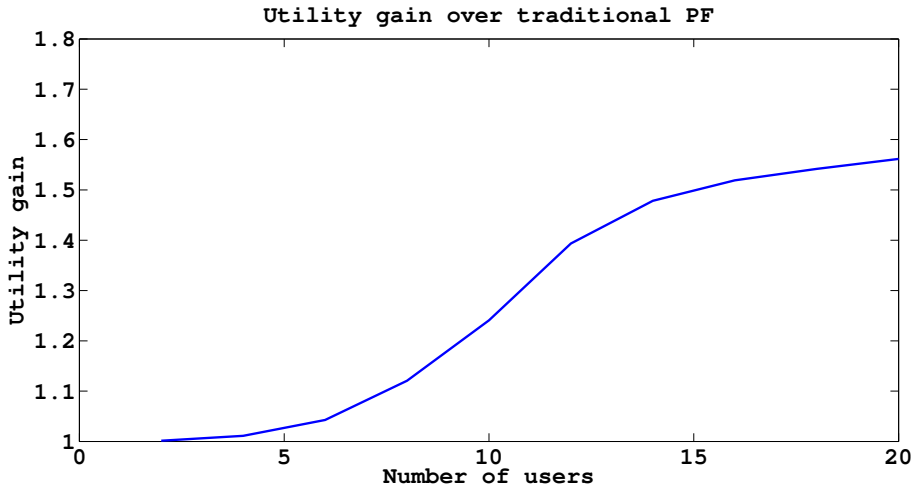


Figure 5.2: Utility gain of the active-average scheduler. Simulation parameters of $\lambda = 0.5$, $\mu = 10$, and $\sigma = 0.2$ were used.

respectively. The utility gain represents the gain in performance of the active-average scheduler in terms of the perceived throughput. Figure 5.2 shows this utility gain for various total number of users, N. Notice that we always have the utility gain greater than 1, because sharing resources between transactions is always suboptimal, as shown in section 5.2. The gain increases with N due to the increase in the number of potentially active users. As N increases, the traditional PF scheduler attempts to split resources equally among the competing users, resulting in a lower perceived throughput.

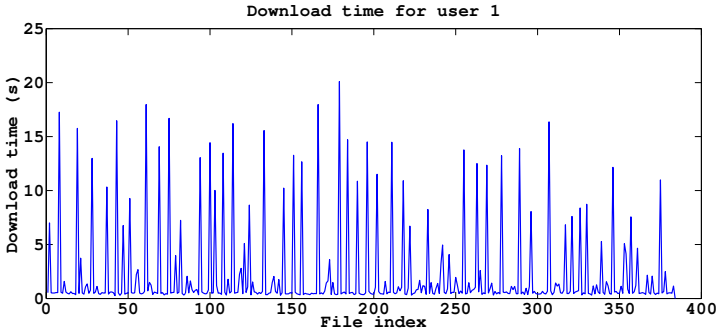
5.5 Variability of Download Times

Although we see a gain in the perceived throughput using the active-average scheduling algorithm, the randomness in the number of active users

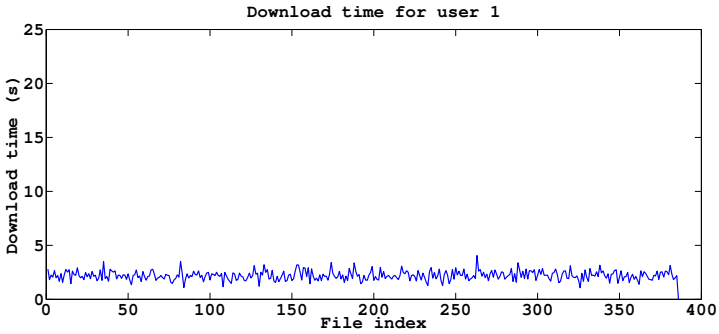
cause variation in the throughput over time. Since the only objective of the scheduler is to maximize the average perceived throughput, the variability in the download times across the transaction sequence is ignored. This is a problem for web download users, because sudden changes in download times directly effects the QoE, as described in the previous chapter. Figures 5.3 (a) and (b) show a plot of the download times for a single user using the active-average scheduler and the traditional PF scheduler, respectively. We can clearly see the large spikes in the download times when using the active-average scheduler compared to the relatively constant download times in the traditional scheduler. The variability becomes worse as the number of users (and consequently potentially active users) increases. We can, however, remedy this problem by adding an exponential weight proportional to the current download time in the scheduling decision, where we modify the scheduling rule as

$$n^* = \arg\max_n \left\{ \exp(a\, W_n(t)) \frac{r_n(t)}{\bar{r}_n^a(t)} \right\}, \tag{5.11}$$

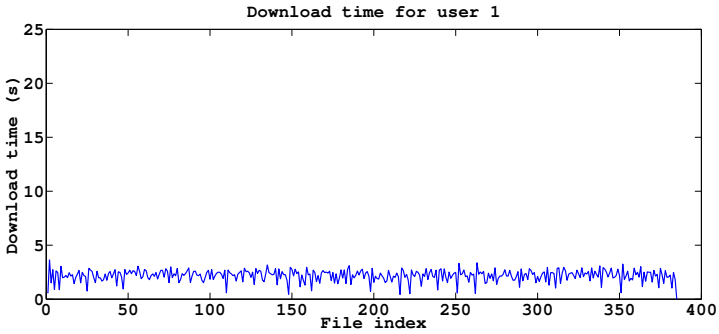
where a is a constant and $W_n(t)$ is the current flow waiting time in seconds for user n at time t. This weight can be interpreted as the priority of the user, which grows exponentially in the current waiting time. Figure 5.3 (c) shows a plot of the download time when the modified active-average scheduling rule is used with $a = 1$. Using this modified approach, we no longer see the large variations in the download time as seen in 5.3 (a). The modified rule provides a simple solution to reduce the variability in the download time across



(a) Active-average Scheduler



(b) Traditional PF Scheduler



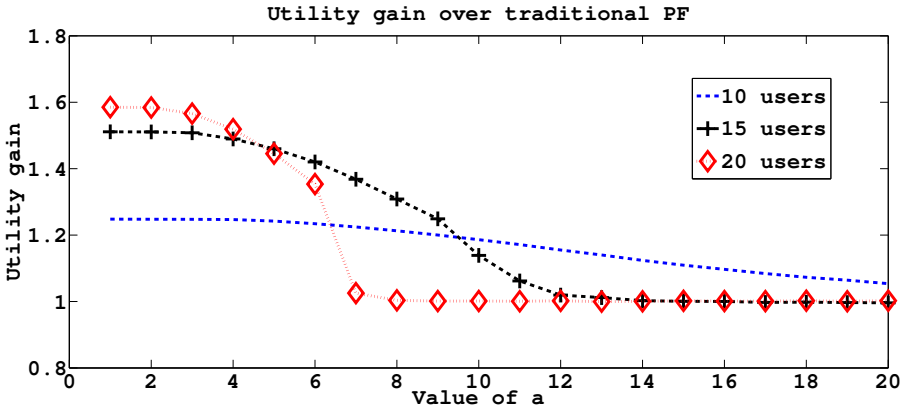
(c) Active-average Scheduler with weight a=1

Figure 5.3: Waiting times across a sequence of downloads for $N = 5$.

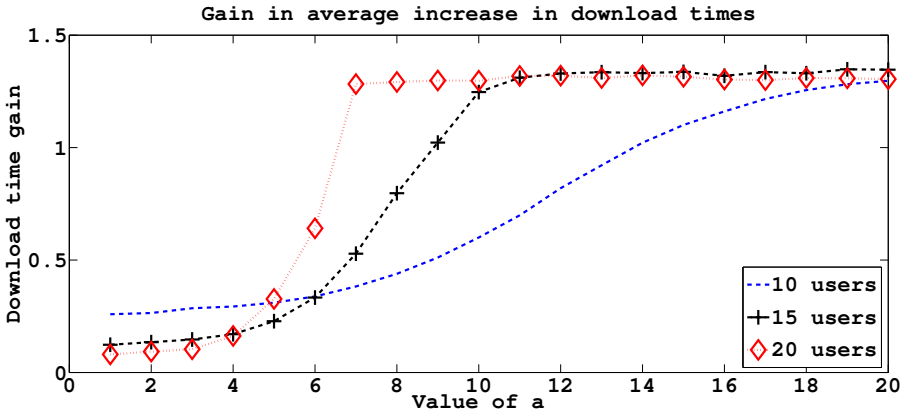
transactions. Let us now investigate this effect by defining the average increase in the download time for user n as

$$\Delta D_n = \frac{1}{K_n - 1} \sum_{k=1}^{K_n - 1} \max(0, D_n(k+1) - D_n(k)) \tag{5.12}$$

where K_n is the total number of downloads requested by user n, and $D_n(k)$ is the total download time of the k^{th} transaction for user n. This value quantifies the average degredation in the user experience due to increases in the download times across subsequent web downloads. We now define the *download time gain* as the ratio of ΔD_n, averaged across all users, of the traditional PF algorithm over the active-average scheduler with the exponential weight. Figures 5.4 (a) and (b) show the utility gain and the download time gain for various values of the constant a. We clearly see a tradeoff in these two illustrations, as increasing the value of a increases the download time gain, but decreases the gain in utility as a function of the perceived throughput. This may be because by increasing the value of a, we prioritize transactions with long waiting times regardless of the file size. Thus, in the case that requests with large file sizes are waiting for a long time, the scheduler prioritizes those users over the others. This causes unfairness in the perceived throughputs across the users, which may result in lower utility gain. However, the exact relationship between a and the utility gain is not explored in this report and is left for future work.



(a) Utility Gain



(b) Download Time Gain

Figure 5.4: Utility and download time gains of the active-average scheduler with exponential weights for various values of a.

5.6 Conclusion

In this section, we developed the perceived throughput as a performance measure that captures the QoE of download traffic that incorporates both the file size and the downloading time. Based on the perceived throughput, we designed a active-average scheduling algorithm for non-real-time web browsing traffic using the gradient-ascent method. Simulation results showed that the traditional proportionally fair scheduling algorithm that operates at regular scheduling time intervals was suboptimal compared to the proposed approach. We also saw that although the active-average scheduler imposes variability in download times, adding an exponential weight as a function of the waiting time can be used to balance the perceived throughput and the variability in download times.

Chapter 6

Conclusion

The increasing variety of applications in mobile devices have developed diverse traffic in MBB networks consisting of both voice and data services. In this report, we explored statistical models of traffic characteristics of various service classes, including VoIP calls, file transfers, web downloads, and video streaming. We also discussed utility-based scheduling and resource allocation algorithms and compared their performance in terms of average delay and delay fairness. Models to quantify the user experience, in terms of the MOS, was also described for VoIP, web browsing, and video streaming applications. Finally, we proposed and analyzed a transaction-based scheduling algorithm for non-real-time web browsing traffic. Simulation results showed that transaction-based scheduling provides peformance gains compared to the traditional scheduling approach.

As more applications become available for use in mobile devices, the diversity of traffic classes in mobile networks is expected to expand in the future. As the heterogeneity of mobile traffic characteristics increases, it is important for network optimization strategies to incorporate application-specific QoE, such as the ones described in this report. For future work, it will be interest-

ing to study QoE models and how it affects scheduling and resource allocation algorithms for more types of mobile traffic, such as social media and machine-to-machine applications. It will also be interesting to study the effects of base station cooperation or interference management techniques in enhancing the user experience.

Bibliography

[1] Ericsson, "Ericsson Mobility Report," Tech. Rep., November 2013.

[2] Sandvine, "Global Internet Phenomena Report," Tech. Rep., July 2013.

[3] 3GPP, "Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN)," 3GPP TR 25.913 version 9.0.0, December 2009.

[4] H. Ekström, "QoS Control in the 3GPP Evolved Packet System," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 76–83, 2009.

[5] P. Ameigeiras, Y. Wang, J. Navarro-Ortiz, P. E. Mogensen, and J. Lopez-Soler, "Traffic models impact on OFDMA scheduling design," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 61, February 2012.

[6] "3GPP TR 36.814 version 9.0.0 Release 9," 3GPP, Tech. Rep., March 2010.

[7] G. C.R1002-0, "cdma2000 Evaluation Methodology," Tech. Rep. Version 1.0, December 2004.

[8] *WiMAX System Evaluation Methodology*, 2nd ed., WiMAX Forum, December 2007.

[9] *Next Generation Mobile Networks Radio Access Performance Evaluation Methodology*, 1st ed., January 2008.

[10] *IEEE 802.16m Evaluation Methodology Document (EMD)*, IEEE 802.16 Broadband Wireless Access Working Group, January 2009.

[11] F. Khan, *LTE for 4G Mobile Broadband : Air Interface Technologies and Performance*. Cambridge University Press, March 2009.

[12] V. Jacobson, "Congestion Avoidance and Control," *In Proceedings of SIGCOMM*, August 1988.

[13] H. Choi and J. Limb, "A Behavioral Model of Web Traffic," in *International Conference on Network Protocols*, no. 7, October 1999, pp. 327–334.

[14] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, June 1999.

[15] F. D. Smith, F. H. Campos, K. Jeffay, and D. Ott, "What TCP/IP Protocol Headers Can Tell Us About the Web," in *in ACM SIGMETRICS*, 2001, pp. 245–256.

[16] [Online]. Available: http://www-tkn.ee.tu-berlin.de/research/trace/ltvt.html

[17] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability,"

Journal of the Operational Research Society, vol. 49, no. 3, pp. 237–252, March 1998.

[18] A. Stolyar and K. Ramanan, "Largest Weighted Delay First Scheduling: Large Deviations and Optimality," *Annals of Appplied Probability*, vol. 11, pp. 1–48, 201.

[19] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting, "Providing Quality of Service over a Shared Wireless Link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150–154, 2001.

[20] T. Lo, "Maximum Ratio Transmission," *IEEE Trans. on Communications*, vol. 47, no. 10, pp. 1458–1461, October 1999.

[21] D. Chhajed and T. Lowe, *Building Intuition: Insights From Basic Operations Management Models and Principles*. Springer, 2008.

[22] R. Jain, D.-M. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," Easter Research Lab, Tech. Rep., December 1984.

[23] S. Shakkotai and A. Stolyar, "Scheduling Algorithms for a Mixture of Real-Time and Non-Real-Time Data in HDR," in *In Proceedings of 17th International Teletraffic Congress*, pp. 793–804.

[24] B. Sadiq, S. J. Baek, and G. de Veciana, "Delay-Optimal Opportunistic Scheduling and Approximations: The Log Rule," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, April 2011.

[25] B. Sadiq, R. Madan, and A. Sampath, "Downlink Scheduling for Multiclass Traffic in LTE," *EURASIP Journal on Wireless Communications and Networking*, November 2009.

[26] G. Song, "Cross-Layer Resrouce Allocation and Scheduling in Wireless Multicarrier Networks," Ph.D. dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology, August 2005.

[27] I.-T. R. G. 107, "The E-Model, a computational model for use in transmission planning," December 1998.

[28] R. G. Cole and J. H. Rosenbluth, "Voice over IP Performance Monitoring," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2, pp. 9–24, 2001.

[29] P. Ameigeiras, J. Ramos-Munoz, J. Navarro-Ortiz, P. Mogensen, and J. Lopez-Soler, "QoE-oriented cross-layer design of a resource allocation algorithm in beyond 3G systems," *Computer Communications*, vol. 33, no. 5, pp. 571–582, March 2010.

[30] T. Hosfeld, S. Biedermann, R. Schatz, and P. A., "The memory effect and its implications on Web QoE modeling," *Proceedings of the 2011 23rd International Teletraffic Congress*, pp. 103–110, September 2011.

[31] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," *In Proceedings of the*

12th IFIP/IEEE International Symposium on Integrated Network Management, 2011.