Multi-user Multi-flow Packet Scheduling for Wireless Channels

vorgelegt von
Ana Cristina Costa Aguiar

Von der Fakultät IV - Elektrotechnik und Informatik
der Technichen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
Dr. Ing.

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr.-Ing. Dr. rer. nat. Holger Boche
Berichter: Prof. Dr.-Ing. Adam Wolisz
Berichter: Prof. Dr.-Ing. Eckehard Steinbach

Tag der Wissenschaftlichen Aussprache: 09.06.2008

Berlin 2008
D83

Abstract

This dissertation focuses on how to share a time-division multiplexed wireless channel among packets of different applications with different packet delivery requirements. The novelty of the approach lies in the goal of the allocation: improving user perceived application quality. The approach proposed is based on the paradigm that resource usage that does not improve user perceived quality is ineffective. Utility curves are used to map network service to perceived user quality, expressing the user sensitivity to delivered network service for each application. Based on the history of the network service delivered to a flow, the current perceived quality can be evaluated. Similarly, the quality increase achieved by transmitting a flow's packet, and the quality decrease caused by the deferral of its transmission for some time, while another flow uses the channel, can be calculated. The packet scheduler proposed—PeLe—allocates the channel to the flow that maximises the sum of the overall quality after transmission of its packet. This accounts for the quality increase of the flow whose packet is transmitted and the quality decrease of the flows which have to defer transmission. The link quality is reflected by the duration of a packet's transmission, which depends on the usable transmission rate; a low link quality requires the use of a robust modulation and coding scheme, which causes a longer channel occupation time. This causes a larger degradation of other flows' quality if that packet is scheduled and leads to a lower overall quality after the transmission.

The behaviour and performance of the PeLe scheduler are studied using discrete-event simulations with realistic traffic models for four traffic types: bulk file transfers, web traffic (both using an implementation of TCP New Reno), VoIP calls and H.264 variable bit rate video streaming. The performance of the scheduler is evaluated using application-specific metrics for user perceived quality. The fast fading of the links used in the evaluation is modelled using measurement traces obtained from a WLAN channel measurement campaign in low mobility environments carried out for the purpose. The results show that the PeLe scheduler provides a higher average number of satisfied users than the reference schedulers, at a maximum cost of 12% lower overall channel throughput.

Channel-adaptive mechanisms require a prediction of the wireless link be-

haviour at the transmitter. This dissertation studies the accuracy of the One Step (OS) prediction heuristic, which assumes the received signal stays constant at the last measured values, and compares it with 3 alternative prediction methods using the WLAN measurement traces. The OS heuristic shows the lowest mean squared prediction error up to horizons of 2 ms; beyond that horizon the prediction performance of all studied mechanisms is comparable to a moving average. The impact of the prediction errors in an adaptive rate scheme is that in up to 30% of the times a less robust modulation than required is chosen. That leads to an increase of an order of magnitude in packet loss rate. Finally, the impact of the link prediction errors on the performance of opportunistic packet schedulers is studied. The results show that the PeLe scheduler is less sensitive to link prediction errors than opportunistic schedulers because the scheduling metric defined only indirectly takes the link quality into account.

Altogether, the results obtained in this work show that the approach proposed can improve the user perceived performance of wireless communications under realistic traffic and system conditions.

Zusammenfassung

Im Rahmen dieser Dissertation wird untersucht wann die Pakete unterschiedlicher Datenströme, mit entsprechend unterschiedlichen Anforderungen hinsichtlich der Übertragung, über einen zeitmultiplexen Kanal bestmöglich gesendet werden können. Da die Qualität der Kommunikation vom Benutzer bewertet wird, ist das Ziel des hier vorgeschlagenen Verfahrens die von den Benutzern wahrgenommene Qualität im Vergleich zu herkömmlichen Verfahren zu verbessern. Utility-Kurven stellen den Zusammenhang zwischen Netzwerkressourcen und der wahrgenommenen Qualität dar und drücken aus, wie empfindlich die Benutzer einer Applikation bezüglich der erhaltenen Netzwerkressourcen sind. Die Qualität eines Datenstroms lässt sich basierend auf den erhaltenen Netzwerkresourcen ermitteln; ebenso lassen sich der Qualitätszuwachs der dadurch erfolgt, dass ein Paket dieses Stromes übertragen wird, und die Qualitätssenkung aufgrund des Wartens, dass ein Paket eines anderen Datenstromes übertragen wird, ermitteln. Die Qualität des Funkkanals beeinflusst die Übertragungsrate und folglich auch die Verweildauer eines Paketes auf dem Kanal; je niedriger die Qualität der Verbindung, desto niedriger die verwendbare Datenrate und desto länger belegt das Paket den Kanal. Der in dieser Arbeit vorgeschlagene Paket-Scheduler—PeLe–teilt dem Datenstrom den Kanal zu, der die Gesamtqualität aller Datenströme nach der Übertragung maximiert. Dabei wird der Qualitätzuwachs des Datenstromes, dessen Paket übertragen wird, mit der Qualitätssenkung aller anderen Datenströme, die darauf warten müssen, ausgewogen.

Das Verhalten des PeLe Paket-Schedulers und seine Leistung werden ausgewertet anhand von diskreten, ereignisorientierten Simulationen, bei denen realistische Verkehrsmodelle für vier unterschiedliche Beispielapplikationen eingesetzt werden: Dateiübertragungen, WWW-Verkehr (beide benutzen eine Implementierung von TCP New Reno), VoIP-Telefonate und H.264 Video. Die Leistung des Schedulers wird ausgewertet anhand von applikationsspezifischen Metriken, die die vom Benutzer empfundene Qualität ausdrücken. Die Fast-Fading-Variationen der Kanalqualität werden in den Simulationen durch Traces aus im Laufe dieser Arbeit durchgeführten WLAN-Kanalmessungen modelliert. Die Ergebnisse zeigen, dass der PeLe Scheduler im Mittel eine höhere Benutzerqualität erreicht, auf Kosten eines um 12% niedrigeren Ges-

amtdurchsatzes.

Da kanaladaptive Verfahren Bestandteil jedes Funkkommunikationssystems sind, ist am Sender eine Vorhersage über das Verhalten des Kanals notwendig. Diese Arbeit untersucht die Genauigkeit einer Vorhersage-Heuristik, die annimmt, dass der Kanal in naher Zukunft konstant bleibt—One Step (OS), und vergleicht sie mit drei anderen Verfahren. Die OS-Heuristik weist den geringsten Vorhersagefehler bis 2 ms auf und ist die zweitbeste für längere Vorhersagehorizonte. Die Vorhersagefehler führen dazu, dass 30% der Pakete mit einer zu hohen Übertragungsrate übertragen werden, und zu einer um eine Größenordnung erhöhte Paketfehlerrate. Schließlich wird untersucht, wie die Wahl einer falschen Übertragungsrate aufgrund von Kanalvorhersagefehlern sich auf die Leistung von kanaladaptiven Paketschedulern auswirkt. Die Ergebnisse zeigen, dass die Leistung des PeLe Paket-Schedulers weniger anfällig gegenüber Vorhersagefehlern ist als andere opportunistische Scheduler, weil die Qualität des Kanals nur indirekt die vom Scheduler benutzte Metrik beeinflusst.

Acknowledgements

I want to thank Prof. Adam Wolisz for his encouragement and patience, and for fruitful discussions and constructive suggestions.
I also thank the members of TKN for the nice working atmosphere and for all their support and help. I especially thank Holger Karl for his scientific advice and many helpful discussions at the beginning of my work.
Jirka Klaue was always there, from the start to the end of this work. His friendship and availability to discuss more or less anything at any time are invaluable. I am also very grateful for his help in debugging, in the analysis of the channel measurement data, and in the integration of Evalvid. I also want to thank Christian Hoene for sharing with me his VoIP knowledge and his evaluation tools. I want to thank everybody who proofread any part of this thesis, and especially Daniel Willkomm and Matthias Bohge for the suggested improvements in the results chapters on such a short notice.
I thank Prof. Eckehard Steinbach for conducting the expertise on my thesis.
I thank João Barros and his group for their kindness to host me at the Department of Computer Science of the University of Porto for 2 months while writing this thesis.
I also thank the Fundação para a Ciência e Tecnologia (FCT) for the 4 year doctoral degree grant SFRH/BD/4610/2001 that financed great part of the work for this thesis.
Finally, I thank all my family, especially my parents and sister, and my friends, especially Sven, Mirco, André and Gabi, for their invaluable support and motivation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Thesis Overview

Following the ubiquitous availability of wireless Internet, users expect nowadays to use over wireless accesses the same kind of applications as they use on fixed ones. Moreover, Internet usage has become diversified and, in addition to traditional Internet applications like file download, email or web surfing, audiovisual and real time applications like voice over IP (VoIP), on-demand video streaming (VoD), radio/TV broadcast, gaming and peer-to-peer applications, have all become an important part of the traffic mix. This thesis deals with the problem of scheduling packets from multiple, diversified data flows towards multiple users sharing a Time Division Multiple Access (TDMA) wireless channel, with the goal of delivering satisfactory user-perceived quality for all applications at high loads.

One way of transmitting data satisfactorily for the users is resource overprovisioning, in which case the requirements of the data flows from all applications can be met without the need for a sophisticated resource allocation mechanism—a look at the Internet backbone suffices to confirm this. But wireless bandwidth is scarce and expensive, thus overprovisioning is not affordable. Therefore, wireless channels are used at higher loads than fixed accesses. In this situation, it is not sufficient to serve data flows regardless of their requirements, as this might lead to impairments in the quality perceived by the users because the specific requirements of each data flow may be met only by chance.

The plethora of applications carried over a network these days has equally diversified requirements on the delivery of its packets. Some applications require error-free and in-order packet delivery, but they are not sensitive to delays or delay variations below a couple of seconds, like file downloads or email. Web traffic or instant messaging tolerate less delay in the delivery of a requested page or message but still require error-free delivery. Differently, audiovisual applications can tolerate some amount of lost packets without significant decrease of the perceived quality, though they have more stringent requirements on delay. Furthermore, there are significant differences among audiovisual applications: voice over IP (VoIP) or video-conference calls have stricter requirements on the delay and jitter due to the interactivity between two parties than audio or video streaming, which counts on a playout buffer to smooth variations in the network conditions.

Besides high load, the wireless medium raises other challenges. Wireless channels are characterised by their time- and location-dependent error behaviour. On the one hand, the quality of a wireless link between a sender and a receiver varies in time, oscillating between periods of high error rates—low link quality—and periods of low error rates—high link quality; on the other hand, the wireless links between a sender and two receivers at different locations experience different qualities at the same time. Efficient multi-user packet scheduling must take these two aspects into account: it should not let a user experiencing bad link quality transmit—channel-awareness—and instead allocate the channel to a user experiencing good link quality to profit from multi-user diversity. Further improvements can be achieved by a clever choice among the users experiencing good links, since the better the link quality the higher the data rate that can be used for the transmission. These approaches focus on effectively using the wireless channel from the channel's perspective, not taking into account at all which kind of data is being transmitted. This work argues that the user perceived quality should also be considered in the scheduling decision.

1.2 Contributions of the Thesis

The main contribution of this thesis is a novel approach to packet scheduling over a TDMA wireless channel with the goal of improving the overall user perceived quality. The packet scheduler proposed—PeLe—ranks packets based on a scheduling metric that considers two aspects: on the one hand the link quality variations intrinsic to the wireless channel, on the other hand the importance of transmitting a packet for the overall quality perceived by the users of individual flows.

This work differentiates between network service quality (for short called network service), e. g. throughput, delay, packet loss, etc, and user perceived application quality (for short called user perceived quality or just perceived quality) which describes how a user experiences the performance of the application. The proposed solution takes advantage of the fact that the user perceived quality is an application dependent, strongly non-linear function of the network service. An application dependent utility curve is used to express that relationship. The utility curve is used to assess the importance of a packet for the user perceived quality and to compare the importance of packets from different flows. The scheduling metric weighs the quality improvement achieved by a packet's transmission with the quality decrease imposed on the flows that have to wait for that transmission. The link quality is taken into account in this last step.

Further, this work studies the trade-off between perceived quality and efficient resource usage in realistic traffic scenarios and using appropriate metrics for user perceived quality. Four applications shall be used to depict the problem and evaluate the solution proposed:

- bulk data transfer (Bulk),
- web traffic (WWW),
- voice over IP (VoIP),
- video streaming (VoD).

These applications stand for background TCP traffic (Bulk), interactive TCP traffic (WWW), interactive audio (VoIP) and streaming video (VoD) traffic and make up representative shares of the traffic mix in current networks. The corresponding main performance metrics are

Bulk: average goodput,

WWW: goodput per page,

VoIP: R-factor,

VoD: the percentage of video frames in an interval that has lower quality than the encoded video before transmission.

Besides the proposed packet scheduling schemes, other well-known wireless packet schedulers are studied: the proportional fair scheduler, the exponential fair scheduler, the modified largest weight first scheduler and a channel-aware round robin schedule.

The previous discussion assumes that the quality of the wireless links is known to the scheduler at the time of the scheduling decision, implicitly assuming that this knowledge is available. Although it is not so in reality, it is possible to predict the link behaviour over short horizons based on past link behaviour. Because that prediction is not perfect, prediction errors sometimes lead to unexpected behaviour of mechanisms that base their decisions on the link quality prediction. This thesis makes a comparative evaluation of heuristics for prediction of wireless link quality in low mobility environments in the ISM band in terms of both the magnitude of the prediction errors and their effects in the performance of an adaptive rate scheme. Based on the results, a change to the rate adaptation mechanism is proposed, which is more conservative when the probability of choosing the wrong rate due to link prediction errors is higher.

Finally, the impact of inaccurate link quality prediction on the performance of channel-aware packet schedulers is studied. Because link quality prediction errors sometimes cause an erroneous behaviour of adaptive rate schemes, they can also affect the performance of packet schedulers that use the adapted transmission rate in the calculation of the scheduling metric. Using a link quality prediction that assumes the link does not change over the next packet's duration—the best prediction according to previous results—the effects of the link quality prediction errors in the user perceived quality obtained by wireless packet schedulers is evaluated.

1.3 Organisation of the Thesis

The next chapter introduces several subjects of relevance for the problems addressed in this thesis. These include the behaviour of a wireless link, the wide variety of available applications and their different characteristics. Chapter 3 introduces the general scenario and the main question to be handled and presents the packet scheduling approach proposed to improve the overall QoS experienced by users in a wireless TDM/TDMA cell. Chapter 4 describes literature related to the problem and relevant state-of-the-art solutions to similar problems. Chapter 5 describes the details of the methodology used for the evaluation of the utility-based scheduler. Chapter 6 describes the functionality of the scheduler and details of its parameterisation, presenting at the end a comparison of different scheduling approaches regarding cell throughput and delivered QoS. Chapter 7 presents a comparative study of heuristics for the prediction of the wireless link behaviour based on past information. It also includes a study of the effects of inaccuracies in the performance of an adaptive transmission rate scheme. Chapter 8 focuses on how the inaccuracies in prediction of link behaviour affect the performance of the schedulers studied in Chapter 6. Finally, Chapter 9 summarises the thesis and presents directions for future work.

Chapter 2

Background

The attenuation that a signal suffers varies in time and from place to place. The amplitude and speed of the variations depends on the geometry and mobility of the environment, sender and receiver, as well as on their relative positions. The next sections give an overview of the different phenomena that influence the propagation of electromagnetic waves and present models of those phenomena.

The influence of the physical phenomena on the communication expresses itself as transmission errors, as explained in Section 2.2. The time-variable characteristic of the error patterns produced by the propagation of electromagnetic waves presents many challenges to communications. To answer them channel-adaptive techniques/mechanisms were designed, which can adapt their behaviour to the link conditions. A very common approach that enables great capacity gains is to adapt modulation and channel coding to the behaviour of the wireless link. Section 2.2.2 presents these state-of-the-art techniques and important related literature.

The multitude of available applications have QoS requirements that differ from one to another. Furthermore, some of the applications have typical behaviours that one can take advantage of to improve statistical multiplexing. Section 2.4 characterises Internet applications relevant for this work and describes their requirements and the metrics that shall be used later to evaluate the user perceived performance.

2.1 Wireless Channel

Consider a wireless environment characterised by the existence of several peers which communicate using the wireless channel. The wireless channel is the physical medium used for communication and is shared among all the peers. The communication between two peers depends on the characteristics of the link between those two specific peers—a wireless link is defined as that between a sender and a receiver. So, the wireless channel shared among all communication peers aggregates all wireless links between any two peers.

This section briefly describes the phenomena that influence the propagation of an electromagnetic wave between sender and receiver and how they are modelled. A more detailed description can be found in the technical report [3] and in references.[4, 5]

2.1.1 Path Loss

The power of the received signal decreases with increasing distance between sender and receiver—what is known as path loss. The reason for the free space path loss is that the total power of the electromagnetic wave stays constant but the area over which it is distributed increases with the distance covered. Consequently the power available at the receiving antenna in free space decreases with the square power of the distance to the transmitter. The relation between the received and the transmitted power, the channel gain, due to path loss in free space is

$$h_{\text{path loss}} = \frac{P_R}{P_S} = \left(\frac{\lambda}{4\pi d}\right)^2,$$

since the wave propagates equally in every direction (like a sphere of increasing radius). This is obviously not true for real systems where the earth limits wave propagation downwards and there are objects between sender and receiver or close to the propagation path. The effect of the ground on wave propagation is often modelled by the two-ray model, which considers the direct path and one reflected path (in this case reflected by the earth) to calculate the power of the received signal. Objects in the path may block the wave or let through some of its energy (refraction), and some energy propagates around their edges (diffraction). Also, some objects around the propagation path reflect the wave (reflection) or scatter it depending on the surface of the object. The total power arriving at the receiver antenna is the sum of the power of the several waves arriving there simultaneously: the direct one in case of line-of-sight (LOS) to the sender, plus the waves diffracted, reflected and scattered by objects around the path.

Given a certain transmitter and receiver, and the details of the objects in the area, it is possible to calculate all possible paths between the sender and receiving antenna; the attenuation suffered by the wave on each path, the delay, phase and frequency shift can also be calculated depending on the phenomena that affected the wave in that path; then all multipath components can be added to build a deterministic model of the link. This is the procedure used by ray-tracing models, which need very detailed information about the terrain and objects (buildings, trees, etc) and are computationally very demanding. As a consequence, they are impractical and statistical channel models are usually used instead.

Empirical and semi-empirical models have been developed to calculate the path loss between a sender and a receiver in specific environments and for specific frequencies. The empirical path loss models are based on extensive measurement campaigns in different environments and the semi-empirical path loss models on a mix of empirical and theoretical data. The models are usually of the form:

$$h_{\text{path loss}} = \frac{P_R}{P_S} = K \cdot \frac{1}{d^{\alpha}} \tag{2.1}$$

or, in dB,

$$h_{\text{path loss}}[\text{dB}] = \frac{P_R}{P_S}[\text{dB}] = 10\,log(K) - 10\ \cdot \alpha \cdot log(d),$$

where the constants K and α are fitted to measurement results according to the areas under consideration. For every new area, calibration measurements are required to calculate correction factors for the general models. The factor K depends on the frequency used, as well as the height of the transmitter and receiver antennas. The distance d is expressed in units of a reference distance and is defined together with the exponent, α, which agglomerates the different propagation effects. The most popular models are the Lee [4] and Hata models (which have several extensions and variants), and an overview of these and others can be found in [6]. Further, several sets of measurement are available in the literature for the calibration of existing models for specific environments. For example [7, 8] suggests $\alpha \in$[2.1:3.8] dB in urban environments at 900 MHz for a reference distance of 100 m. Reference [9] measured the following parameters for a wide range of environments at 5.3 GHz for a reference distance of 1 m:

Environment	Model Type	Used antenna height (m)	α	$10 \cdot \log K$ [dB]
Urban	LOS	4	1.4	58.6
Urban	non-LOS	4	2.8	50.6
Urban	LOS	12	2.5	35.8
Urban	non-LOS	12	4.5	20.0
Urban	LOS	45	3.5	16.7
Urban	non-LOS	45	5.8	-16.9
Suburban	LOS	12	2.5	38.0
Suburban	non-LOS	5	3.4	25.6
Rural	LOS	55	3.3	21.8
Rural	non-LOS	55	5.9	-27.8

2.1.2 Shadowing

The model presented in the previous section enables the calculation of the path loss for a fixed distance between sender and receiver. However, the received power measured at different locations which are all at the same distance from the sender shows statistical variations as a consequence of location-specific propagation phenomena like reflections (e. g. on buildings and ground), diffraction (e. g. around buildings), refraction (e. g. through walls or windows), scattering (e. g. on buildings, trees or ground). Averaging many received power values for the same distance yields the path loss from the calculations. Thus, for a given fixed distance, frequency and transmission power, the actual received power varies around the deterministic pathloss value depending on the different objects in and around the signal path. These stochastic, environment dependent variations are called shadowing. Shadowing is not a physical phenomena like reflection or diffraction, but a designation for the statistical variations observed in the path loss. So, shadowing reflects the differences between the measured received power and the deterministic path loss obtainable from Eq. 2.1.

The variations of the measured power relative to the deterministic path loss value were measured for a variety of environments and distances and its distribution was found to be log-normal with 0 average [10]. The shadowing variations of the path loss can therefore be calculated from the log-normal distribution

$$p_{h_{shadowing}}(a) = \frac{1}{\sqrt{2\pi} h \sigma_{shadowing}} e^{-\frac{\log h^2}{2\sigma^2_{shadowing}}} \tag{2.2}$$

and superimposed on the path loss calculated from the expression in Eq. 2.1 to obtain the gain

$$h = h_{pathloss} \cdot h_{shadowing}. \tag{2.3}$$

$\sigma_{shadowing}$ usually takes values in the interval [5:12] dB [11], depending on the communications system used. For cellular communication networks a value of 7–8 dB [12, 4] is usually used. More values of the standard deviation of the path loss obtained from measurement campaigns can be found in the literature; for example, in [8, 7] for European cities at 900 MHz, in [13] for Tokyo at several UHF frequencies, and in [9] for different outdoor environments at 5.3 GHz.

Environment	Model Type	Used antenna height (m)	$\sigma_{PL_{SF}}$ [dB]
Urban	LOS	4	3.7
Urban	non-LOS	4	4.4
Urban	LOS	12	2.9
Urban	non-LOS	12	1.7
Urban	LOS	45	4.6
Urban	non-LOS	45	2.8
Suburban	LOS	12	4.9
Suburban	non-LOS	5	2.8
Rural	LOS	55	3.7
Rural	non-LOS	55	1.9

Because the propagation paths to two locations close to another suffer correlated effects, values of $\sigma_{shadowing}$ at close locations are correlated. Based on measurements, [14] proposes a negative exponential correlation with distance

$$R_{h_{shadowing}}(d) = \sigma^2_{shadowing} \cdot \epsilon_D^{d/D}, \tag{2.4}$$

where d is the distance between two points and ϵ_D is the correlation between two points at a reference distance D. The parameters of the model—ϵ_D and D—should be fitted to measured values. For suburban environments at 900 MHz $\sigma_{shadowing}$ was estimated to be 7.5 dB and $\epsilon_D = 0.82$ for a distance of 100 m. In urban environment at 1700 MHz the estimated model parameters were $\sigma_{shadowing} = 4.3$ dB and $\epsilon_D = 0.3$ and $D = 10$ m.

2.1.3 Flat Fast Fading

As mentioned above, due to multipath propagation, several waves with different power, phase, and delay arrive at the receiver instead of a single one. The delay between the first and last arriving waves, delay spread, compared to the inverse of the signal bandwidth determines how strongly a signal can be distorted by multipath. If the delay spread is low compared to the inverse of the bandwidth then the distortion is insignificant—flat fast fading. Otherwise the multipath components from a pulse can interfere with the following pulse causing inter-symbol interference (ISI), in which case the channel is frequency selective. This section deals only with flat fading, i.e. with the case where the delay spread is low compared to the inverse of the bandwidth and the transmission gain can be considered constant over the whole signal bandwidth.

Due to movement in the environment, from the receiver or transmitter or objects in and around the propagation path, the reflectors and scatterers that originate the multipath components change in time. This results in multipath components that vary in time and lead to time variant fast fading. Furthermore, in the presence of movement each multipath component suffers a frequency shift—Doppler shift. These variations occur in a much smaller timescale than the variations of the pathloss or shadowing, hence the designation fast fading.

A model of the complex received signal $r\bar{(t)}$ can be written as

$$r(t) = \sum_i H_i \cdot e^{2\pi j \cdot cos(\gamma_i) \cdot t \cdot f_d} \cdot s\left(t - \tau_i\right) = \sum_i H_i \cdot e^{2\pi j \cdot \nu_i \cdot t} \cdot s\left(t - \tau_i\right), \quad (2.5)$$

where

$H_i = h_i e^{\phi_i}$ is the complex gain of multipath component i, with h_i the amplitude and ϕ_i the phase shift, which depends on the delay of the path;

$f_D = \frac{f_C}{c} v = \frac{v}{\lambda}$ is the Doppler frequency, with f_C the carrier frequency, c the speed of light, v the speed of movement and λ the wavelength;

$\nu_i = f_D \cdot \cos\gamma_i$ is the Doppler shift on multipath component i, obtained by projecting the Doppler frequency from the direction of movement on the direction of incidence of the multipath component through the angle γ_i;

τ_i is the delay of the i-th multipath component.

Since only flat fading is considered, τ_i can be neglected and the received time variant complex signal can be written as

$$\bar{r}(t) = s\left(t\right) \cdot \sum_{\forall i} H_i \cdot e^{2\pi j \cdot \nu_i \cdot t}. \quad (2.6)$$

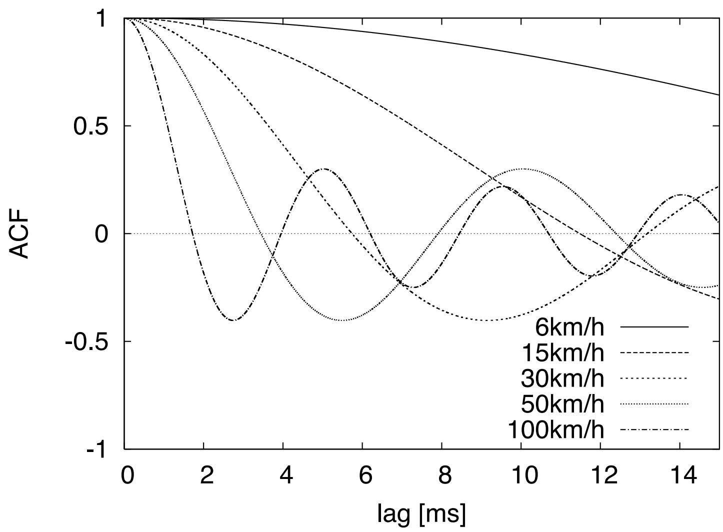
Using this equation, the characteristics of the received signal for the case of a harmonic unmodulated transmitted signal $s\left(t\right) = e^{2\pi f_C t}$ can be studied (a detailed derivation can be found in [5, 15]). Assuming no LOS component and an uniform scattering environment (angle of arrival of the multipath components is evenly distributed in $[0; 2\pi]$), the complex received signal $r(t)$ is composed of uncorrelated, zero-mean jointly Gaussian real $r_I(t)$ and imaginary $r_Q(t)$ components. The probability density function (PDF) of the amplitude of the complex received signal $|r(t)| = \sqrt{r_I^2(t) + r_Q^2(t)}$ is given by the Rayleigh distribution

$$p_{|r|}(|r|) = \frac{|r|}{\sigma_r^2} \cdot e^{-\frac{|r|}{2\sigma_r^2}} = \frac{2|r|}{P_R} \cdot e^{-\frac{|r|}{P_R}} \quad (2.7)$$

where $\sigma_r^2 = \frac{1}{2} P_R$ and P_R is the average received signal power. This can be obtained as $P_R = h \cdot P_S$ from equations 2.1 and 2.3. The power of the received signal $|r(t)|^2$ is exponentially distributed

$$p_{|r|^2}(|r|^2) = \frac{1}{2\sigma_r^2} \cdot e^{-\frac{|r|^2}{2\sigma_r^2}}. \quad (2.8)$$

Figure 2.1: The autocorrelation of the received signal on a channel subject to Rayleigh fading is proportional to the 0th-order Bessel function. The plot shows the ACF for a carrier frequency of 2.4 GHz and several environment speeds for time lags up to 15 ms.



In case there is a LOS multipath component, the real and imaginary part of the complex received signal amplitude are no longer zero-mean. Instead, their means are the real and imaginary parts of the LOS complex multipath component and the PDF of the amplitude of the complex received signal is given by the Rice distribution.

$$p_{|r|}(|r|) = \frac{2|r|(K+1)}{P_R} \cdot e^{-K-\frac{|r|^2(K+1)}{P_R}} I_0 \left(2|r|\sqrt{\frac{K(K+1)}{P_R}} \right), \tag{2.9}$$

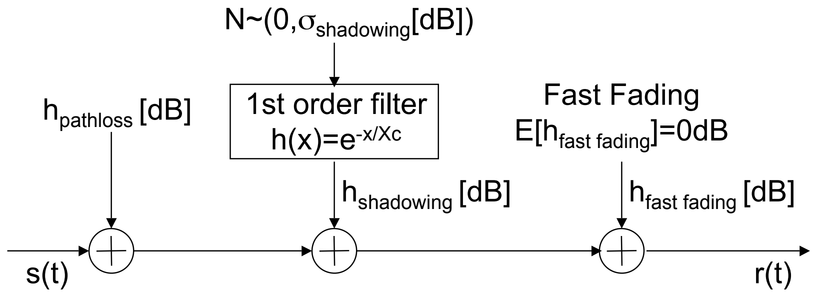
where K is the ratio between the power of the LOS and the power of the non-LOS components, and P_R is the average power of the received signal. In this case, the received power is χ^2-distributed. The derivation of the expressions and further details can be found in references [5, 15].

Another important property of the amplitude of the complex received signal is its time-correlation, which is a consequence of the multipath propagation environment. Under the assumption of no LOS component and an uniform dense scattering environment, the autocorrelation of $r(t)$ is proportional to the 0-th order Bessel function: the signal de-correlates for a distance of $\lambda/4$ (or the equivalent time), but is highly correlated for $\lambda/10$. Figure 2.1 shows examples of the autocorrelation of the received signal for several speeds and a carrier frequency of 2.4 GHz. For pedestrian speeds (6 km/h) the autocorrelation is higher than 0.9 up to 7.66 ms, but for increasing speeds the time lag over which the received signal is strongly correlated decreases and for 50 km/h the autocorrelation lies below 0.9 for a lag of only 0.91 ms. On one hand, a strong correlation for long lags is advantageous since the variations of the received signal can be predicted more accurately. However, on the other hand, if the received signal is correlated over long periods, in case the received signal is low, it takes longer for it to improve.

2.1.4 Putting It Together: A Statistical Model of a Wireless Link

The previous sections introduced statistical models for the effects of different propagation phenomena that occur at different timescales. Path loss and shadowing depend on the movement of the scatterers around the propagation path or the relative position between these and the receiver, whereas variations due to multipath fading occur in timescales of the order of fractions of a wavelength.

Figure 2.2: Stochastic model of the wireless link.



The models for the different phenomena can be superposed to build a stochastic model of the wireless link and of the received signal, as shown in Figure 2.2. The received signal $r_(t)$ is an attenuated version of the transmitted signal $s(t)$, whose amplitude varies over timescales of a fraction of a wavelength according to the fast fading distribution around a value determined by $h_{\text{pathloss}}[dB] + h_{\text{shadowing}}[dB]$. The value of $h_{\text{shadowing}}$ is picked from a normal distribution with 0 mean value (in dB) with an exponential time correlation depending on the speed of the receiver or of the environment. This can be achieved by passing the normal distributed values by a first-order low-pass filter with a pole at $z = \epsilon_D^{vT/D}$, where ϵ is the correlation at the reference distance D, v is the speed and T the sampling interval of the process [14]. The value h_{pathloss} is determined using Eq. 2.1 and the distance between sender and receiver.

Figure 2.3-a shows an example of the received signal subject to shadowing variations around the deterministically calculated pathloss. Figure 2.3-b shows an example of the received signal when multipath fading is superimposed on the pathloss and shadowing models.

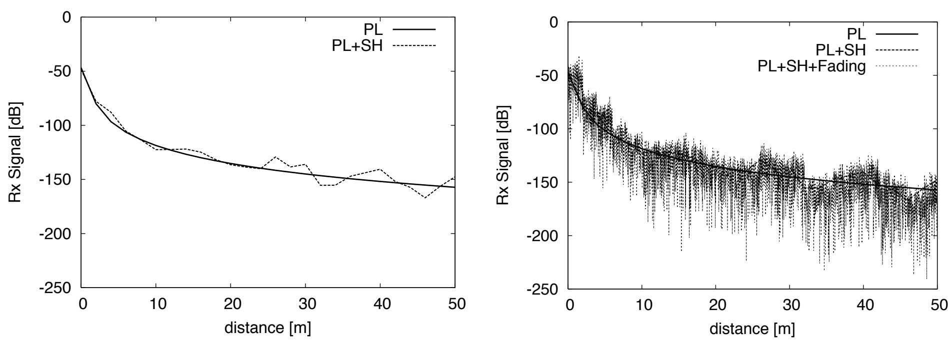
The implementation of the multipath fading model is not as straightforward as modelling the other propagation phenomena. A widely used implementation is a deterministic sum of cosines, also known as the Jakes model [16], which was used to generate the received signal seen in the examples. Another possibility, suggested in [17], is to generate two independent gaussian processes, pass them through linear filters to achieve the desired correlation properties, and treat them as the real and imaginary components of the complex amplitude of the received signal (see Section 2.1.3). A further Rayleigh fading generator using table lookups is proposed in [18].

2.2 Digital Communications

2.2.1 Communication errors

The transmitted signal $s(t)$ in the previous section is used to convey data to the receiver, which tries to recover the information from the received signal $r(t)$. At the transmitter, digital data modulates a carrier where a pulse is used to convey one information symbol which can consist of one or more bits depending

Figure 2.3: Example of modelled received signal subject to the different propagation phenomena: pathloss (PL), shadowing (SH) and multipath fading.



(a) (b)

on the modulation used. At the receiver, the received signal is converted back into a bit sequence—a process known as demodulation. The received signal is a weaker and distorted version of the transmitted signal and has added noise, which can lead to errors in the identification of the transmitted symbol and consequently to communication errors—the received bit sequence is not equal to the transmitted bit sequence.

The demodulation can be separated into two phases: the demodulation itself, and the detection. The digitally modulated signal can be seen as a linear combination of signals, which form an orthonormal base of a vector space. During each symbol period, the demodulator decomposes the received pulse into a vector in that space. This vector contains the value of the projections of the received waveform onto the signals in the orthonormal base of the vector space. This can be achieved either by a matched filter or a correlator [19], depending on the type of signal being demodulated. The detector compares the vector output from the demodulator with the values of the M possible waveforms. The distances in the vector space are compared and the "closest" possible waveform is chosen so that the probability of making a wrong decision is minimised.

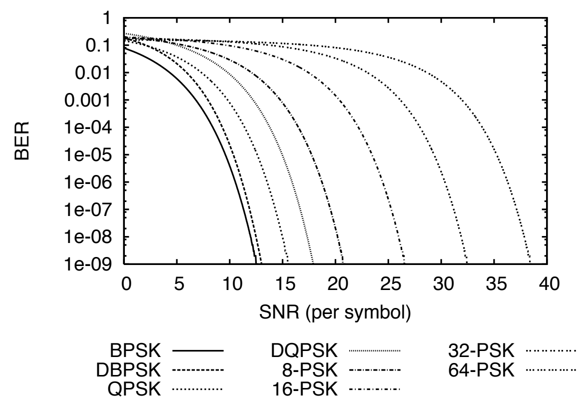
The received signal is thus converted into a sequence of bits and there is a probability that the sequence obtained differs from the originally sent bit sequence, due to the probability that the sent waveforms are falsely identified. This, in turn, can happen because the received signal is distorted and weaker and has added noise and interference, which can transform a waveform into another from the point of view of the demodulator. The probability of making a wrong decision depends thus on the power of the received signal relatively to the noise—the signal to noise ratio (SNR). For each symbol, the probability of making an error in demodulating it and the consequent bit error probability can be calculated depending on the modulation and detector used. Details of the derivation of expressions for digital modulations assuming a time invariant channel with additive white gaussian noise (AWGN) at the receiver can be found in [15, 19]. Table 2.1 shows the expressions for the probability of a bit error as a function of SNR per symbol, γ, for M-ary Phase Shift Keying (PSK) modulations, as they are of relevance for this thesis. Figure 2.4 plots the bit error rate (BER) for those modulations as a function of the SNR per symbol: for a same received SNR, the more bits $k = \log_2 M$ are carried in a symbol, the higher the probability that bit errors occur, i. e. higher order modulations are less robust againts errors. So, although higher data rates can be achieved by using high order modulations, the probability that transmission errors occur is also higher. Thus, those modulations should only be used when the received SNR is high.

Transmission errors can be detected or even corrected (with high probability) if coding is used. Redundant information is transmitted, which can

Table 2.1: BER as a function of the SNR per symbol γ for PSK modulations.

Modulation	Bit Error Probability P_b	Comments
BPSK	$Q\left(\sqrt{2\gamma}\right)$	—
DBPSK	$\frac{1}{2}e^{-\gamma}$	—
QPSK	$2Q\left(\sqrt{\gamma}\right)\left[1-\frac{1}{2}Q\left(\sqrt{\gamma}\right)\right]$	—
DQPSK	$Q\left(a,b\right)-\frac{1}{2}I_0(ab)\exp^{-\frac{1}{2}(a^2+b^2)}$	$a=\sqrt{\gamma(1-\sqrt{\frac{1}{2}})}$, $b=\sqrt{\gamma(1+\sqrt{\frac{1}{2}})}$
M-PSK	$\approx 2Q\left(\sqrt{2\gamma}\sin\frac{\pi}{M}\right)$	—

Figure 2.4: BER as a function of the SNR per symbol for PSK modulations.



be used to verify whether transmission errors occured and sometimes correct them. Thus, at the cost of some overhead, the BER for a certain SNR at the receiver can be reduced by using channel coding—usually called the coding gain. Channel coding shifts the curves in Figure 2.4 to the left by an amount equal to the coding gain so that the BER at a certain SNR is improved.

The performance of coding is usually limited in case errors do not occur independently, as in AWGN channels, but in bursts, as on wireless links due to the correlated fading. To account for this interleaving is used to separate error bursts and thus increase the performance of channel coding.

As data is usually organised in packets, the packet error rate (PER) is a performance metric of interest and can be obtained from the BER assuming that the bit errors in the packet are independent, an acceptable assumption due to the use of interleaving:

$$\mathrm{PER} = 1-\left(1-\mathrm{BER}\right)^L, \tag{2.10}$$

where L is the length of the packet.

2.2.2 Adaptive modulation and channel coding

It was seen in the previous sections that the wireless link is time-variant due to mulitpath fading (see Figure 2.3-b) leading to a time variant received signal and a time variant received SNR. Consequently, the BER of communications over a wireless link is not constant but varies in time with the SNR, making it difficult to design an efficient system. For any modulation and coding scheme (MCS) there are times when the link is very good so that the BER is very low—good link quality, but there are also times when the link is bad leading to very high BER and many communication errors—bad link quality. On one hand, if a MCS is chosen for taking advantage of the high data rates achievable during periods of good link quality, it will lead to very bad performance in the periods of bad link quality. On the other hand, if a robust MCS is chosen so that the BER can be kept low during bad link periods, the channel will be inefficiently used during good link periods. Under these conditions and taking into account that wireless bandwidth is an expensive and scarce resource, using a static, pre-defined modulation and coding scheme is inefficient and it is more sensible to adapt the MCS to the variations of the link quality.

Adaptive MCS consists of two independent steps: predicting the behaviour of the wireless link and adapting the MCS to the predicted link behaviour. Some MCS adaptation algorithms [20, 21] increase the transmission rate sequentially, using MCS that carry more bits per symbol, as successful transmissions are acknowledged; the rate is decreased as packet errors occur by using more robust MCS (that carry less bits per symbol). In these cases, the link quality prediction is implicit in the positive or negative acknowledgments and the adaptive algorithm is reactive. But many adaptive MCS algorithms trust that a prediction of the SNR at the receiver is available and use it to choose the MCS, usually based on pre-defined SNR thresholds. The choice of the thresholds for the adaptation is usually targeted at not exceeding a certain BER using the available MCS [22, 23, 21], but can also be designed to maximise spectral efficiency [24]. Adapting MCS to wireless link quality is very important to improve the efficiency of bandwidth usage (usually called spectral efficiency and measured in bit/s/Hz) and is nowadays state-of-the-art in most wireless systems: reference [25] describes the adaptive MCS schemes of several cellular systems; the 802.11b [26] and g [27] standards also extend the original standard for Wireless Local Area Networks (WLAN) [28] to support adaptive modulation, although the adaptation algorithm is not defined.

Despite the recognised importance of adaptive MCS schemes to the efficiency of wireless communications and the fact that their performance relies on link quality prediction, the latter has not been thoroughly studied, neither have the effects of its inaccuracy in the adaptive MCS. This is the object of study in Chapter 7.

2.3 Link-State Aware Packet Transmission

It was seen in the previous section that the characteristics of a wireless link vary with time and can cause severe impairments to wireless communications. Link layer retransmissions are a solution to improve the reliability of the wireless channel. But concerns about efficient use of available bandwidth and power consumption of wireless devices drove the appearence of a scheme that defers transmissions if they are likely to suffer errors. The link-aware transmission scheme proposed in [29] transmits a packet only if the link is predicted to be in a good state, i.e. if there is little probability that the packet suffers transmission errors. If, on the other hand, the link is predicted to be bad, the packet transmission is deferred until the link quality improves.

The work in [30] extends the link-aware packet transmission above to a wireless channel shared by several Wireless Terminals (WT)—a multiuser scenario. When there are several WT and their locations are independent, if one WT experiences a bad link there might be another one experiencing a good link. In this case it is beneficial to transmit to the second WT—transmission schemes that use this approach are known as channel-aware schedulers. The use of channels instead of links here is inconsistent with the terminology used in this work, however an exception will be made due to the widespread use of the term channel-aware scheduling.

2.4 Typical Applications and Their Quality of Service

The previous sections dealt with the phenomena that influence the propagation of electromagnetic waves, the time-varying characteristics of the signal received across a wireless link and their effects on wireless communications. This section describes the applications present in the traffic mix used in this thesis and whose different characteristics and requirements illustrate the challenge of supporting such a wide range of applications.

The availability of wireless access lead to the widespread use of the Internet and to the deployment of new applications, changing the characteristics of the traffic mix offered to communication systems. Traditional Internet applications—Web browsing, email, file transfer and remote login—transmit chunks of data (an HTML file, an email, any kind of file or a command and its arguments) where data is useful only if correctly received and after the whole data chunk is received. This data is carried by the Transport Control Protocol (TCP) [31] which guarantees error-free and in-order delivery of the data. New applications are often audiovisual, real-time applications like video on demand, VoIP, teleconferencing, telemetry, remote surgery, games [32, 33]. These applications differ from traditional Internet applications in that they use the data as it arrives instead of waiting until the whole data is received before using it, and are thus sensitive to the end-to-end delay and its variations—known as jitter. Furthermore, these applications have packet delivery requirements that differ from one-another, having different delay tolerances [34], some being more sensitive to jitter than others, and some tolerating some amount of packet losses (audiovisual applications) while others do not (telemetry, gaming or remote machine control) [33]. Supporting this new mix of applications raises new challenges to communication systems, which were designed to deliver best-effort service.

From the wide range of new applications, this thesis shall further consider only audiovisual applications like VoD and VoIP, which shortly buffer incoming data before displaying it while more data is arriving in the background. Also, because humans tolerate a certain amount of errors without perceiving bad quality, these applications can tolerate a certain amount of packet loss and still function without extreme service degradation. Audiovisual applications mostly use the User Datagram Protocol (UDP) as a transport protocol, aided by session and presentation protocols like RTP, RTCP [35], RTSP [36].

The next sections describe details of the two types of applications, TCP-based and audiovisual, describing their characteristics and requirements, as well as some proposed solutions for their support over wireless links. Further, the next sections also introduce and describe the metrics that shall be used to evaluate the user perceived quality of the applications used in this work.

2.4.1 TCP-based applications

Traffic statistics for the Internet backbone from years 2001 and 2002 [37] [1] show that around 90% of the packets and 95% of the bytes were carried by the TCP transport protocol. According to those statistics, the TCP traffic mix consisted of diverse applications, the most important being WWW traffic (45–65%), file transfers, Email, Telnet and SSH, whit the latter two making up less than 1% of all TCP flows. These applications can be characterised by a server-client behaviour in the sense that data exchange obeys the following pattern: a client starts a connection by sending a data request to a server that answers by sending a certain amount of data, the response, in the opposite direction, after which the client can send another request or close the connection. Although this behaviour is common to the applications mentioned, they differ in the

[1] Unfortunately more recent statistics are not available.

absolute and relative size of the requests and responses. For Telnet and SSH the requests and responses consist of character inputs and their echoes, which are approximately of the same size, and the applications are very interactive; for WWW traffic the responses are much larger than the requests and the applications are also interactive, although less than the previous ones; for email and file transfers the requests are short and the responses vary much in size, and the degree of interactivity is much lower than that of the other applications.

TCP has several mechanisms to provide reliable end-to-end service and congestion control, like retransmissions, slow start and congestion avoidance [38], but these were developed with wired networks in mind, where packet losses are usually a consequence of network congestion. Therefore, a lost packet, besides being retransmitted, sets off several mechanisms destined at congestion control. This reduces the transmission rate and enables TCP to adapt to variable bandwidth, which is why it is often classified as an elastic application. However, when the end-to-end path includes a wireless link, packet losses occur more often than in wired networks and are not necessarily a sign of network congestion, but a consequence of link errors. Setting off the congestion control mechanisms as a reaction to link errors greatly reduces the efficiency of the communication [39, 40, 41]. New versions of TCP include fast retransmit and fast recovery [38] to reduce the throughput degradation when segment losses are not a consequence of network congestion. In a comparison of the different TCP versions over a lossy link [39] TCP NewReno (with fast retransmit and fast recovery) shows the best performance under different conditions, proving to be the best version to be used on wireless systems. As such, it is the TCP flavour used in this work.

Requirements

Contrary to the opinion of Shenker et al. [32], different TCP-based applications awake in their users differing expectations according to the degree of interactivity [42, 34, 33, 43]. The response times for Telnet (or SSH) echoes are expected to be much shorter than the arrival of a Web page (less than 150 ms and less than 5 s, respectively, according to [43]). A file download is usually a background process due to the much larger amount of data to be transmitted, which is why a large waiting time is tolerated and the degree of interactivity is very low. These different user expectations lead to differing QoS requirements of the TCP-based applications and to their different classification as background or interactive TCP traffic: Telnet, SSH and Web traffic belonging to the latter and file transfers to the first. As a consequence of the different expectations and QoS requirements, TCP-based applications should not all be equally handled by the network [43], and some degree of differentiation should be provided, since short TCP-connections suffer more from packet losses than long ones. This shall be taken into account here by differentiating between Bulk data transfer and WWW traffic.

Mechanisms for Support of TCP on Wireless Links

There have been several proposals of methods to improve the throughput of TCP over wireless links using mechanisms located beyond TCP [44, 45, 41, 46]. In a comparative study [44] of such solutions it is found that the greatest performance improvement is obtained by avoiding coarse retransmission timeouts, which can stall transmission for long periods of time. This is best achieved by selective link layer retransmission of the lost packets, which are much faster than TCP retransmissions, and by suppressing duplicate acknowledgments, which prevents setting off slow start and congestion avoidance mechanisms and is of great relevance for large bandwidth-delay products. Furthermore, since the loss of SYN or SYN+ACK packets leads to extremely long timeouts, which affect especially short TCP connections, these packets should be explicitly protected against losses [43]. In this work, link-layer retransmissions shall be used for TCP-based flows.

Metrics

As explained above, TCP has several mechanisms to provide reliable service that influence its behaviour over error-prone links, like retransmissions, congestion control, and flow control. So, the performance of applications that use TCP should be measured taking the effects of those mechanisms into account. Bulk Transfer Capacity (BTC) has been defined as "a measure of a network's ability to transfer significant quantities of data with a single congestion aware transport connection" [47]. It is credited to produce results more relevant for applications than other bandwidth measurement techniques [48], thus it shall be used to measure the performance of TCP-based applications. The BTC is the "long term average" rate of data transmitted on a single connection per unit time [47] and is, therefore, a measure of the goodput seen by the application. Details of the calculation of the application goodput for Bulk file transfer and WWW traffic are given in Chapter 5.

2.4.2 Audiovisual Applications

Audiovisual applications are applications whose data consists in audio or video or a combination of both, like VoIP, VoD, video conferencing, and audio streaming. These applications differ from TCP-based applications in the characteristics of the data, the traffic pattern, the way in which the users experience the service delivered and, consequently, the requirements on the network.

Audio and video are continuous, analog data that is sampled, digitised and encoded (with or without losses) before being transmitted over a network, resulting in a regular sequence of audio or video frames. In the context of audiovisual applications, a frame is the output of the codec, corresponding to an audio or video sample; a packet refers to the data aggregation seen by the network and can contain part of a frame (usually the case for video due to the large frame sizes), one frame, or several frames (usually the case for audio due to the very short frame size).

Audiovisual traffic is usually carried by RTP over the User Datagram Protocol (UDP) [49]. The latter provides connectionless delivery of the data since timing is more important than reliability, while RTP provides the means to support functions relevant to the support of audiovisual requirements like detection of frame losses and re-sequencing (through the use of sequence numbers), distinction among frames using a marker, identification of the syntax and semantic of the codec, and synchronisation of different media types. There is also a timestamp of packet generation that can be used to calculate timing relationships like jitter. The functionality provided by RTP enables the implementation of mechanisms for packet differentiation, codec rate adaptation, and playout buffer adaptation. Interactive audiovisual applications need to negotiate at the start several session-level parameters, like the codec to be used, the data rate, or the ports to use. It is also important to know whether the communication party is there to "pick up the phone". Furthermore, some parameters (e. g. the codec) can be changed during a session but both communication parties must agree on it so that it is necessary for a session level protocol to coordinate the connection setup and break-down and the negotiation of parameters. The Session Initiation Protocol (SIP) and the H.323 have been developed for this end: they provide a connection-oriented session to audiovisual applications. So, nowadays there are protocols that provide the necessary means to implement the mechanisms mentioned in the previous section and which provide for good performance of audiovisual applications over communication networks with variable bandwidth conditions.

Requirements

The user of an audiovisual application expects a continuous, ordered display of the data, regardless of whether the application is interactive like a VoIP call or streaming like VoD, i. e. there is a deadline for the arrival of a frame at the receiver beyond which the user will notice degradation of application performance. This is why audiovisual is real-time data. Furthermore, there is a bound on the delay variation of the different frames—jitter—beyond which the user will see interruptions and jumps in the displayed media. However, audiovisual applications are tolerant to frame losses, as long as these are not strongly time-correlated [50, 51, 52], because the human eye and ear are not perfect and do not notice a frame loss. Furthermore, nowadays most codecs are capable of error concealment—covering up missing frames using information in adjacent frames—which increases the quality in the presence of isolated losses. Thus, ordered delivery, bounded end-to-end delay and jitter with some tolerance to packet losses are the characteristic requirements of audiovisual traffic.

Mechanisms for Support of Audiovisual Communication

Because audiovisual data should be displayed continuously, but packets traversing a network always suffer delay variations, audiovisual applications have a buffer where incoming frames are stored before being played to the user—the playout buffer. By buffering frames and delaying their play time, jitter can be smoothed out as frames that are too fast can be buffered and frames that are too slow may still arrive in time since the previous frames which were saved in the buffer are still being played. The longer the playout buffer the larger delay variations remain unnoticed, but at the cost of a larger buffer capacity and a larger end-to-end delay. Conversely, if the playout buffer is short less jitter can be accounted for and more frames will be dropped. Due to the larger variations in the service delivered over wireless networks and to avoid too large end-to-end delays that decrease the perceived quality of interactive applications, nowadays playout buffers of audio and video applications are not fixed but adapt the playout time of the frames to the delay variations in the network [53, 54, 55]. Adaptive playout buffering reduces the requirements on network end-to-end delay and delay jitter, or conversely helps increase the application performance for a same network service, thus playing an important role for audiovisual quality.

In contrast to TCP-based traffic, not all frames are equally important for audiovisual traffic: some video or voice frames are more important than others for the quality perceived by the user. For example, for MPEG-4 [56] and H.264 [57] video streams are organised in Groups of Pictures (GOP) which consist of an initial I-frame followed by a number of predictively encoded (P-frames) and bi-directionally encoded frames (B-frames) which all depend on the initial I-frame. In such a case, due to inter-dependencies among the different frame types created by the encoding algorithm, losing an I-frame is worse than losing a P-frame which again is worse than losing a B-frame. For VoIP, the loss of a packet also has different impact on the perceived speech quality depending on whether it contained background noise, a unvoiced/voiced transition or normal speech [58].

The different importance of frames can be used by the network to handle packets of a single flow differently in periods of congestion or when a wireless link becomes a bottleneck due to fading. For example, packets can be dropped (at the source or at routers along the path) according to the importance and inter-dependencies of the frame(s) they carry instead of indiscriminately, as is proposed in [59, 58, 60] for VoIP and in [61, 62] for video. Also, high importance packets can be retransmitted with higher persistency on a wireless link or sent using more redundancy [63, 64] or can be given higher priority in channel access [65].

The different importance of frames belonging to a audiovisual flow can be used not only by the network, but also by the application itself to react to network outages or congestion periods by not sending low priority frames at all, as proposed in [64, 66]. Besides this, applications can also react to variations of the bandwidth available for transmission by adapting the coding rate, provided that there is some feedback mechanism between the receiver or the network and the sender application [67].

In this work some of these mechanisms shall be used for the support of audiovisual traffic. For VoIP, the Moon adaptive playout buffer is used at the receiver in the evaluation of user perceived quality. The VoD-specific support uses a different number of link layer retransmission for each of the video frame types. Details of the specific support given to each audiovisual flow are provided in Chapter 3.

Metrics

The quality of the service delivered to users of audiovisual applications is evaluated by the human ear and eye—perceptual QoS. So, should the performance of audiovisual applications be evaluated using objective or subjective (perceptual) metrics? Objective evaluation using network metrics like packet losses and packet delay distributions has the advantage of being reliable, comparable and reproducible, but there is little guarantee that the results correlate with the evaluation made by humans. There are methods defined for human subjective evaluation of audio [68, 69] and video [70], where many users rate the perceived quality of media samples according to the Mean Opinion Score (MOS) [68, 70] in a scale of 1 (unsatisfactory) to 5 (excellent). But subjective evaluation is very time-consuming, unreliable and difficult to reproduce, although it expresses the performance actually perceived by the users.

It is nowadays widely accepted that the quality of the service delivered by a network should be ultimately evaluated by user satisfaction [32, 34, 71, 64]. Accordingly, the performance of audiovisual applications shall be evaluated using metrics that predict human quality rating—quality models—and are strongly correlated with the subjective quality assessed by humans in subjective tests. The next two paragraphs describe widely accepted metrics for the perceptual evaluation of VoIP and video that shall be used in this work.

Voice over IP—PESQ, E-model The Perceptual Assessment of Speech Quality (PESQ) standardised by the Telecommunication Standardization Sector of the International Telecommunications Union (ITU-T) [69] is an algorithm than evaluates the quality of speech in terms of MOS by comparing an original sample with its degraded version, obtained after transmission and decoding. PESQ takes into account the coding distortion, mean packet loss rate, delay variations, single packet losses and switching the coding mode, but it does not account for the absolute end-to-end delay, which does not affect speech quality. The E-model is also standardised by the ITU-T [72] and is a planning tool for interactive voice communications. It takes into account the additive effect of several impairments like circuit, room and device noise, quantisation distortion, end-to-end delay and coding distortion, where the latter includes the effects of random packet losses. The result is measured in terms of the R-factor in a scale of 0 (worst) to 100 (best), where 70 is the toll quality of standard telephony. Neither of these models takes all network and coding factors into account in the calculation of the quality of a VoIP call, so Hoene et al. proposed in [73] a method that uses the PESQ model to calculate the speech quality and then feeds it into the E-model together with the end-to-end delay to calculate the final R-factor of the VoIP call. This resulted in a model for evaluation of the perceived quality of interactive speech. The details of the calculation of the R-factor will be described in Section 5.5.3.

Video—PSNR, DIV The Peak Signal to Noise Ratio (PSNR) [74, 75, 76] is the most widely used metric of video quality and has a high correlation with the user perceived quality. It was originally used for the evaluation of picture compression algorithms and is a measure of the difference between two pictures. The PSNR is the mean squared error, pixel per pixel, of a decoded picture in relation to the original picture, and can be written as

$$\mathrm{PSNR} = 10 \cdot \log_{10} \left(\frac{V_{max}}{\mathrm{MSE(d,o)}} \right), \tag{2.11}$$

where V_{max} is the maximum possible energy in the picture (and is equal to the sum over all pixels of the square of the colour depth), and MSE(d,o) is the Mean Squared Error (MSE) between the decoded (d) and original (o) picture.

The PSNR can be mapped to the subjective quality metric Mean Opinion Score (MOS) through the following heuristic mapping [76]

PSNR [dB]	MOS
>37	5 (Excellent)
31–37	4 (Good)
25–31	3 (Fair)
20–25	2 (Poor)
<20	1 (Bad)

For a video, the PSNR is a time-dependent metric which is calculated for every frame. The values of PSNR are usually in the interval 30–40 dB for a video which is encoded and decoded [62]. Although the correlation between MOS and PSNR depends strongly on the contents of the pictures and encoding [77] this metric is still widely used in the evaluation of picture and video quality due to its simplicity and availability [78]. Also, even though the correlation between the subjective quality estimated by PSNR and the actual MOS can be below 80% in some cases, the PSNR is trustworthy as a mean to compare the quality of two videos in case the same codec is used, i. e. when the PSNR of a video is higher/lower than the PSNR of another video, the subjective quality of the first is actually higher/lower than the quality of the latter [75, 76].

As a video is a sequence of pictures and a MOS value is calculated for every picture, there is a time sequence of MOS values per video. This is an uncomfortable metric, which is difficult to use in evaluations and comparison. The average and standard deviations of the metric could be used, but this metrics would smooth out bursts of bad quality, leading to a high average MOS although the video quality would be unacceptable for a user due to jumping quality levels (it is known that users would rather live with a low quality than with quality variations). So, a new metric is introduced in [62]—the Distortion in Interval (DIV), which counts the amount of received frames per interval that have a MOS lower than that of the frame sent. The video quality is considered acceptable if the DIV stays below a threshold during a certain percentage of the duration of the whole video and is considered unacceptable otherwise. This metric shall be used to evaluate the quality of the VoD streams.

Chapter 3

Utility-based Packet Scheduling: a Cross-Layer Approach for QoS Support of Multiple Flows on a Multi-User Wireless Scenario

Whenever data from more than one flow must to be conveyed over the same channel, it is necessary to define an allocation policy. In a TDMA channel, the shared resource is time and it is important to distinguish between two types of resource allocation: bandwidth and transmission opportunity. The first refers to the allocation of a certain bandwidth on average to a flow and can be defined over shorter or longer periods of time. The latter refers to the chance to use the shared channel for a certain period of time—also referred to as packet scheduling. Packet scheduling is about deciding when each flow can access the channel. It can be used to implement the bandwidth allocation, i. e. the transmission opportunities can be given so that the flows receive the bandwidth that was allocated by a bandwidth allocation policy. However, this work focuses on packet scheduling with a different goal, as will be described later in this chapter.

Section 3.1 introduces the problematic of scheduling over a time shared wireless channel in the general context of real-time systems. Then, Section 3.2 describes the system model and scenario underlying this work and Section 3.3 formulates the problem studied in this thesis. Section 3.4 proposes a novel approach to wireless packet scheduling. Finally, because the solution proposed advocates application-specific link layer support, Section 3.5 presents possible flow-specific policies for support of the flow types used as exemples in this thesis.

3.1 Wireless Packet Scheduling and Scheduling in Real-time Systems

Transmitting packets that compete for a wireless channel with time multiplexing fits into the category of real-time system problems, where the transmission of each single packet is a job and the wireless channel is the shared processor. The release time of a job—the instant when it becomes available for execution—is the arrival of the packet at the scheduling system. The response time is the time between release time and the instant when the job is completed— i. e. the instant when the packet is transmitted. The relative deadline of the job is the maximum allowed response time—in the context of packet scheduling the relative deadline of a job is just called the deadline of the packet.

Relative deadlines represent time constraints to the behaviour of the system and can be classified as hard or soft deadlines—that is, the jobs are classified according to the classification of their deadlines. Although there are several, mostly qualitative, definitions of hard and soft deadlines [79], the more relevant for this work is based on tardiness and usefulness of the results for system performance. Tardiness measures the completion of a job relative to its deadline, being 0 if the job is completed before or at the deadline, and equal to the difference between completion time and deadline if the job completes after the deadline. A hard job is a job whose usefulness decreases very fast with positive tardiness; the usefulness for system performance of a soft job decreases more slowly. The link layer of a wireless communication system where the packet scheduler resides, is a soft real-time system where jobs have soft deadlines, since it is often better to deliver packets even if they are late as they may still be useful for the receiving application. Further, since the users of audiovisual systems and, more generally, wireless communications do not require proof that the time constraints are met, the quality of service and validation requirements are soft (see reference [79] and Section 2.4.2).

A real-time system model consists of a workload model, a processor model, and the scheduling algorithms. The workload model retains the relevant features of the applications that use the system. In a wireless packet scheduler they are the models of the application traffic, e. g. the distributions of the packet arrival times and of packet lengths. If these exist for audiovisual applications, they do not for TCP traffic, which represents the majority of Internet traffic in bytes (see Section 2.4).

The processor models the resources that process the jobs in the system and have an attribute, the speed, which describes the rate of progress of a job. For a wireless packet scheduler, the processor is the wireless channel and the speed is the data rate used for transmitting a packet. However, the data rate over a wireless channel depends on the quality of the link used to transmit the scheduled packet. Since the link quality varies in time and from one user to another (Sections 2.1 and 2.2.2) the processor speed depends on the scheduling decision and is not constant in time. Although one could use the worst case scenario and take the largest possible execution time of a job into account, the result would be a conservative scheduler which is undesirable in a wireless system, where the resources are scarce and expensive.

The third part of the model, the scheduling algorithm—how to allocate the processor to the jobs—is the subject of this thesis and the requirements of the different applications set the constraints to the allocation. The relevance of the scheduling algorithm for system performance depends on the relationship between the speed of the processor, the workload, and the constraints. When the constraints of the workload can be easily met by the processor (happens when the resources are abundant) the effects of different scheduling algorithms remain mostly unnoticed. But when resources are scarce compared to the workload, different allocation policies play a major role in how often the constraints are met, i. e. the effects of different scheduling policies can be best seen when the workload is high compared to the processor speed.

In summary, a wireless packet scheduler is a soft real-time system, where the execution time of a job is not constant in time but depends on the scheduling algorithm. Furthermore the workload cannot be properly modelled and the jobs have soft QoS requirements with time-varying deadlines. Accordingly, it is nearly impossible to model a wireless packet scheduler for multiple applications in a realistic way for theoretical analysis. Following these considerations, the packet scheduling approach presented in Section 3.4 is based on heuristics and rules derived from well-known characteristics of both the channel, the traffic, and the applications. Then, it is evaluated in Chapter 6 using discrete-event simulations described in Chapter 5 because they enable a more precise characterisation of the behaviour of the scheduling algorithms studied. The performance evaluation will be done in high load scenarios, so that the different behaviours of scheduling policies can be best observed and studied.

3.2 System Model

The scenario considered in this thesis consists of an Access Point (AP) communicating over a time shared wireless channel with several WT—a wireless cell. A centralised architecture is assumed, where WT do not communicate directly with one-another but only over the AP, and only single hop communication is considered. In the direction from the AP to the WT—downlink—the medium is shared as TDM, and in the opposite direction—uplink—as TDMA. Communication in the cell is asynchronous: there is no time framing or synchronisation; a transmission can start at any time instant when the channel is free.

Each packet carries in its header enough information to identify its destination (in the downlink) or sender (in the uplink); also, there is an instantaneous way to inform a WT of uplink allocations. Furthermore, it is assumed that there is no delay between the decision of which packet to transmit and the beginning of its transmission.

It is assumed that the physical layer technology supports adaptive MCS, so the duration of a packet's transmission depends on the data rate used. Furthermore, it is assumed that whenever a scheduling decision has to be made the link layer knows how long each of the packets competing for transmission would occupy the channel. The study of an algorithm that efficiently rules the adaptation of the MCS to link quality is beyond the scope of this thesis; the threshold-based adaptation algorithm used will be described in Section 5.2.1.

The packet scheduler mechanism is located in the link layer of the AP and can make channel allocation decisions for both downlink and uplink traffic, assuming that the AP has information about the traffic backlogged at each WT. The mechanism that enables the exchange of that information is beyond the scope of this work. The QoS requirements and traffic characteristics for each type of data flow supported by the system are known to the link layer of the AP as well as user and flow priorities. It is assumed that it is possible at the link layer to obtain information about each packet, for example about which flow it belongs to and which type of packet it is within a flow. That information can be extracted from the headers of higher layer protocols (e. g. TCP vs. UDP), protocol labels (e. g. RTP markers) or another mechanism. The specific packet classification mechanism is out of the scope of this thesis.

Finally, the model used for the proposed solution assumes that the link layer of the AP has access to profiles that describe the details of the QoS requirements of each supported flow type. Further, it is assumed that the link layer can obtain all necessary information about characteristics of the traffic generated by the applications, e. g. the codec used by multimedia applications (which can be obtained from RTSP or SIP, as was mentioned in Section 2.4.2), so that the correct QoS profile can be chosen.

Technologies that match the features of this system model are 802.11 WLAN [28] and the High Speed Downlink Packet Access (HSDPA) [80] of the Universal Mobile Telecommunications System (UMTS) [81].

3.3 Problem Statement

Challenges of Multi-User Multi-Flow Wireless Packet Scheduling

The load offered to a wireless packet scheduler consists of a diversified mix of data flows with different requirements for packet delivery. Because the ultimate evaluation of communications is done by the user, only data that can be used by the receiving application is important. All data transmitted but not received due to link errors, not delivered to the application because of protocol errors, or delivered so that is cannot be used e. g. because it is too late, is useless for the receiving application and its transmission is seen by the user as a waste of resources. Thus, it is important to express the importance of the packet for application satisfaction in the metrics used to rank the packets competing for channel access.

Furthermore, the importance of a packet must be evaluated from two points of view. On one hand, packets in a same flow can have different importance for the user perceived quality. On the other hand, scheduling data from several flows requires that the importance of packets from different flows be comparable. Thus, it is not enough to evaluate the importance of a packet A for the quality of the flow a it belongs to; it is also necessary to evaluate whether packet A is more important for flow a than packet B for flow b. Due to the diversified traffic characteristics and requirements of the applications, it is not straightforward to rank the importance of packets from one application relative to another.

Moreover, the importance of a packet is not constant but changes with time. Consider a packet from a WWW page and a short VoIP packet which can tolerate a delay larger than that which the transmission of the WWW packet would take. First, the importance of a VoIP packet depends on whether other VoIP packets have been lost on the channel recently or not, since burst packet losses are more damaging than single losses. Per se a VoIP packet is not more important than the WWW packet as long as it is not in danger of missing its deadline. But, if there have been recent losses of VoIP packets, the importance of transmitting that packet on time is higher. Furthermore, as the deadline for the VoIP packet is approached, its importance grows. All these aspects relevant for the user's perspective should be expressed by the metric used to rank packets.

However, because wireless bandwidth is scarce it should also be efficiently used from the channel's perspective. In a multi-user scenario, there are two aspects to consider for efficient channel usage. On one hand, the time variations of the SNR lead to a time-varying error behaviour at a location and it should be avoided to transmit over a link with high error probability to improve the efficiency of channel use—channel-aware scheduling. On the other hand, when the link towards a WT has low quality there is probably another WT with high link quality and it is more efficient to allocate the channel to that second WT—multi-user diversity. The more WT share the channel, the higher the probability of always having a link with good quality, and the higher the resource efficiency of such a packet scheduler. However, delaying a packet until its link has the best quality leads to increased latency. This can cause packets to be delivered when they can no longer be used by the application—a waste of resources that leads to low efficiency.

Thus, it is necessary to find a balance between efficient channel usage in terms of high spectral efficiency and the delivery of satisfactory quality to the users. This work focuses on how, at each scheduling instant, packets from different flows can be ranked and which packet should be chosen for transmission so that the user perceived quality is improved.

Prediction of Link Quality

In the discussion about efficient channel use and adaptive MCS it was assumed that the quality of each link is known at the time of the scheduling decision and that the adequate data rate is chosen. This assumes that it is possible to perfectly predict the future behaviour of a wireless link. Although this is not realistic, it is possible to predict the future link behaviour from samples of past link behaviour. However, that prediction is not perfect and the actual link behaviour may differ from the predicted—prediction errors. This can lead to wrong data rate choices and wrong scheduling decisions if either the predicted link quality or the usable MCS is used in the calculation of the scheduling metric. As a consequence, the system behaviour can differ from what is expected under the assumption of accurate prediction of link behaviour.

As such the accuracy of link prediction needs to be studied together with the effects of prediction errors in the performance of adaptative mechanisms. Since the usable data rate is often used to express the efficiency of channel use in the calculation of ranking metrics for packet scheduling algorithms (as shall be seen in the next chapter), errors in the MCS adaptation can lead to systematic wrong scheduling decisions, severely changing the scheduler behaviour. The study of the accuracy of link prediction and how it affects the performance of rate adaptation mechanisms and packet schedulers is the subject of Chapters 7 and 8.

3.4 The PeLe Packet Scheduler

The approach proposed is based on widespread observations about desirable scheduling goals and application behaviour. First, if the average link quality at a receiver is bad, trying to meet the requirements of an application on that WT may cause too much degradation to the service delivered to other users sharing the channel. So, the scheduler should provide only limited compensation for bad coverage. Second, it only makes sense to keep serving a certain flow if the user is minimally satisfied with the resulting application performance. So, if the load is too high for the available channel resources, gracefully degrading the service to all users may lead to delivering unsatisfying user perceived quality to everyone. Third, the user's perception of the communication performance depends on the application and so does how the perception relates to the delivered network service. Finally, users can tolerate some degradation of the perceived quality; the acceptable amount and duration of degradation depend on the application and on the data. A most important consequence hereof is that it is imperative to provide application-specific support

Based on the considerations above, the goal of packet scheduling should be to achieve acceptable user perceived quality.Thus, it seems straightforward that the scheduling metrics should be based on user perceived quality. The approach proposed here relies on the concept of utility curves, which are explained in more detail in Section 3.4.1. Section 3.4.2 then introduces the calculation of the metrics used to evaluate the importance of a packet and Section 3.4.3 introduces scheduling rules.

3.4.1 Utility Curves

The scheduling approach presented here is based on the existence of curves that map network service delivered to a flow to the quality perceived by the user of the application—utility curves. For clarification, the utility curve is the function; quality is the value of the dependent variable; network service (also only service) is the independent variable and is the "result" of allocating a certain amount of resources to a flow. Resources, as was defined above, is channel time. The utility functions are flow-dependent both in their form and the service that they depend on, as introduced by Shenker at al. [32]. For example, the quality of a file download is typically dependent on the allocated throughput, but the quality of a VoIP conversation depends rather on packet delivery rate and packet delay. Utility curves are increasing functions of the network service—the independent variables of the utility curve. Examples of network service are data rate, packet delivery ratio, packet service time. An important feature of this approach is that how the service delivered to a flow maps to the quality (the form of the utility function) must not be the same for all flow types.

Utility curves express more or less sensitivity of an application's performance to network service. Intuitively, in case it is not possible to meet the requirements of all flows, a scheduler that tries to improve overall quality should allocate more resources to applications that are more sensitive to service degradation. Thus, the form of the utility curves determines the priority of different applications in the case of resource scarcity.

Since the utility curves are used to calculate the scheduling metric and rank packets from a flow against its competitors, the quality values for different flow types should be comparable, i. e. they should refer to the same scale. To achieve this, all utility curves are normalised by their maximum value. Thus, the maximum possible quality is 1 and is achieved when all the network service requirements of the flow are met. i. e. when the flow is satisfied.

There is a "hidden" variable in the use of the utility curves, as described thus far: the service delivered to a flow must be calculated over a certain time interval—θ, in the recent past. This interval determines the timescale over which the quality is enforced. For example, take an application with a bandwidth-dependent utility curve. If the time interval θ is large, a single packet causes little variation in the amount of bits transmitted on the interval θ, leading only to small variations of the service. As a consequence, it can take a long time for the flow's quality to reflect low service. But if the time window θ is shorter, a packet impacts more the amount of data transmitted in the interval, and produces a higher variation of the quality. The appropriate choice of θ depends on the sensitivity of each flow to time constraints.

However, there is though still the question of how to obtain utility curves? To obtain the application-specific relationship between the network service allocated to a flow and the user perceived application quality it is necessary to study each application by itself. One must realise its sensitivities to variations in delivered service. For example, how does the perceived quality of a VoIP react to one or two lost packets? How does TCP react to lost packets or to bandwidth changes? And in which timescales? It is also important to observe the traffic patterns generated by applications, e. g. whether the generated traffic consists of short packets like VoIP, or mostly large packets like bulk TCP, or whether traffic bursts can occur, with which magnitude and frequency. This information can be obtained from empyrical studies and observations, as will be seen later on in the study of possible utility curves for the example applications.

3.4.2 Estimating Future Application Quality

Everytime a scheduling decision must be made, i. e. the next packet to transmit must be chosen, the head of queue (HOQ) packet of each flow i with backlogged data competes for the channel. Each FM knows the amount of network service delivered to the flow i over the interval $[t - \theta_i; t]$—ν_i^t—and the corresponding current quality of the flow—$\xi_i = u_i(\nu_i)$. Two values are then used to express the importance of a packet from flow i for the overall quality:

- the estimated quality increase for flow i due to the transmission of its packet—ξ_i^i for flow i;
- the estimated quality decrease for another flow j, $j \neq i$, due to postponing its packet while the packet from flow i is transmitted—ξ_j^i for flow j when flow i is served.

These two metrics can be combined to rank flows taking into account not only the estimated benefit of transmitting a packet, but also the estimated cost of that transmission expressed by the estimated quality loss of the other flows. The second metric is important when the goal of the solution is to improve the overall quality.

Before a scheduling decision at time t, ξ_i^i and ξ_i^j are calculated for each flow i which has backlogged packets according to the following procedure:

1. calculate the total time that the packet would occupy the channel if transmitted (including overheads, like headers, polling, inter-frame spacings or acknowledgements, if necessary)—d_i, which depends on the MCS used;
2. calculate the amount of network service that would be delivered to flow i after the transmission of its HOQ packet (at $t + d_i$)—$\nu_i^{t+d_i}$ (because flow i's service increases $\nu_i^{t+d_i} \geq \nu_i^t$;
3. calculate the corresponding quality estimate—$\xi_i^i = u_i(\nu_i^{t+d_i}) \geq \xi_i$;
4. gather information about the duration of the other flows' packets—d_j, $\forall j \neq i$;
5. calculate, according to d_j, the service that would be delivered to flow i at time $t + d_j$ if it is not served (receives 0 resources in the interval $[t; t + d_j]$)—$\nu_i^{t+d_j} \leq \nu_i^t$;
6. calculate the corresponding quality estimate if flow j is served instead of flow i—$\xi_i^j = u(\nu_i^{t+d_j}) \leq \xi_i$;

Then, with these values the scheduler builds a matrix of future quality estimates, where line i contains the values of the estimated future quality of each flow j if flow i is scheduled[1].

$$\begin{bmatrix} \xi_1^1 & ... & \xi_j^1 \\ ... & \xi_i^i & ... \\ \xi_1^N & ... & \xi_N^N \end{bmatrix} \tag{3.1}$$

The link quality information is indirectly taken into account in the estimates of quality decrease of the flows that are not served. The link quality is used to choose the MCS for transmission, and consequently the channel occupation time d_i: the worse the link quality, the lower the usable data rate, and the larger the duration of the packet from flow i on the channel d_i. For all flows j that would have to wait for the transmission of flow i, a larger d_i means a bigger quality decrease due to the longer time without service. As a consequence, ξ_j^i, $j \neq i$ is smaller the worse the channel quality of flow i.

[1]The setting up of the matrix of future qualities for purposes of calculating a scheduling metric has been awarded patent nr. 10355117 by the German Patent and Trade Mark Office.

3.4.3 Scheduling Metrics and Rules

The scheduling metric used for ranking the flows is calculated from the matrix of estimated future qualities. The matrix (3.1) might be used in several ways to pursue different goals. Because the goal of this scheduler is to improve overall user perceived quality, the sum of each line is used as the scheduling metric: $\sum_j \xi_j^i$ is the sum of the estimated quality of all flows after the transmission of the packet from flow i.

$$\begin{bmatrix} \xi_1^1 & \dots & \xi_j^1 \\ \dots & \xi_i^i & \dots \\ \xi_1^N & \dots & \xi_N^N \end{bmatrix} \rightarrow \begin{bmatrix} \sum_j \xi_j^1 \\ \sum_j \xi_j^i \\ \sum_j \xi_j^N \end{bmatrix}$$

Since ξ_i is the quality of flow i at the scheduling instant, ξ_j^i can be written as the sum $\xi_j + \Delta\xi_j^i$, and $\sum_j \xi_j^i = \sum_j \xi_j + \sum_j \Delta\xi_j^i$. The term $\sum_j \xi_j$ is common to the scheduling metric of all flows, since it is the sum of the current qualities and does not depend on i; what ranks the flows is $\sum_j \Delta\xi_j^i$, the overall sum of the estimated quality variations due to the transmission of the HOQ packet of flow i. Since the quality curves are monotonically increasing with the service delivered (Section 3.4.1), it follows that

$$\Delta\xi_j^i = \begin{cases} > 0 & , j = i \\ \leq 0 & , j \neq i \end{cases}.$$

Thus, the scheduling metric

$$\sum_j \Delta\xi_j^i = \Delta\xi_i + \sum_{j \neq i} \Delta_\xi j^i \tag{3.2}$$

weighs the estimated quality increase due to serving flow i with the estimated quality decrease induced in all other flows j, $j \neq i$.

The scheduler chooses flow

$$i : \text{argmax}_i \sum_j \xi_j^i \tag{3.3}$$

that maximises the overall estimated quality variation at each scheduling decision. In case of a tie, the scheduler chooses from among the tied flows the one with lower current quality

$$i : \text{argmax}_i \xi_i. \tag{3.4}$$

If there is still a tie after applying both previous rules, it is broken randomly. Henceforth this scheduler will be called PeLe.

3.4.4 Weighting Utility Curves

The relative importance of different flow types is determined by the shape of the utility curves, but it may be desirable to change the flow type hierarchy at runtime or to give some users priority over others; in this case utility curves can be weighted. The scheduling metric defined in Eq. 3.3 is determined by the variations in the quality of each flow compared to its current quality. Those variations are determined by the gradient of the utility curve at the current service $u'(\nu)$, such that a flow x with weight a_x sees its scheduling metric multiplied by that weight (assume here that flow y has $a_y = 1$)

$$\Delta\xi_x^x = a_x \cdot \Delta\xi_y^y.$$

At the same time, the scheduling metrics of competing flows also decrease by an amount a_x times larger, since

$$\Delta\xi_x^j = a_x \cdot \Delta\xi_y^j.$$

In this way, flow x has a larger relative scheduling priority than flow y, as long the flows are not satisfied ($\xi_i = 1$). When flows are satisfied, their quality cannot be further increased and the scheduling decision is taken mostly by the tie solving rule (Eq 3.4), which picks the flow with lowest quality. Using a weight in this case would lead the flow with largest weight never to be scheduled, something that is clearly not intended. So, the weight is used alone in the calculation of the main scheduling metric in Eq. 3.2.

If a certain flow y should receive more service than another flow x, that should be expressed in the service requirements of the flow. I. e. if user x should receive twice as much bandwdith than flow y then $r_x^{\max}$ corresponding to the maximum quality of flow x should be twice $r_y^{\max}$, the maximum service requested for flow y.

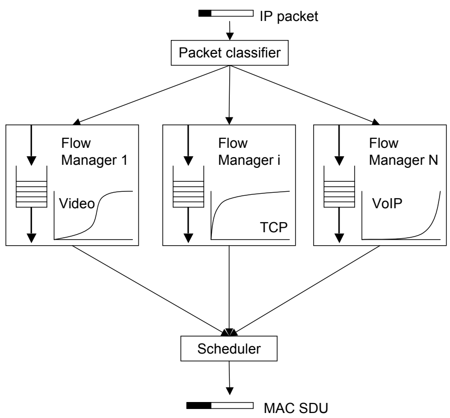
In summary, weighting the utility curve of a flow x—$u_x(s) = a_x \cdot u_y(s)$—increases its servicing priority in case of resource scarcity; to weight the maximum desired service, the service requests should be weighted—$u_x(a_x \cdot s) = u_y(s)$—i. e. flow x experiences a certain quality ξ at a service a_x larger than the service necessary for flow y to experience that same quality.

3.5 Flow-specific Support

As argued at the beginning of the previous section, providing satisfying user perceived quality to different flows requires that each flow be handled differently at the AP. The previous section introduces a packet scheduler that uses flow-specific utility curves to estimate the quality of a flow depending on the service it has received in recent past. But other kinds of application specific support can be provided at the link layer, such as queueing and dropping policies and packet differentiation, as was seen in Section 2.4.

The architecture in Figure 3.1 is introduced to provide link layer application specific support (queueing, utility function, tracking of received service, etc). All flow specific support, including the calculation of the scheduling metric, is enclosed in architectural entities called Flow Managers (FM). The FM implement a flow specific queueing policy, track the network service delivered to each flow over the time window θ, and calculate the estimated quality metrics according to the flow specific utility curve. The scheduler does not know which flow type it is dealing with, which queueing policy is being used or whether the flow is delay or loss sensitive—all flow specific information is hidden in the FM and summarised in the scheduling metric. This modular approach is very flexible, as specific support of an existing application can be easily changed or new applications added to the system. This is an important issue regarding the fast development of wireless communications and the growing variety of applications.

Figure 3.1: Architectural flow-specific support for the proposed utility-based packet scheduler.



This thesis uses four example applications to illustrate the problem studied and evaluate the performance of the scheduler proposed. The rest of this section describes possible implementations of the concepts described thus far to support those applications. For each flow, a queueing discipline, an estimate of a utility function and possible additional support (like packet differentiation) are proposed. The proposed estimates of utility functions for each application type are heuristic approaches, based on widespread knowledge of the application behaviour, in which case references to sources are provided, or based on an empyric study of several possibilities, in which case that study is presented. The mathematical expressions used for the estimate utility functions of

Bulk, WWW or VoD flows are chosen so that they describe the relationship between service and user perceived quality according to knowledge of application behaviour—they too are heuristics. These solutions are neither unique nor the results of optimisation. The choice of a value for the interval θ for each application will be studied in Chapter 6.

3.5.1 Support of Bulk File Transfer

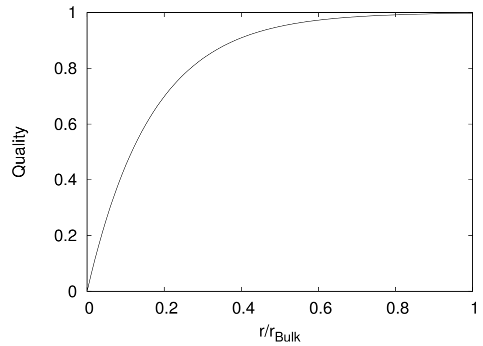
Queueing The queue for TCP-based bulk data transfer flows is a drop tail FIFO queue of pre-defined length $q_{\mathrm{max,Bulk}} = 2^{16}$ Bytes, which is the value of the maximum TCP window size, to avoid packet drops in case of a burst data arrival.

Estimate of a Utility Curve It was seen in Section 2.4.1 that TCP flows can adapt to variable available bandwidth and that bulk data transfers have no interactivity requirements. So the utility curve used should express elasticity with respect to the received throughput, as argued in [32]. The utility function used is described by

$$u_{\mathrm{Bulk}}(x) = 1 - e^{-ax} \tag{3.5}$$

and can be seen on Figure 3.2 for $a = 6$. The network service on which these flows depend is the throughput, which is normalised to the value of the requested throughput r_{Bulk} and is calculated over moving intervals of θ_{Bulk}. From the plot it can be seen that even if the throughput delivered to the flow in the period θ decreases to half of the requested, the experienced quality is still very high. Actually, significant quality decreases can only be seen when the received throughput threatens to decrease below 40% of the requested.

Figure 3.2: Estimate of a utility function of a bulk data transfer.



Bulk Transfer Specific Support Since TCP is designed to adapt itself to variable bandwidth in timescales of hundreds of milliseconds, this work assumes that the congestion control mechanism can cope with the variable bandwidth that TCP flows experience on shorter timescales. According to [44] the issues that most disturb TCP behaviour are packet losses and out-of-order packet delivery, which lead to timeouts and retransmissions. The channel-aware scheduling ensures that packet losses are kept below a certain threshold by postponing TCP packets to periods of good channel. Also packets are delivered in order, as far as the wireless link is concerned, due to the FIFO queue. As argued in [40], as long as the channel is re-allocated to a TCP flow with high frequency (at a higher frequency than the timer's granularity), the throughput is not degraded. This is the case in our system assumptions, where the coherence times of the channel are lower than the TCP timers, which usually have a granularity of 500 ms. The choice of this link layer behaviour takes into account the considerations on cautious use of cross-layer mechanisms in [82], according to which TCP support over wireless should not be overcomplex due to the risk of having mechanisms at different layers working against each other.

3.5.2 Support of WWW

Queueing The FM for WWW traffic is similar to the FM for Bulk traffic: it has a drop back FIFO queue of size 2^{16} Bytes so that an entire maximum congestion window of data fits in.

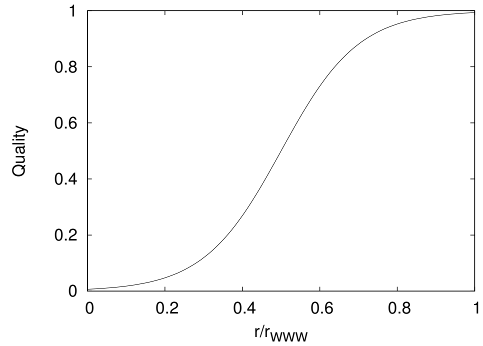
Estimate of a Utility Curve The service that the utility curve depends on is the throughput, as this is also a TCP-based application. Due to its interactive nature, Web traffic is expected to be more sensitive to low service than file transfers [43], so an inelastic utility curve seems more appropriate to support WWW traffic. The inelastic utility curve has the form

$$u_{\mathrm{WWW}}(x) = \frac{1}{1 + exp^{-a(x-b)}}, \tag{3.6}$$

where a defines the steepness of the curve and b the point $u(x) = 0.5$, and can be seen in Figure 3.3 for $a = 10$ and $b = 0.5$. In Chapter 6 this utility curve will be compared to other possible ones and this choice justified.

WWW-Specific Support Due to the interactive nature of WWW traffic, it is more important to avoid retransmission timeouts for these flows than it is for Bulk data transfers, especially at flow start when the timers have not yet been adapted to the real round trip time. So, packets should be extra protected against transmission errors by allowing for one link layer retransmission; for SYN packets a higher value should be used [44].

Figure 3.3: Estimate of a utility function for a WWW flow.



3.5.3 Support of VoIP

Queueing The VoIP queue is a drop front FIFO queue and its length plays an important role in the QoS support offered, as VoIP traffic is delay-sensitive. The queue is dimensioned so that packets are kept for at most the maximum acceptable delay $\delta_{\text{max,VoIP}}$, taking into account that the flow has constant data rate $\bar{r}_{\text{VoIP}}$ and constant packet length L_{VoIP}.

Take a FIFO queue of size q_{max} bits, where the oldest packet is dropped if the queue is full on a new packet arrival. If the queue is not served at all (worst case scenario), how long does it take for a packet that arrives at the full queue to be dropped? Since on a new arrival the oldest packet in the queue (the front one) is dropped, a packet moves from the back to the front of the queue at the packet arrival rate $\bar{r}_{\text{VoIP}}/L_{\text{VoIP}}$. The maximum time that the packet spends in the queue before being dropped is, then,

$$\frac{q_{\text{max}}}{\bar{r}_{\text{VoIP}}/L_{\text{VoIP}}}$$

Accordingly, the size of the queue is calculated so that a packet does not experience more than a pre-defined maximum delay:

$$q_{\text{max,VoIP}} = \left\lceil \frac{\delta_{\text{max,VoIP}} \cdot \bar{r}_{\text{VoIP}}}{L_{\text{VoIP}}} \right\rceil, \tag{3.7}$$

L_{VoIP} the length of a link layer VoIP packet (voice samples plus RTP, UDP, IP, MAC and PHY headers)[2]. The ceiling function is used in case the maximum delay does not correspond to an integer number of packets at the given data rate. This can lead to a higher maximum delay than required by at most $L_{\text{VoIP}}/\bar{r}_{\text{VoIP}}$, but it is assumed to be affordable because the requirements are soft and VoIP packets short.

[2]The values of $\bar{r}_{\text{VoIP}}$ and l_{VoIP} are encoder-dependent and it is assumed that the encoder used can be assessed during the flow set-up phase.

Estimate of a Utility Curve VoIP packets that arrive too late at the receiver cannot be decoded and are dropped. So, what is relevant for the decoder is the amount of missing coded voice samples [1], no matter whether they were dropped in the network or arrived too late at the buffer. As such , the abscissa of the VoIP utility curve for a VoIP application is the packet delivery rate (PDR). The PDR is tracked by two counters, one counting the total number of packets and one tracking the number of delivered packets. Each time that a packet is dropped in the queue, the event is reflected in the received network service by increasing the total packet count, but not the count of delivered packets. The PDR is counted over a sliding window of duration θ_{VoIP}=200 ms (details shall be given in Chapter 6).

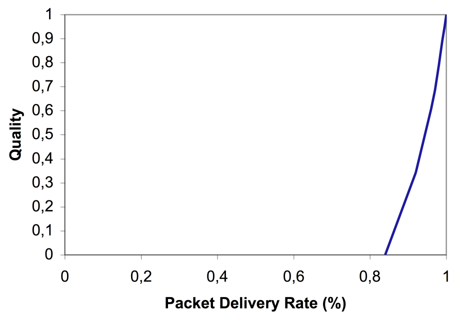
The estimate of a utility curve for VoIP applications is based on values of the ITU for the impairment factor of VoIP calls defined in [1]. The impairment factor I_e is used in network planning to describe the quality of speech transmission [1, 72] and maps the quality of the decoded VoIP speech to the voice samples loss ratio. The best service that a network can provide is a VoIP packet loss ratio (PLR) of 0, equivalent to PDR=1. Table 3.1 shows the PLR and I_e obtained from [1], the PDR and the corresponding mapping to quality assuming that for $I_e \leq 16$ the quality is 0. Figure 3.4 shows a plot of the corresponding estimated VoIP utility curve as is used in the evaluation.

Table 3.1: VoIP Impairment factor as a function of packet losses [1] and corresponding estimate of VoIP utility.

PLR [%]	I_e	PDR [%]	Quality
0	11	1	1
0.5	13	0.995	0.945
1	15	0.99	0.895
1.5	17	0.985	0.842
2	19	0.98	0.789
3	23	0.97	0.684
4	26	0.96	0.605
8	36	0.92	0.342
16	49	0.84	0

VoIP-specific Support The network latency is the sum of the delay that the packet suffers between the sender and the AP, the queueing delay at the AP, and the transmission time. The link layer can only influence the queueing delay and the dropping at the AP queue, and this is done by the queueing policy described above. A better VoIP support could be achieved, for example

Figure 3.4: Estimate of a utility function for a VoIP call.



by tracking the sequence numbers in the RTP headers and taking packet drops upstream from the AP into account, or by making the delay budget packet dependent by tracking the timing information in the RTP header. However, in the scenario used for the evaluation (which will be described in Chapter 5) these mechanisms are not relevant.

3.5.4 Support of VoD

Queueing The queueing support for VoD takes into account the different importance of the frames for the video quality [3]. The queue is a FIFO with a packet dependent dropping policy. If there is not enough place to insert an arriving packet, one or more packets are dropped, depending on the frame type that the data carried by the arriving packet belongs to, according to the following rules:

B-frame the oldest packets containing B-frames are dropped until the arriving packet can be inserted into the queue; if there is still not enough place for the new packet, it is dropped;

P-frame the oldest packets containing B-frames are dropped until the arriving packet can be inserted into the queue; if there is still not enough place for the arriving packet, the oldest packets containing P-frames are dropped until the arriving packet can be inserted; if the packet still does not fit into the queue, it is dropped;

I-frame all packets containing B-frames are dropped; if there is still not enough place for the arriving packet, the oldest packets containing P-frames are dropped until the queue fill state is below a threshold and the

[3] This queueing policy is not novel and similar policies have been proposed in the literature [61, 62]

packet with the I-frame is then inserted; if there are only packets carrying I-frames in the queue, the oldest ones are dropped until the arriving packet can be inserted.

The VoD queue is 2^{16} Bytes long, so that it can support the rate peaks that can occur in a VBR video stream.

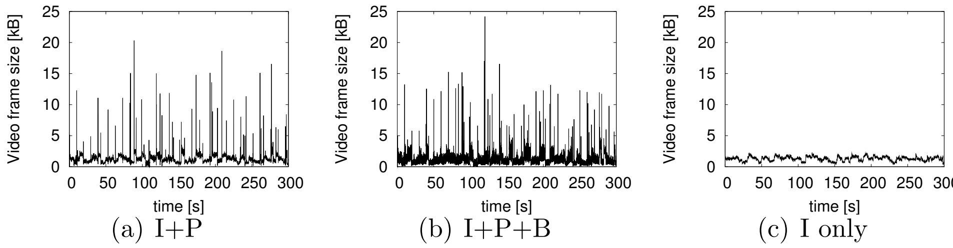
When packets from an I- or P-frame are dropped from the queue, it means that the flow has been receiving less service than requested. Since this is not an elastic application, packet drops should be reflected in the service delivered to the flow (and consequently on its estimated quality). To achieve this, when a packet is dropped, its length is added as negative service in the FM, giving the VoD flow higher priority after a drop and helping avoid further drops.

Estimate of a Utility Curve Utility curves appropriate for VoD traffic are non-elastic throughput dependent utility curves described by a sigmoid function

$$u_{\mathrm{VoD}}(x) = \frac{1}{1 + \exp^{-a+(x-b)}} \tag{3.8}$$

as supported by available results [32, 83, 84]. The parameterisation of the

Figure 3.5: Size of the frames of the H.264-encoded sample video depending on the GOP. The range of the y-axis are the same in the three cases for better comparison.



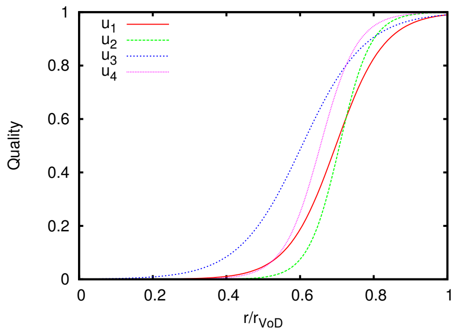
VoD utility curve flow must take into account the irregular traffic pattern of VBR video flows. Although the average bitrate $\bar{r_{\mathrm{VoD}}}$ of H.264 and MPEG-4 streams can be preset before the encoding, it has a variance which depends on the contents of the video and on the parameters of the encoding (e. g. GOP), as can be seen in Figure 3.5 for the same video encoded using different GOP structures. So, the utility curve should be a function not only of the mean requested throughput $\bar{r_{\mathrm{VoD}}}$, but also of the maximum expected/desired throughput $r_{\mathrm{VoD}}^{\max}$. The steepness a and location b of the utility curve can be determined by the two points: the maximum quality $Q_{\max}$ corresponding to the maximum rate $r_{\mathrm{VoD}}^{\max}$ and the quality Q_X corresponding to the average flow rate $X = \bar{r_{\mathrm{VoD}}}/r_{\mathrm{VoD}}^{\max}$. A higher X means that the rate peaks are small compared with the average rate, like in Figure 3.5-c, whereas a lower X expresses higher rate variations as in Figure 3.5–a. The parameters a and b can be calculated according to the following heuristics

$$b = \frac{\ln\left(\frac{1}{Q_X} - 1\right) - X \ln\left(\frac{1}{Q_{\max}} - 1\right)}{\ln\left(\frac{1}{Q_X} - 1\right) - \ln\left(\frac{1}{Q_{\max}} - 1\right)} \tag{3.9}$$

$$a = \frac{-\ln\left(\frac{1}{Q_{\max}} - 1\right)}{1 - b} \tag{3.10}$$

Several curves obeying Equations 3.9 and $u(X) = X = \frac{\bar{r}}{r^{\max}}$ are shown in Figure 3.6. The differences in the support given by the different curves will be studied in Chapter 6.

Figure 3.6: Possible non-elastic utility curves for VoD.



VoD-specific Support Due to the very high importance of I-frames (the loss of an I-frame renders all following frames up to the next I-frame useless) they are protected by link layer retransmissions. All packets containing parts of an I-frame are entitled to up to 3 retransmissions if they suffer transmission errors. Packets containing parts of P-frames can be retransmitted once in case of transmission errors, because the large GOP leads to long error periods if a P-frame gets lost. The number of retransmissions for each frame type could be further fine-tuned, but the values used have shown to be sufficient to achieve a very low amount of packets lost due to transmission errors.

Chapter 4

Wireless Scheduling and QoS Support

This chapter presents relevant work in resource allocation in wireless TDM/ TDMA systems, showing also the evolution of the problem of wireless scheduling. Through the years, the way of looking at wireless resource allocation and packet scheduling changed with the evolution of wireless systems. For example, adaptive MCS techniques were initially not widespread and the main goal was to achieve fair allocation of bandwidth shares despite transmission errors. For the sake of completeness, solutions developed for that scenario are also presented in Section 4.1, even though it differs from the scenario assumed in the rest of the thesis. As wireless systems and the traffic carried evolved, the problem formulation and approaches changed, as is shown in Sections 4.2. In Section 4.3, some approaches that also use utility functions are presented. The main differences between those and the approach proposed in the previous chapter are discussed in Seciton 4.5.

4.1 Wireless Link Sharing

The problem of resource allocation in communication networks appeared in the context of sharing the outgoing link of a gateway, switch or router among several sources—multiplexing in fixed networks. The main concern was flow separation, i. e. eliminating the influence that a misbehaving flow could have in the service delivered to the others [85, 86]. These concerns motivated the appearance of several policies designated as Fair Queueing (FQ), named queueing because the underlying model consisted originally of a single queue shared by different flows. A different queue per flow was introduced together with a scheduler that chooses which queue to serve—the first enables flow isolation and the second enables controlled sharing of the bandwidth, according to flow weights. Ideally, the different queues should be served according to the Generalised Processor Sharing (GPS) algorithm [87], a concept borrowed from processor sharing, which simultaneously serves all queues at rates corresponding to their share of the link bandwidth.

But GPS is a fluid model, fully theoretical and one which cannot be implemented in a network where data is organised in packets, so a lot of effort was put into approximating GPS in real packet networks. Weighted Fair Queueing (WFQ) [85], Virtual Clock (VC) [88], Worst-case Fair Weighted Fair Queueing (WF^2Q) [89], and EVVDF [90] try to approach the performance of GPS by minimising the differences between a real network and the time in an emulated fluid network; and Self-Clocked Fair Queueing (SCFQ) [91] avoids that complexity by not needing to simulate the fluid system. A comparison of these algorithms can be found in [92] in terms of flow isolation, complexity of implementation and delay performance. The development of these algorithms was driven by the appearance of real-time services on fixed networks and the need to provide services other than best-effort on shared links.

Then wireless communications emerged with shared, error-prone wireless links that had to be accounted for. Most adaptation of FQ to error-prone channels build mechanisms to compensate flows for their bad link periods when the link is good again, taking care not to impact flows which did not suffer errors. Several approaches were proposed that aim at distributing the available bandwidth according to pre-defined shares/weights.

The Server-Based Fairness Approach (SBFA) [93] is a scheme designed to provide limited compensation of bad link periods to any FQ scheme above. The Channel-State Independent Fair Queueing (C-IFQ) [94], the Idealised Wireless Fair Queueing (IWFQ) [95] and the Wireless Fair Service Algorithm (WFS) [96] simulate a error-free systems where the queues can be served according to a desired FQ policy, but in the real system the same queue is chosen only if the link is good. Lag and lead counters are introduced to keep track of the positive or negative difference, respectively, between the service that a flow actually receives and the service it is entitled to according to the weights. The lead and lag counters are limited so that the compensation given to a flow causes only limited degradation of the service to other flows. The first observation regarding desirable packet scheduling goals in Section 3.4 follows from this observation.

The Class-Based Queueing with Channel State Dependent Packet Scheduling (CBQ+CSDP) [97] follows a different approach. The actual available channel bandwidth is re-calculated after each packet transmission to account for transmission errors and the scheduler tries to allocate shares of that bandwidth instead of shares of the theoretically available bandwidth. In this case the effects of link errors are distributed among all WT according to their fairness weigths. The Havana framework [98] compensates flows for bad link periods using a lag counter, and the compensation for bad link periods is done at the cost of all other flows. Additionally, it deals with long-term channel bandwidth degradation by dropping low priority packets, which have to be marked by the application.

These scheduling algorithms aim at sharing the wireless channel according to pre-defined bandwidth fractions expressed by weights and additionally compensating for losses due to bad quality of the wireless link. This implies a throughput-oriented QoS paradigm, i.e. the requirements of each flow are expressed exclusively in terms of fractions of the available bandwidth. Furthermore, compensation of flows for bandwidth losses and spreading the effects of link errors are also done fairly according to the weights. This approach implicitly assumes that bandwidth degradation has an equally bad effect on all flows. It is well known that this assumption is not realistic as each application has different sensitivity to service variations [32, 99, 100, 101]. Besides the throughput-oriented QoS concept, the approaches mentioned in this section assume a channel with a total capacity known a priori. But most wireless systems today support adaptive rate mechanisms, as was seen in Section 2.2.2, where the channel throughput cannot be known in advance, because it depends on the quality of the links towards the different users and on the multi-user diversity gain, as will be seen in the following section. In that case, it is at most possible to define percentage of channel usage time, again using a throughput-based QoS paradigm.

The Effort-limited Fair (ELF) scheduling approach proposed by Eckhardt et al. [101] addresses the subjects above. It proposes an adaptation of weighted FQ to wireless channels which enables one to invest more resources (effort) in the compensation of some flows more than of others, to account for different importance of flows and different sensitivity to service degradation. It also proposes to limit that effort on a flow specific basis, so that a single flow cannot starve all others just because it has high priority and a bad link. The ELF neither distributes capacity losses nor compensates fairly according to the bandwidth weight. Instead, it enables the increase of the fair weights by amounts that compensate for the link errors suffered (up to a limit) in an application dependent way, so that the desired "outcome" of each flow is achieved. The major downturn of this approach is that it still relies on a concept of QoS based on throughput. Further, although the proposed implementation does not rely on an a-priori knowledge of the link capacity, it is not clear how the flows' weights should be set when the error-free capacity of the link is known in advance due to adaptive rate mechanisms.

4.2 Multi-User Diversity and Opportunistic Schedulers

As was seen in Section 2.2.2, wireless channel variations in a multi-user environment can be explored to increase the throughput of a system with rate adaptation capability. When data must be sent to several users at independent locations, the probability that at any time at least one user experiences excellent link quality increases with increasing number of users. If data is sent to the user with the best link quality, more data can be transmitted overall on the shared channel than the average link quality would allow—this gain in channel capacity is called multi-user diversity gain [102] and schedulers that take advantage of it are usually called opportunistic schedulers (OS).

The Proportional Fair Scheduler (PFS) [103, 104, 102], developed for the downlink of a cellular system, exploits multi-user diversity by allocating the channel to the user which has the current best link relative to its own average link quality. This accounts for unfairness due to different average link quality of the users, since users with low link quality would be discriminated against if the absolute link quality were used, as the max C/I does. For each timeslot, each user i feedbacks the rate that can be used for transmission in that slot R_i (an indication of link quality). The scheduler tracks the throughput allocated to each user, T_i, over a time window t_C using a low-pass filter, and allocates the channel to the user with highest $\frac{R_i}{T_i}$ among all backlogged users. The time constant of the low-pass filter determines the amount of time that a flow can be deferred whilst waiting for a better linkİf it is large, there is a larger probability to see a better link, but it also expresses a larger acceptable delay. Details of the behaviour of PFS were analysed in [105] and [106] and improvements to the update mechanism of T_i proposed in [107]; but, in [108] the scheduler was proven unstable for finite queue backlogs. Despite its simplicity the PFS only cares about the overall efficient and fair channel use, without regarding application requirements like bandwidth or delay.

Andrews et al. in [109, 110] propose a scheduler that differentiates between two different kinds of QoS requirements: one expressed as a delay threshold δ_i and a maximum probability of meeting it t_i for flow i, and another expressed as a minimum bandwidth r_i. The Modified Largest Weighted Delay First (M-LWDF) allocates the channel to the user with the largest metric $a_i W_i R_i / T_i$, where R_i is the current possible rate for flow i and T_i is the average rate for that user, as for the PFS. W_i is the delay for the oldest packet in the queue of flow i if the flow has delay requirements; for flows with minimum bandwidth requirements, a token bucket filled at the desired rate is used and W_i is the delay of the oldest token in the bucket. a_i are weights that takes a value depending on the type of flow: $-\log \delta_i / t_i$ for delay-sensitive flows; for bandwidth-sensitive, a_i can be freely set, whereby a larger value with respect to other flows reduces the timescale over which the scheduler tries to meet the requested rate. The term R_i/T_i makes the allocation more likely close to each user's quality peaks, like the PFS, but it is weighted by the flow's delay/rate requirements. This scheduler enhances the PFS by incorporating application requirements in the scheduling metric and allowing for a differentiation of the service delivered to different applications.

The Exponential Rule (EXP) scheduler proposed later by the same authors [109, 111] weighs the link condition term R_i/T_i with the delays of the queues, trying to balance the weighted delays of all queues. The scheduler chooses the user with the highest metric $a_i \frac{R_i}{T_i} \exp\left(\frac{a_i W_i - a\bar{W}}{1+\sqrt{a\bar{W}}}\right)$, where $a\bar{W} = \sum_i a_i W_i$ is the weighted average of the delays of the first packets of each queue. If $a_i W_i$ is larger than $a\bar{W}$ by more than $\sqrt{a\bar{W}}$, the exponential term becomes very large, overriding the link condition term. This scheduler uses token queues for providing minimum throughput, similarly to the M-LWDF above, substituting packet delays by token delays in the calculation of the scheduling metric. The EXP scheduler performs better than the EXP in balancing the delays of real-time users while providing high throughput to non-real-time flows. However, although it is mentioned that the weights can be used to trade off balancing the delay with being proportional fair, it is not how the weights should be set, especially when the variety of applications, their requirements and the number of users are not constant. Furthermore, balancing the delays, even if weighted, can lead to a degradation of several flows as a consequence of a bad channel period of a single flow with large weight, similarly to the cases with compensation for link errors in the previous section.

More recently, a multitude of opportunistic schedulers which weigh the link quality with service-related metrics has been proposed. The Sender Buffer Sensitive (SB) scheduler [112] introduces an extension to the PFS to support real-time traffic based on the length of the send queue backlogs and on the assumption that when these are large the playout buffer at the receivers are small and vice-versa. At each scheduling time only the best non-real-time flow competes with the real-time flows for channel access. The metric used for the real-time flows is the relative link quality weighted R_i/T_i weighed by the ratio of incoming to outgoing data and a buffer weight, which increases the priority of a flow when its queue exceeds a threshold. The SB scheduler achieves the same performance as the PFS and EXP for real-time flows, but delivers a higher throughput to the best-effort flows. This is achieved by allocating the channel to best-effort flows when real-time flows can afford to wait (their playout buffers are not starved). The results show that giving real-time flows a higher priority than best-effort flows all of the time may be unnecessary; letting real-time data wait until the deadline is enarly reached can deliver a better throughput to rate-based flows without degradation of the service delivered to real-time flows.

Reference [113] combines link quality information with information about the size of data chunks to transmit to minimise the time necessary to complete all transmissions. The Traffic-Aided Opportunistic Scheduling (TAOS) weighs the link quality metric above with the inverse of the size of the file to transmit. In another approach (TAOS-2), the scheduler chooses the user which locally minimises the remaining time to complete all the file transfers, using the remaining amount of data to transmit and the link quality.

Shin et al. propose in [114] an extension to the EXP scheduler that supports 4 different QoS classes where the scheduling metric is calculated differently for each class, taking into account queue state and packet deadlines for interactive traffic, and additionally packet drops for conversational and streaming classes. Background traffic is scheduled according to a maximum rate criteria only if no delay-sensitive class has data to send. This scheduler provides good QoS for a mix of 4 traffic types (similar to the ones in this thesis) as well as fair service degradation, although (as was seen in the previous section) this may not be a desired goal..

Abedi proposed the Multi-dimension QoS Packet Scheduler (MPQS) [115] defining a scheduling metric, which is a weighted sum of parcels that express different factors to be taken into account in the scheduling decision. The first parcel is the QoS satisfaction of a user, which is measured by the amount of packets delivered within a pre-defined deadline. The second parcel is the link quality. The third parcel is the effectiveness with which the channel can be used and is a function of the possible transmission rate, the amount of queued data and the history of packet error rate for the chosen transmission rate. The fourth parcel is the relation between the incoming and outgoing data rate for the flow. Finally, the fifth parcel is a function of the delay and is higher for flows with packets reaching the deadline. Appropriate choice of the sum weights makes the scheduler give more importance to the desired features and therefore it can be used to change the goal of the scheduling policy [116, 115]. Although the scheduler is very flexible, it provides little flow isolation since the parcels of the sum are calculated with respect to the worst performing flow. As a consequence, the scheduler tries to balance the performance of all flows in the aspects that have higher priority (higher weights), a kind of max-min approach that can lead to the degradation of the quality of all flows if a few become very bad.

These solutions are based on a balance between using periods of good link quality and meeting QoS requirements expressed as data rate and/or maximum delay. Accounting for different application classes is solved by setting weights of each application or class, although it is not clear how this should be done. The next section presents a very different approach to QoS and algorithms aims to provide good service to users according to it.

4.3 Utility based QoS Support

In 1995, pioneering work by Shenker [32] described the need to extend the best-effort Internet service model to support real-time and multimedia services and eventually other applications that might appear besides data traffic. For the first time, he highlighted the fact that different applications have different service requirements and their performance varies in different ways with the service provided. Thus, the fundamental goal of network design should be to satisfy the users rather than to meet network-centric goals like bandwidth, delay or percentage of correctly delivered packets. Furthermore, since different applications have different requirements, the performance of the network must be evaluated as the degree to which the specific requirements of each application are satisfied, instead of using network metrics. To achieve this, the notion of utility is introduced for the first time in the context of networks and QoS. Given the set of services relevant to each application, the utility function maps that set of services into the performance of the application, such that a higher utility means a better application performance and user satisfaction. According to this paradigm, the goal of network design should be to maximise the total utility of the applications using the network, and this can be achieved only by providing services that are closely matched to each application's needs. This enables the shifting of resources from applications which are less sensitive to service degradation to others which would suffer more from low service. In the end, the first receive worse network service while the latter receive better service; but the sum of the utilities of all applications is increased precisely because the first are less sensitive than the latter to the service delivered. When the goal of network design is the maximisation of the total utility, the load status of the network also depends on the utility functions of the applications using the network at any time. Thus admission control also depends on the form of the utility functions. For example, for increasing, strictly concave functions of the bandwidth the addition of a new flow always increases the total utility whereas for a convex function the addition of a new flow reduces the total utility. Finally, in the context of admission control, several forms of utility curves are proposed which express different sensitivities of applications to network services. Bulk data traffic, being transported by TCP, is classified as elastic due to its capacity of adaptation to available bandwidth and its typical utility curve is defined as increasing and strictly concave. This is the generally accepted notion of elasticity and utility function of elastic traffic.

While Shenker argues for differentiated QoS provisioning and a user-centric QoS paradigm, Hayes et al. [100] argue that satisfying a maximum number of users should be the goal of resource allocation in a network from the point of view of the network operator, since that minimises the amount of user complaints. Accordingly, providing fairness in bandwidth allocation does not guarantee that any users are satisfied, and fair service degradation can actually lead to only unsatisfied users. The authors propose the Dual Queue (DQ) packet scheduler which purposefully degrades the service to some flows in case of congestion so that the other can keep a satisfying service.

Initially, the new concept was used for designing bandwidth allocation mechanisms for fixed networks, i. e. to set the weights of FQ algorithms according to application-specific utility curves instead of plain bandwidth fractions. In [117], a grade-of-service based pricing scheme based on utility functions per application class is proposed, where bandwidth allocation tries to maximise the sum of the utilities of all flows, while any flow entering the system must pay a price according to the quality degradation that it will cause to the flows already in the system. The utility max-min fair bandwidth allocation algorithm proposed in [118] allocates bandwidth to flows so that all receive the same utility, but the utility functions do not have to be the same for all applications.

Kelly et al. [119] formally model the problem of rate allocation for elastic traffic in a fixed network with a single bottleneck as maximisation of the sum of the rate-based utilities of the users subject to capacity constraints. They show that different utility curves express different types of fairness, assuming however that the utility curves are the same for all users. Utility curves of the type $U(x) = a \cdot \log x$ implement proportional fairness: the relative increase in the rate of a flow is always smaller than the sum of the relative decreases in the rates of the other flows. Using utility curves of the type $U(x) = -(-\log x)^a$ gives priority to flows with low allocation and implements max-min fairness when $a \to \infty$: no increase in the rate of a flow can compensate for the decrease in the rate of another flow with an already lower allocated rate. It is shown in [120] how a user can increase its priority by downscaling his utility function and consequently receiving higher bandwidth allocations. They also present an example of a strategy to adapt the scaling of the utility function during a file download according to the already transferred amount of data so that a deadline is met, adapting the bandwidth allocation to another kind of application layer information.

The utility fair bandwidth allocation algorithms above were designed for fixed wired networks, where the total available capacity of the links is known. As was the case for fair queueing, utility-fair bandwidth allocation in wireless networks must deal with the error-prone characteristic of the channel.

Bianchi et al. [99] use the capability of audiovisual applications to adapt to changes in available bandwidth, delivering different perceived quality to the users, in order to build utility functions that map bandwidth to MOS. These are then used to adapt the weight of each flow to the varying bandwidth of the wireless channel so that all users see the same perceived quality. The weights, thus adapted, are used with a packet scheduler that can account for transmission errors, as presented in 4.1. Reference [121, 122] presents the utility based Wireless Fair Scheduling (UWFS) framework for providing utility fairness at the application level in wireless systems. It uses throughput-based utility functions to calculate the weights necessary for the link-layer packet scheduler to provide the utility-fair data rates to the applications.

The problem of utility maximisation for elastic traffic is analysed in [123] for the realistic case of bursty traffic, i.e where users are not always ready to send. A good approximation of the utility maximisation is achieved by setting the weights of a WFQ algorithm in proportion of the relative link qualities elevated to an exponent which defines how strongly the users with better average link quality are given higher priority. To deal with the time-varying link, the weights are changed dynamically according to the link quality variations.

The previous works used utility curves to implement utility fairness in terms of bandwidth allocation, instead of a weight-based fairness. But soon the utility curve was also being used for packet scheduling, i. e. to directly rank packets of different flows waiting for a transmission oportunity. Liu et al. in [124] use utility curves to calculate the importance of transmitting in a certain slot for a user[1]. One of the proposed scheduling policies allocates the channel to the user that has the highest utility in each timeslot, i.e. the average user performance is maximised by locally maximising the user utility.

In [125] Liu et al. propose the U'R scheduler which uses a non-linear utility function of a packet's delay to express decreasing user satisfaction with increasing packet delay. The scheduling metric is the gradient of the utility function multiplied by the usable data rate and expresses the increasing urgency of transmitting a packet when its delay increases.

The Maximum Income Greedy Scheduler (MIGS) [126] chooses the user that maximises network income in each timeslot. But the income is the difference between the price charged per unit bandwidth that the network delivers to a user and a convex function that expresses the penalty due to delivering less than the agreed service, which can be seen as a measure of the user's dissatisfaction.

The Gradient algorithm with Minimum/Maximum Rate constraints (GMR) [127] maximises the system utility using a variation of the U'R scheduler that has the capability to support minimum and maximum rates. The utility is a function of the throughput and its gradient is used as a factor in the scheduling metric. Other factors are the usable data rate in the timeslot, to account for link state, and an exponential function of a token counter, to enforce the minimum and maximum rates. The token bucket is increased at the minimum or maximum rate, depending of whether the flow has received too much or

[1]In the paper, the utility function used as an example is called user performance and is a function of the SNIR.

too few throughput, respectively; it is decreased when the flow is served. If there are too many tokens in the bucket, the flow has been served under the minimum rate and the exponential factor in the scheduling metric gives higher priority to the flow. The opposite occurs if the token counter is negative. By choosing different forms of the utility function, different kinds of throughput fairness are implemented when the utility function is the same for all flows, as above in [119], but here the curves are used to rank packets at each scheduling decision.

The Adaptive Cross Layer scheduler (ACL) [128] strives towards meeting as many deadlines as possible and achieving high throughput by taking advantage of good link conditions. The scheduler decides the service order for entire rounds, where each backlogged flow queue is served once (it was designed for GPRS). The scheduling metric takes into account delay requirements, packet delays and a penalty for missing a deadline. The flows with better links tend to be scheduled first because they use the channel for shorter times and impose lower delays on other flows. This scheduler improves the throughput and deadline misses of both EDF and WFQ in medium and high load situations. However, it may be difficult to express all kinds of QoS requirements in terms of deadlines and it is not clear how other parameters should be set. Furthermore, scheduling a whole round requires a prediction of link quality for the whole round, a much larger prediction horizon than for a single packet.

Khan et al. in [129] propose a framework for the joint optimisation of parameters of several layers to maximise user satisfaction, expressed by application-specific utility curves. The framework uses abstractions that model the behaviour of the different layers and their tunable parameters so that the optimisation can be done independently of the underlying system. An example of the application is shown for delivery of MPEG-4 video streams to several clients in a cell. The cross layer optimisation allocates the channel for an entire GOP so that the sum of the user perceived video quality across all users is maximised. For this purpose, the streaming servers send with the data rate distortion profiles for the GOP (similar to utility curves) that are used with all possible combinations of loss patterns to choose the one allocation that achieves the best video quality. Although the results show an improvement of video quality, the optimisation proposed implies the calculation of the perceived video quality for all users and loss pattern combinations for every GOP—a huge computational overhead which is hardly feasible in real-time, as remarked by the authors.

All these approaches have the ultimate objective of satisfying the user and rely on the mapping of network service to user perceived quality (or dissatisfaction as for the GMR)—the utility function. But in all cases, the utility curves are functions of the same network service for all applications, mostly the throughput, but also delays and deadlines as with the ACL. Also, often the same utility curve is assumed for all users and applications although it is widely known that neither all applications are sensitive to the same service nor the utility functions for different applications have the same form [32, 114, 115].

4.4 Cross Layer Scheduling

The previous section defends the point of view that supporting QoS for flows of different types over a wireless link requires information on the link layer about the application being supported. This information is not available if the layered Internet architecture model is respected. Using details about the application at the link layer is a cross layer approach. So all utility based solutions for bandwidth allocation or packet scheduling presented in the previous sections can be classified as cross layer schedulers.

But cross layer is not only about using application information at the link layer schemes: it is a general designation that applies to any approach that combines in a single mechanism information from more than one layer of the Internet architecture [130]. In this sense, channel-aware and opportunistic scheduling are also a cross layer mechanism, because they use link information typical of the physical layer for link layer packet scheduling.

4.5 Discussion of the Proposed Solution

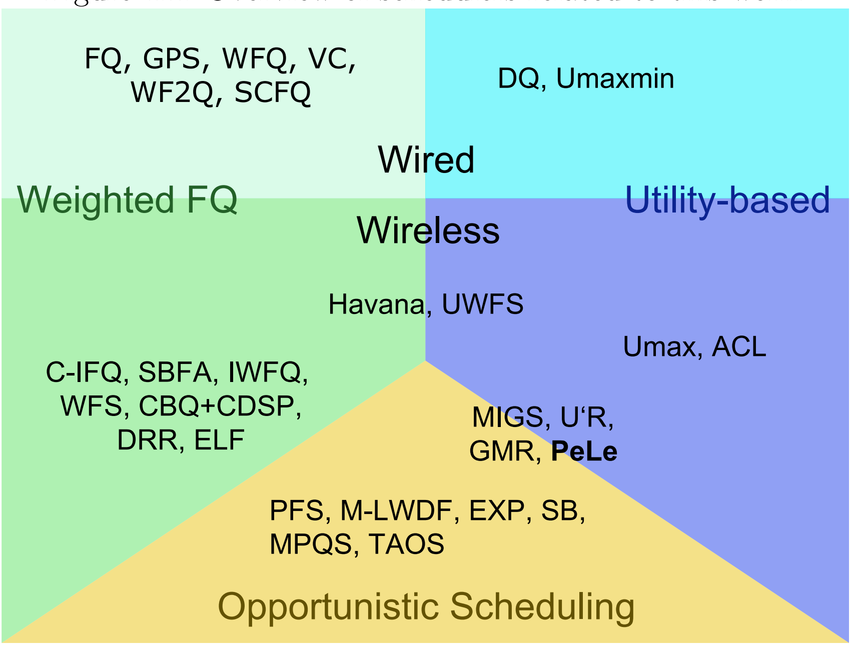
The problem studied in this thesis is that of allocation of the opportunity to transmit on a time-shared wireless channel to different types of flows destined to multiple users with the goal of increasing overall user satisfaction. The solution is based on concepts presented in the works of Shenker [32] and Eckhardt et al. [101] but differs from either approach in fundamental aspects. Shenker introduces the use of utility curves for bandwidth allocation in fixed networks, whereas this work focuses on packet scheduling over wireless channels. The work of Eckhardt et al. is based on similar observations to the ones that motivate this work, but it neither considers adaptive MCS, nor does it make use of utility curves to explicitly express user perceived quality.

The heuristic scheduling metric used by the PeLe scheduler studied combines two aspects: the increase in quality due to allocating the channel to a user and the quality cost for the flows that have to wait for that transmission before competing again for channel access. Recently, similar concepts have been proposed, however separately: the first is used by the U'R [125] and GMR [127] schedulers and the latter by the ACL [128]. The combination of the two aspects, benefit and cost of the packet scheduling decision to the overall quality, thus remains unique.

Due to the need to efficiently use the error-prone wireless channel the solution proposed is channel-aware and takes link quality into account. However, that is not the single factor accounted for, in contrast to opportunistic schedulers [103, 104, 102]. There are also several schedulers that balance a factor expressing link quality in the scheduling metric, with QoS goals, like supporting differentiated delay- and rate-sensitive flows (M-LWDF [109, 110], EXP [109, 111], SB [112]), supporting several differentiated traffic classes [114], or minimising the time necessary to transmit data chunks (TAOS [113]). Contrary to those solutions that use the link quality as a factor in the scheduling metric, the PeLe scheduler uses the link quality only indirectly. In Chapter 8, this shall prove to be a more robust approach in the presence of inaccuracies in link quality prediction.

Figure 4.1 shows an overview of the most relevant schedulers described in this chapter and positions the PeLe scheduler among them.

Figure 4.1: Overview of schedulers related to this work.



Chapter 5

Methodology for Performance Evaluation and Parametrisation of the System Model

The performance of the utility-based multi-user packet scheduler PeLe is evaluated using discrete-event simulations. An analytical evaluation of the proposed solution is not feasible for two reasons. First, there are no models for a traffic mix of different flow types. Traffic models for a coded VoIP flow or an MPEG-4 encoded video stream do exist. Also, there are models for the traffic generated by a single TCP-based application. However, the packet-level behaviour of a TCP flow cannot be theoretically modelled as would be required for the analysis of the packet scheduler because those processes depend on the channel allocation due to the closed-loop congestion control of TCP. Furthermore the traffic arriving at the AP is a superposition of different traffic flows that is not feasible to mathematically characterise. Second, the main metric used to evaluate the performance is the user perceived quality, which cannot be expressed in closed form in terms of network metrics for all applications used; it is calculated offline from packet traces after the simulation ends as will be explained in Section 5.5.

The simulation is parameterised using traces from a measurement campaign for the simulation of the fast fading characteristics of the wireless links. Section 5.1 describes that link measurement campaign, which was carried out in environments typical for WLAN. Section 5.2 presents the simulator developed for the performance assessment in a wireless cell scenario corresponding to the system model described in Chapter 3.2 and Section 5.3 describes the details of the traffic generators of each example application. The details of the schedulers used as reference for performance comparison (channel-aware Round Robin, Proportional Fair scheduler, Modified-Largest Weighted Delay First and Exponential) are presented in Section 5.4. Finally, Section 5.5 describes the calculation of the metrics used to evaluate the performance of each application and their accuracy.

5.1 Measurement of Link Quality in Wireless Local Networks

Wireless local area networks typically match the scenario considered in this thesis (Section 3.2), however the behaviour of the wireless link quality in such scenarios is not well known. Thus, a link measurement campaign using WLAN cards in environments where WLAN coverage is plausible was designed and carried out to study the link behaviour seen by the network interface card (NIC) measurement system. The mechanisms studied in this work are hosted in the link layer of an AP and the link quality indicator available to adaptive mechanisms is the output of a measurement system in the NIC which usually expresses the average signal level over the duration of a received packet. Although this is not the most accurate way to measure link quality, the information delivered by the NIC is often the only indication of link quality available to adaptive link layer techniques.

The link quality traces obtained from these measurements are used to parameterise fast fading in the simulations in Chapter 6, in the evaluation of heuristics for the prediction of link quality in Chapter 7 and in the evaluation of the effects of inaccurate link prediction in the performance of wireless schedulers in Chapter 8. A more detailed description of the measurement setup, data processing and results can be found in [131]. The original and processed traces are publicly available at `www.tkn.tu-berlin.de\~aaguiar\wlan`.

5.1.1 Measurement Setup and Environments

The measurements were performed using two laptops running the Linux operating system (kernel 2.4.17) and equipped with Lucent WLAN cards with the the PRISM2 [132] chipset. The choice of the Network Interface Cards (NIC) was driven by the availability of the source code of the drivers that needed to be changed to perform the measurements. The driver of the WLAN cards [133] was changed so that no acknowledgments were sent, since waiting for link layer acknowledgments would increase the time interval between two channel samples and it was not relevant for the measurements whether packets suffered bit errors. Also, since the signal measured by the NIC could only be retrieved together with the packet, packets with a wrong CRC-check were not discarded to increase the number of signal samples especially at low signal.

One laptop—the Transmitter—sent UDP packets carrying 1 Byte of data every 1.3 ms, which was the minimum possible sending interval. In a test phase before the actual measurements shorter send intervals were tried out but packets never arrived at shorter intervals at the NIC due to delays in the Linux protocol stack. The other laptop—the Receiver—re-sent each received packet (even the ones with wrong CRC), so that there are two signal traces for each measurement run.

Table 5.1 describes the environments where the link quality measurements were made. The criteria for the choice of the measurement scenarios was that they should have the characteristics of environments where WLAN coverage might be available in terms of mobility and surroundings. The environments Mensa and Maths are big halls and thus indoors, with some people moving around. In Mensa there are only the 4 concrete walls of the double-storey room as big reflectors, whereas in Maths the hallway is smaller and has some pilars, gangways at half height and metal stairs. The Carpark environment is the university's parking lot, an outdoor open area surrounded by 3 buildings, some parked cars and no other moving objects. In the Archi environment, the transmitter and receiver are positioned close to a very busy roundabout, so it is an outdoor open area with buildings 50 m away on the one side and many moving cars closer by on the other side. The Road environment is an wide open area, outdoors, with cars crossing the LOS path between transmitter and receiver. In the scenarios described above, both transmitter and receiver are static, whereas in the following the receiver moves. In the Walk scenario, the environment is a wide open outdoor grass-covered area surrounded by bushes and trees, where the receiver is being carried by a person walking in a circle around the transmitter. Both Stadium environments are measured in the same place, a wide open outdoor area in front of the olympic stadium, with concrete on the floor and surrounded on 3 sides by trees, at a distance of more than 50 m. In Stadium1 the receiver is on a bike and in Stadium2 the receiver is carried by a walking person, both in circles around the transmitter.

5.1.2 Processing of the Measurement Traces

Due to internal queues in the protocol stack of the laptops and in the NIC, and to process scheduling by the Linux kernel—factors which were not controllable—the packets were not sent at equidistant times; consequently, the link quality samples were not equidistant and could not be processed as a time series. To obtain equidistant data the original traces were re-sampled at 1 kHz, each sample being the result of the linear interpolation between the two values measured closest to the sampling instant

After re-sampling, the measured signal traces were smoothed to remove noise using a moving average filter of length 30 (the value that showed to be a good compromise between noise reduction and information loss). The discrete Fourier transform of the measured signal amplitude before and after

Table 5.1: Scenarios for the WLAN measurements. R is the number of measurement runs made in each scenario. K is the total number of sample values in each scenario after re-sampling at 1 kHz.

Scenario	R	K	Environment	Mobility
Mensa	7	1088694	Student canteen of the TU Berlin at rush hour	Few people moving between and around Transmitter and Receiver
Maths	4	618950	Foyer of Maths building during intervals between lectures	People moving between and around Sender and Receiver
Carpark	7	1084395	Parking lot surrounded by buildings on 3 sides	No mobility
Archi	7	888249	Busy roundabout	Traffic between Transmitter and Receiver
Road	7	892994	Busy street	Traffic between Transmitter and Receiver
Walk	3	408778	Grass surrounded by trees and bushes	Receiver at Pedestrian speed
Stadium1	2	342189	Wide open area in front of the Olympic Stadium	Receiver at Pedestrian speed
Stadium2	2	341960	Wide open area in front of the Olympic Stadium	Receiver at bicycle speed

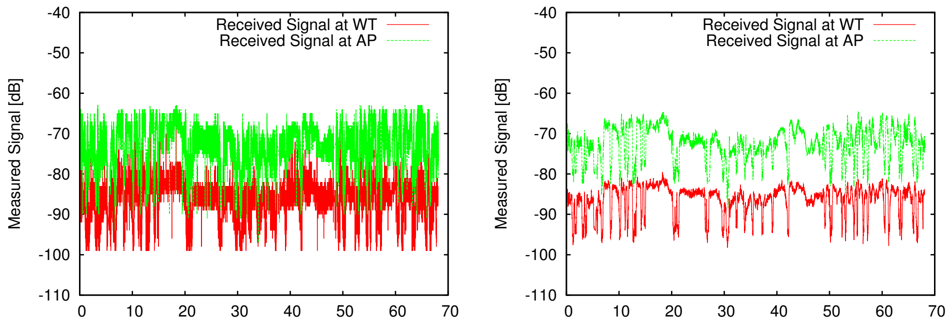
noise filtering can be seen in the figures of reference [131], showing that the power spectrum of the signal up to 10 Hz (a Doppler frequency corresponding to 4.5 km/h) is kept whereas higher frequencies are attenuated. Figure 5.1–a shows the original measured values for one measurement run of the scenario Road and Figure 5.1–b the corresponding equidistant smoothed time series. The time plots of all measurement runs after re-sampling and smoothing can be found in Chapter 3 of reference [131].

Since the objective of the measurement campaign was to characterise the fast variations of the received signal—fast fading—, the measured values were normalised to the mean of each measurement run. Accordingly, the fast fading traces have an average value of 1 (0 dB), as requested by the model in Section 2.1.4.

5.1.3 Statistics of the Measured Received Signal

The different environments and mobility conditions lead to different fast fading behaviour, as can be seen in Figure 5.2 which shows examples of the time variations of the received signal at the Transmitter and Receiver for each environment. In the scenarios Carpark, Maths and Mensa the signal varies less than in Road or Archi, where faster and deeper variations occur due to the

Figure 5.1: Example of measured signal of scenario Road before and after smoothing.

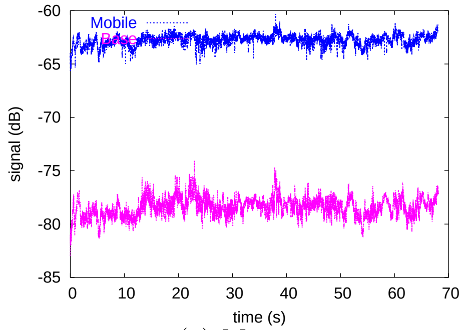


(a) Measured Values

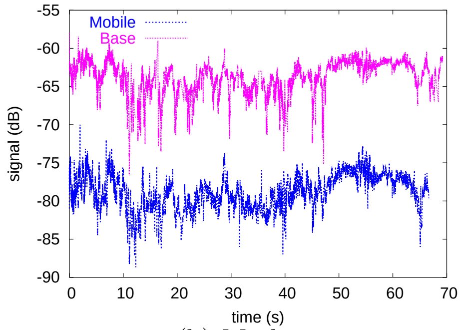
(b) Smoothed Values

fast movement of reflecting objects with respect to the propagation path. The faster mobility of the Receiver in scenario Stadium1 than in Stadium2 can be seen in the faster variations of the signal measured. The variations of the average signal are due to pathloss variations because the Receiver was not moving in a perfect circle around the Transmitter. Because this invalidates a stationarity assumption, the fast fading traces obtained from these scenarios will not be used further.

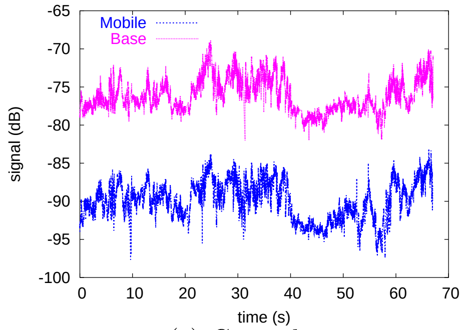
Figure 5.2: Measured received signal (after re-sampling and noise filtering).



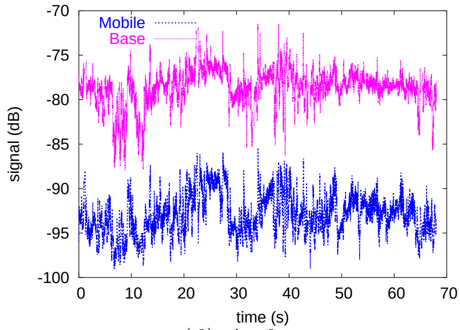
(a) Mensa



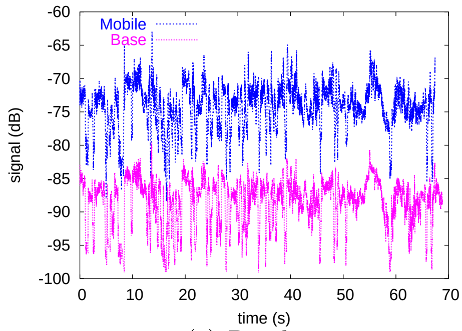
(b) Maths



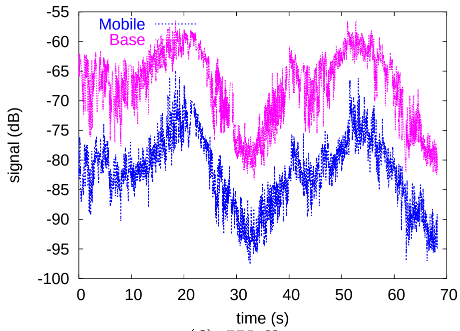
(c) Carpark



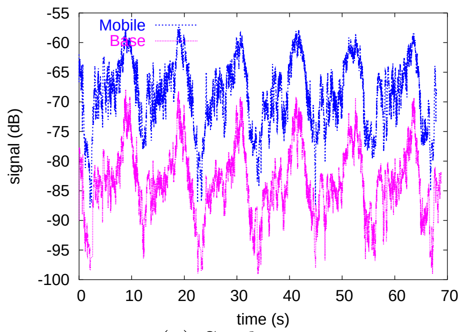
(d) Archi



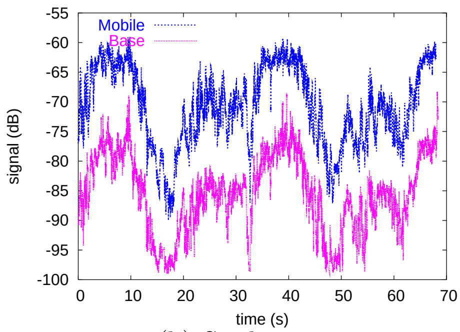
(e) Road



(f) Walk

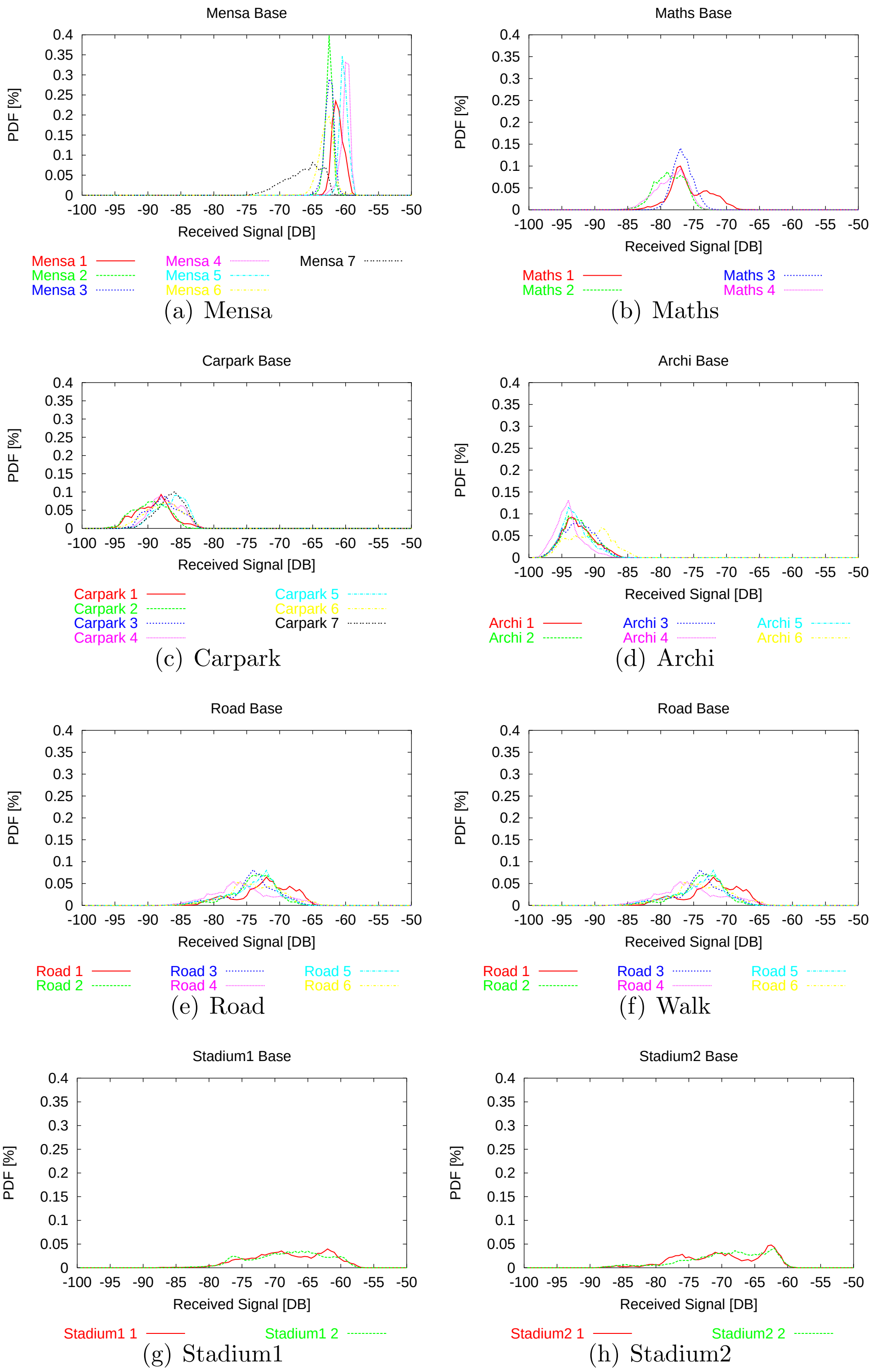


(g) Stadium1



(h) Stadium2

Figure 5.3: Distribution of the received signal.



(a) Mensa (b) Maths

(c) Carpark (d) Archi

(e) Road (f) Walk

(g) Stadium1 (h) Stadium2

Figure 5.4: Variance of the nomalised measured signal for all measurement environments. Each point is the variance of the signal in a single run.

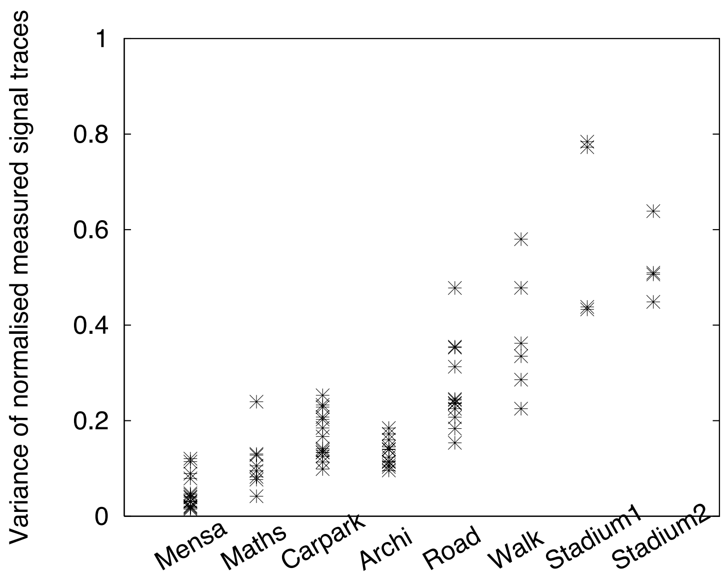


Figure 5.3 shows the histograms of each measured signal trace. The variance of the distribution of the received signal is highest in the Stadium and Walk environments due to the slow fading effects which can be seen in Figure 5.2. In the Road environment, the received signal has a high variance compared to others due to the movement of cars in the propagation path. This produces a large number of multipath components and alternation between LOS and non-LOS. The variance in the scenario Archi is similar to scenario Carpark, but lower than in the previous case. In the first the variations are due to multipath caused by the moving cars nearby, and in the latter due to multipath components from the surrounding buildings and cars which reflect the signal. The indoor scenarios Maths and Mensa show the lowest variance of the received signal amplitude due to the low amount of reflectors and scatterers and little movement. The lack of multipath in scenario Mensa can be seen in the very concentrated distribution of the received signal in Figure 5.2–a.

Figure 5.4 shows the variance of the received singnal in each measurement run per environment. The variance of the normalised signal increases with the mobility and with the amount of reflectors around the propagation path as seen above.

5.2 Simulation Environment

The scenario used for the performance evaluation is implemented using the C++-based environment for communications and networking simulation Omnet++ [134]. It comprises of a circular cell of radius $R_{\text{cell}} = 500$ m with N evenly distributed WT, where each WT receives data directly from the AP. The raw channel bandwidth is W and the Nyquist signalling rate is assumed, so that the symbol duration is $t_S = \frac{1}{2W}$ [19]. AWGN is assumed at all receivers with power NPwr=-115 dBm, and includes both background thermal noise and co-channel or neighbour channel interference.

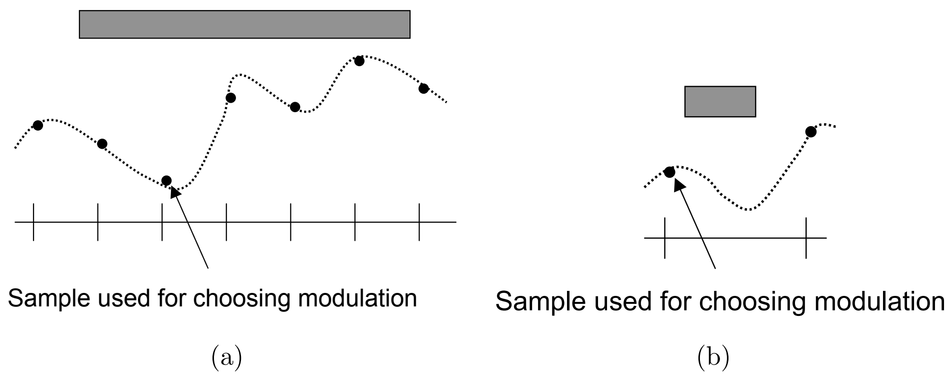
An underlying asynchronous time division channel multiplexing is used, i. e. the AP can start transmitting a packet at any time instant (as long as the channel is free) and the packet can have any duration. Each packet sent on the physical channel has the overhead of a physical (PHY) and a medium access control (MAC) header of lengths h_{PHY}=172 bits and h_{MAC}=192 bits, (values taken exemplarily from the IEEE 802.11b standard [26]). The MAC and PHY headers are sent using the lowest available modulation, and represent a constant overhead of $464/2W$ [s] per transmitted packet.

5.2.1 Model of an Adaptive Modulation Scheme

An adaptive uncoded M-PSK modulation scheme is considered with $M \in \{2, 4, 8, 16, 32, 64\}$. Although M-PSK may not be a realistic assumption for high M due to technological limitations to its practical implementation, it is the only family of digital modulations for which exact closed form expressions for the relationship between per symbol signal-to-noise ratio and BER exist [19]. A PER requirement—PER_{max}—is defined for each flow type; the corresponding maximum acceptable BER—BER_{max}— is calculated according to Equation 2.10 assuming independent bit errors in a packet, and the modulation to use is chosen accordingly. The data rate R of the packet's payload is directly proportional to the usable modulation depth $k = \log_2 M$—$R = 2W \log_2 M$— and these terms will be used interchangeably from now on.

Because the fast fading traces are discrete in time and sampled at 1 kHz (see Section 5.2.2), some packets stretch over several fast fading samples. In this case, the modulation is chosen using the lowest predicted SNR value within the packet duration (Figure 5.5–a). If the packet lasts less than the interval between two fading samples, the sample closer to the middle of the packet is used as the predicted SNR to choose the modulation (Figure 5.5–b). First, the adaptation module checks whether the headers can be sent using the most robust modulation, BPSK. Then, the highest available modulation that meets the PER_{max} requirements at the minimum predicted SNR in the duration of the payload is looked up. If the packet's PER_{max} cannot be met, the packet is

Figure 5.5: Fast fading samples used by the AP to choose the modulation in case the packet stretches over several samples and in case it lays in the interval between two samples.



not eligible for transmission. The conversion from PER_{max} to BER_{max} is done using Eq. 2.10 using the amount of bits in the header or payload, respectively.

5.2.2 Model of the Wireless Link

The wireless link between the AP and each WT is modelled by the received SNR. The average pathloss is calculated according to expression

$$h_{\text{pathloss}} = H_0 - 10 * \alpha * \log d, \tag{5.1}$$

where d is the distance to the AP, the pathloss exponent α=2.5 and a reference attenuation of H_0=-46.7 dB at 1 m. The simulations assume no WT mobility, so the h_{pathloss} remains constant throughout a simulation run. Shadowing is not considered and $h_{\text{shadowing}}$ =1.

The fast fading behaviour is obtained from the WLAN measurement traces described above, and the received signal calculated from

$$S_{\text{received}} = \text{TxPwr} + h_{\text{pathloss}} + R_{\text{fading}}. \tag{5.2}$$

R_{fading} is used in the expression as the measurement traces contain the normalised variations of the received signal, and not the gain. To obtain the SNR, for each received signal value a noise value N_{receiver} is obtained from the realisation of a Gaussian distributed random variable with 0 mean and variance NPwr. The SNR is calculated as the difference of the two:

$$SNR = S_{\text{received}} - N_{\text{receiver}}. \tag{5.3}$$

5.2.3 Model of the Receiver

At each WT the receiver checks whether the packet suffered errors according to the received SNR values, which are calculated from the fast fading samples available in the duration of the packet transmission. The fading samples corresponding to the packet start and end are used as well as all other samples within the packet's duration. The packet is checked for errors according to the following procedure, which is done separately for the PHY and MAC headers and the payload as they can use different modulations:

1. the average between every two fast fading samples for the packet is calculated—these are the received signal samples that can be seen in Figure 5.6;

2. for each received signal sample, a noise sample is generated from a gaussian distribution with 0 mean and variance NPwr [dB] and the SNR calculated as the ratio of the two;

3. the amount of bit errors for all intervals between fading samples is evaluated as a realisation of a binomial random variable with n the number of bits in the interval and p the BER corresponding to the SNR obtained in the previous step according to the used modulation[1];

4. the packet is correctly received if no errors occur.

Figure 5.6: Received signal samples, calculated from the channel samples, used at the WT to check for packet errors. For each signal sample a noise sample is generated. Constant SNR is assumed between two samples.

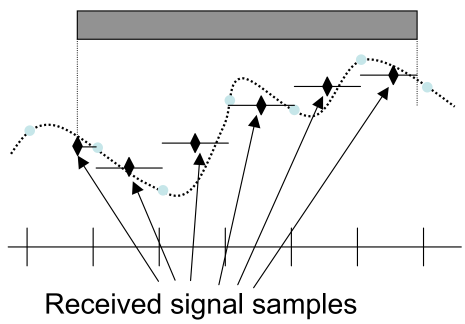
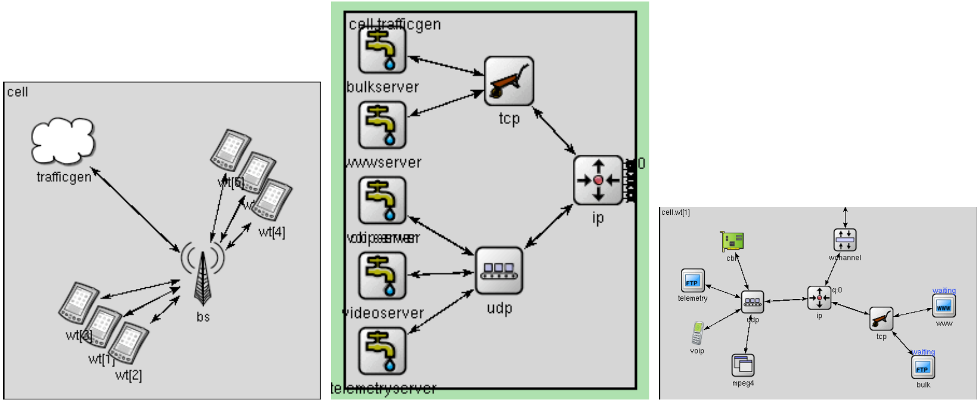


Figure 5.7: Simulated cell with traffic servers co-located with the AP and clients located in the WT.



(a) Cell (b) Traffic Generator (c) WT

5.3 Traffic Generators

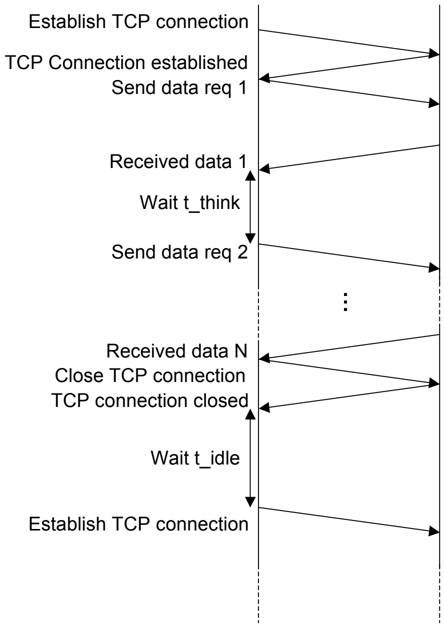
The traffic used in the performance evaluation is a mix of different applications in client-server configuration. The servers lie in a host co-located with the AP (Figure 5.7–a and b) whereas the client applications are located in the WT (Figure 5.7–c). The client applications start the data flows by sending requests to the corresponding server at random start times evenly distributed in the first 2 seconds of each simulation run (and are different for each run). In the scenario, the uplink application traffic consists of the clients' requests, TCP SYN, FIN and ACK packets and eventual link layer acknowledgements. The downlink traffic consists of the superposition of the data flows generated by the servers. Only downlink traffic shall be considered in the evaluation of the PeLe scheduler. The rest of this section describes the details of the downlink traffic generated by each application server.

Bulk File Download The client-server connection model is the same for both TCP-based traffic types and is shown in Figure 5.8. The TCP connection is initiated by the client, which sends a request for a certain amount of data D_{Bulk} after the TCP connection is establishes. This triggers the server to answer by sending the amount of data requested. When all data is received, the client can send another request after a waiting period t_{think} or close the TCP connection. After closing the TCP connection, the client waits for an idle period t_{idle}, establishes a new TCP connection and re-starts the process

[1]This assumes that the channel remains constant between two samples and that the bit errors are not correlated in that interval.

described.

Figure 5.8: Model of client behaviour for TCP-based traffic.

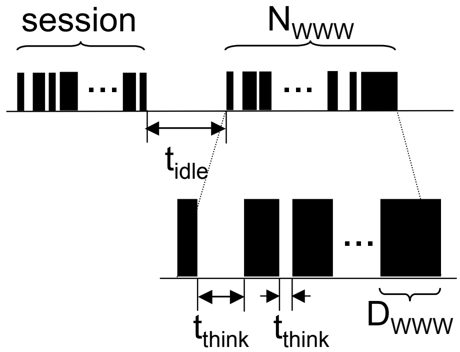


For Bulk file downloads each connection contains a single request with demand for D_{Bulk} =1 MBytes of data and the idle interval between two connections is t_{idle}=0 s, as this is background traffic that should be always running.

WWW traffic The behaviour of a WWW client is modelled as a sequence of sessions consisting of one or more pages, as proposed in [2, 135] and illustrated in Figure 5.2. In the model, a session corresponds to a TCP connection. During the session, the user visits several web pages and reads them—the sequence of pages separated by the time to read is modelled by the sequence of data requests and the think time by t_{think}. After visiting several pages—N_{WWW}, the user stays idle for t_{idle} and then starts a new session.

The values for the model parameters proposed in [2] were extracted from real-world traces of WWW user behaviour and are shown in Table 5.2. The values of the think and idle times are large and would require very long simulations to achieve a significant number of transmitted pages, while the WWW server and clients would be inactive waiting for long periods of time. To keep the necessary simulated time low and to increase the number of pages received by each client in each simulation run, the average think time t_{think} is reduced to a fourth of the value in Table 5.2 by reducing the scale parameter of the gamma distribution to $\beta = 1.044$.

Table 5.2: Parameters of the WWW client model [2].



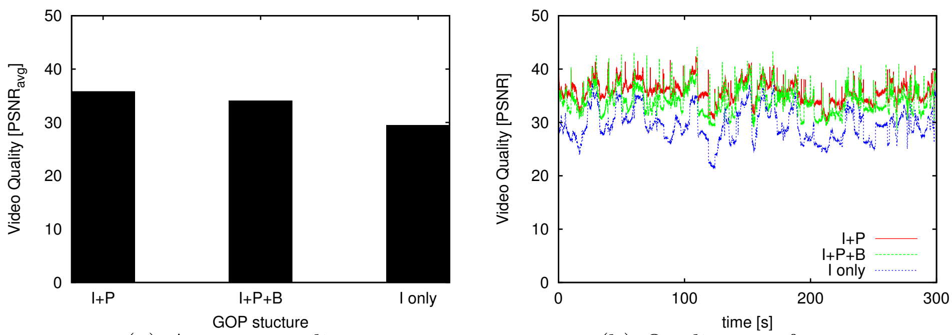
Parameter		Distribution	Dist. Parameters	
Idle Interval [s]	t_{idle}	Exponential	$\mu = 5$	
# Pages per Session	N_{WWW}	Lognormal	$\mu = 25.81$	$\sigma = 78.75$
Page Size [Bytes]	D_{WWW}	Pareto	$\alpha = 1.76$	$\beta = 30000$
Think Time [s]	t_{think}	Gamma	$\alpha = 8.448$	$\beta = 4.177$

Voice over IP Upon receiving a request from a client, the VoIP server starts sending a constant bit rate flow (CBR) which models G.729 traffic [136], with two 10 Byte voice samples (20 Byte packets) every 20 ms, generating 8 kbps at the application layer. To each packet a 12 Byte header is added, corresponding to an RTP protocol header [35]. This flow has a data rate of 47.2 kbps on the wireless channel, when the UDP, IP, MAC and PHY headers are added to the application data. Once started, a VoIP flow lasts until the end of the simulation.

Video on Demand The VoD client issues a request to the server which replies with a stream of packets following the pattern saved in a trace file and carried by RTP and UDP. The tracefile is obtained offline by sending a real H.264-encoded video over an unloaded network. The video to stream is a 5 minute cut of a football match, which has a lot of movement and is a realistic and demanding content. It is encoded using the H.264 codec [57] codec (which has become part of the MPEG-4 standard as Part 10: Advanced Video Coding [137]) in CIF-format (352x288 pixel) at 25 frames per second (fps) to achieve a target average bitrate of 256 kbps at the application layer.

The quality of the H.264-encoded video stream depends on the encoding options, among others on the structure of the Group of Pictures (GOP) used: the more frames can be encoded as B-frames and have a small size, the larger the I-frames can be and the better quality they can have. For a pre-defined amount of frames per second and target bitrate, a video encoded using only I-frames, where each frame must carry all the information for itself, has lower quality than when some frames can be predictively encoded. Since less data is needed for the predictively encoded frames, the I-frames can be larger and be encoded with higher quality each. However, this leads to higher variability of the bitrate of the encoded stream. Figure 5.9 shows the average quality and the quality per frame of the sample video encoded using different GOP, confirming that a video encoded with P-frames has better quality than a video encoded only with I-frames, for a same frame and average bitrates. Further, adding a B-frame between P-frames does not significantly increase the quality of the encoded video.

Figure 5.9: Quality of H.264-encoded video (before transmission) depends on GOP.



(a) Average quality (b) Quality per frame

Since P- and B-frames depend on I-frames to be decoded, the loss of an I-frame means that no futher frames in a GOP up to the next I-frame can be decoded. This means that more losses can be tolerated without significant decrease of the video quality (when compared to the quality before transmission) if the GOP consists only of I-frames. But since the quality before transmission is much lower when only I-frames are available, it is more realistic to use a GOP with P-frames. So, according to the information available, a GOP of 250 frames (10 s) with only I and P frames is chosen: an I-frame followed by 249 P-frames. Although the choice of the most appropriate encoding parameters for wireless transmission plays an important role in the performance, a thorough study of the subject is beyond the scope of this work.

The size of the packets varies depending on the size of the frames to send and on the MTU, so that the exact overhead per packet cannot be calculated. For the video user, and assuming a packet size of 1460 Bytes, the overhead caused by the headers of the protocol stack is approximately 3%, the average and maximum data rate of the VoD flow at the physical layer is 264 kbps and 370 kbps, respectively.

5.4 Reference schedulers

5.4.1 Channel-aware Round Robin—caRR

The Channel-aware Round Robin (caRR) scheduler serves flows one after the other but taking into account link quality: if the link towards the destination of a flow is experiencing bad quality ($\mathrm{PER}_{\mathrm{max}}$ cannot be guaranteed), the queue is not eligible to transmit and the next flow in row with a good link is served. The flow experiencing bad channel quality has to wait for the next round to be served. The caRR gives fair opportunity of channel access to all WT disregarding link state; however, WT with a lower average SNR experience a bad link more often than WT with a higher average SNR and have actually fewer opportunities to access the channel.

5.4.2 Proportional Fair Scheduler—PFS

The Proportional Fair Scheduler (PFS) [103, 104, 102] introduced in Section 4.2 is a channel-aware scheduler that serves the flow towards the WT with the highest ratio of current possible data rate to average data rate in the recent past, according to

$$i:\ \mathrm{argmax}_i \frac{R_i}{T_i}, \tag{5.4}$$

where $R_i = 2Wk_i$ is the data rate that can be currently used for transmission to the destination of flow i. T_i is an exponentially weighted average of the data rates used in the transmissions of flow i in recent past, calculated according to the low-pass filter

$$T_i(t) = \begin{cases} T_i(t-1)(1-1/t_C) + R_i * 1/t_C & \text{, flow i is backlogged} \\ T_i & \text{, flow i is not backlogged} \end{cases}. \tag{5.5}$$

The current possible data rate R_i is calculated using the prediction of link behaviour as described in Section 5.2.1; in case no modulation can guarantee $\mathrm{PER} < \mathrm{PER}_{\mathrm{max}}$, the flow is not eligible for transmission due to bad link quality. This scheduler takes advantage of the periods of best link quality of each WT—often called “riding the peaks” of link quality, achieving a high overall throughput and guaranteeing that all flows access the channel, but there it does neither provide delay guarantee nor flow or packet differentiation.

5.4.3 Modified Largest Weighted Delay First—M-LWDF

The Modified Largest Weighted Delay First (M-LWDF) [110, 109] proposes an enhancement to the PFS scheduler above by differentiating between two types of QoS requirements:

users with stochastic delay constraint $P(W_t > \delta_i) \leq t_i$, where W_t is the steady-state packet delay, δ_i is the acceptable delay threshold and t_i is the maximum probability of exceeding it;

users with rate constraint $\rho_i > r_i$, where ρ_i is the throughput of user i and r_i the minimum required.

This scheduler improves the performance of the PFS by trying to meet delay requirements for delay-sensitiv flows, like audiovisual.

The M-LWDF scheduler chooses the flow

$$i : \operatorname{argmax}_i \gamma_i W_i R_i,$$

where W_i is the largest delay of a packet in queue i, R_i is the data rate that WT i can support according to the predicted link state and γ_i is a positive constant, whose value can be used to control the delay distribution of the flows.

Setting $\gamma_i = a_i/T_i$ with $a_i = -\log \delta_i/t_i$ and T_i is the low-pass filtered data rate of user i according to Eq. 5.5, allows the support of the QoS requirements of a maximum number of users. The theoretical background for these values can be found in [109].

To provide throughput guarantees to flows with rate constraint requirements, the value of W_i in the flow weight is not the maximum delay of a packet in the queue, but the delay of the longest waiting token in a virtual token bucket, where tokens arrive at a constant rate r_i and are removed as the flow is served. The parameter $\gamma_i = a_i/r_i$ can be used to control the timescale for providing the throughput guarantees: the higher the value of γ_i relative to the other flows, the tighter the timescale on which the minimum throughput can be provided.

5.4.4 Exponential-Fair Scheduler—EXP

The Exponential-Fair (EXP) scheduling algorithm [130] supports the same two types of QoS requirements as above, but chooses the flow i

$$i : \operatorname{argmax}_i \gamma_i R_i \exp \frac{a_i W_i - a\bar{W}}{1 + \sqrt{a\bar{W}}},$$

where $a\bar{W} = 1/N \sum a_i W_i$ is a weighted average of the flow's delays, and $\gamma_i = a_i/T_i$, $a_i = -\log \delta_i/t_i$ as above. The exponential term becomes very big for flows experiencing weighted delays bigger than the weighted average by more than $\sqrt{a\bar{W}}$. This gives them very high priority and tries to balance the delays.

To support rate-sensitive flows a virtual token bucket is used, as with the M-LWDF, and the delay of the oldest token in the bucket is used instead of the packet's delay to calculate the scheduling metric. The weight a_i can be used to control the priority of rate-sensitive flows with respect to others, as with the M-LWDF.

When compared to the PFS and the M-LWDF, this scheduler achieves the best fairness in the service delivered to both types of users, at the cost of a lower total cell throughput. The value of the parameter a_i must be tuned according to the traffic mix and channel bandwidth W, a downturn of the M-LWDF and EXP schedulers. For the results obtained in this thesis, the values of the a_i were calibrated using extensive simulations where the value of a_{Bulk} was varied while keeping the value of a_{VoIP} constant. The results show that acceptable VoIP quality is obtained only if $a_{\text{Bulk}} \leq 0.001$, so a_{Bulk} is set to 0.001 for the reference schedulers EXP and M-LWDF.

5.5 Calculation of the Metrics and their Significance Level

The performance is evaluated for each WT individually depending on the type of flow that each user receives, so the accuracy level of the results must be calculated for the perceived quality metric relevant for each user. Because the performance is measured on perceived quality and calculated offline from packet traces for audiovisual flows, it is impossible to determine at simulation time the accuracy of the metrics. However, for simulations with VoD traffic the duration of the simulation—Δ—is limited by the duration of the video to transmit, because after the VoD transmissions finish the traffic conditions change drastically. The duration of simulations is set to Δ=350 s. If is not enough to achieve a significant result, several simulations under the same conditions but with different random seeds must be run until an acceptable accuracy level over the link conditions is reached. In this case, the different X runs (if necessary) are equivalent to a single long simulation run of duration $X \cdot \Delta$. Unless otherwise mentioned, $X = 5$ runs is used.

Additionally, it is necessary to verify that the results are valid for different placements of the WT in the cell, i. e. that the average SNR of each WT and the relation between the average SNR of different WT does not influence the performance of the scheduler for the different traffic classes. For this purpose, several scenarios with different average SNR for each WT are simulated for the same traffic mix. In each run the WT are evenly distributed in the circular cell with distances to the AP ranging between 5 and 500 m. These different simulation runs are not independent repetitions of the same scenario, as was the case in the previous paragraph, where the average SNR stayed constant across different runs. The simulation of Y different runs in this case is equivalent to simulating the same cell with $Y \cdot N$ WT and bandwidth $Y \cdot W$.

The next paragraphs explain the calculation of the perceived quality metrics and respective confidence intervals for each traffic type; the calculation of the aggregate metric over simulations with different SNR per WT and respective accuracy is described at the end.

5.5.1 Goodput of Bulk Clients

For bulk data transfer clients the user perceived quality is measured as bulk transfer capacity [47], and it can be measured using several possible methods [138]. In this work, it shall be computed measuring the response time of several file transfers, similarly to [139] and the measurement tools NetPerf and Iperf. Because the goal is to measure the long term data transfer rate, the files to transmit should be large enough for the congestion algorithm to be in the steady-state, which is why the requested file size was set to 1 MB. After the reception of each entire file, a sample of the data transfer rate is calculated by dividing the size of the file by the time elapsed etween file request and its arrival, Δt_{Bulk}:

$$g_i = \frac{D_{\text{Bulk}}}{\Delta t_{\text{Bulk}}} \tag{5.6}$$

The BTC is calculated as the average of the transfer rates of the single files and the 95% confidence interval calculated according to [140]

$$\bar{g} \pm t_{n-1,1-\alpha/2}\sqrt{\frac{S_g^2}{n}} \tag{5.7}$$

where n is the number of samples, $\alpha = 0.05$ and

$$\bar{g} = \sum_{i=1}^{n} g_i,\ S_g^2 = \frac{\sum_i (g_i - \bar{g})^2}{n-1} \tag{5.8}$$

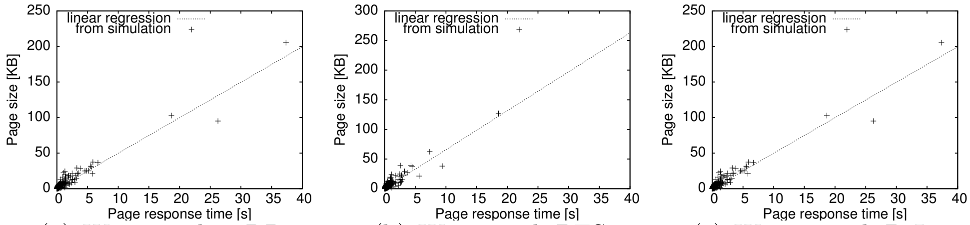
are the mean and variance of the observed goodput samples.

5.5.2 Page Goodput for WWW Clients

The metric for WWW traffic should also be the long term average transfer rate seen by the application, measured by the BTC, but its calculation is not straightforward. According to the traffic model used (Section 5.3), the requested page sizes vary according to a Pareto distribution which has a high variability and generates some very short data transfers. As is remarked in [138] for short pages, the steady state of the underlying flow and congestion control mechanisms is not reached, as required for the measurement of BTC. Thus, the goodput seen by the application must be measured differently.

The quotient between the page size and the page response time—goodput per page—expresses the speed of the service independently of the page size and including the TCP-specific dynamics, so it seems like an appropriate metric for the evaluation of the quality perceived by user of a WWW client application. Table 5.3 shows the correlation coefficient

Figure 5.10: Response time vs. page size for WWW clients.



(a) WT 8 with caRR (b) WT 4 with PFS (c) WT 10 with PeLe

$$r_{xy} = \frac{\sum_i x_i y_i - n * \bar{x}\bar{y}}{\sqrt{(\sum_i x_i^2 - n * \bar{x}^2)(\sum_i y_i^2 - n * \bar{y}^2)}}$$

between the response time and the page size for each WT in a 900 s simulation run with channel bandwidth W=300 kHz and 12 WWW clients. Independently of the scheduler used, the correlation between the page size and the response

Table 5.3: Correlation coefficient between page response time and page size per WT in a 900 s simulation with 12 WWW clients and W=300 kHz.

WT	caRR	PFS	PeLe
0	0.9329	0.9522	0.9356
1	0.8981	0.9085	0.9565
2	0.9142	0.9532	0.9324
3	0.9425	0.9472	0.9476
4	0.8958	0.9340	0.9476
5	0.7947	0.8742	0.8496
6	0.9484	0.9830	0.9033
7	0.8787	0.8150	0.9341
8	0.9809	0.9410	0.8868
9	0.9236	0.8874	0.9552
10	0.9682	0.9085	0.9707
11	0.8981	0.9104	0.9565

time is very high, and the page goodput can be captured by a linear regression. Figure 5.10 shows the samples pairs page size versus the page response time for some exemplary WT together with the corresponding linear regression.

These examples support the usage of the average page goodput, the slope of a linear regression with x the page response time and the page size of the WWW pages downloaded by a user during a simulation run, as the perceived quality metric for WWW clients. According to [141], and assuming that the residuals of the regression have a normal distribution, the 95% confidence interval for the slope of the regression line can be calculated as

$$g \pm s_{y|x} \cdot t_{n-2,1-\alpha/2} \cdot \sqrt{\frac{1}{\sum_{i=1}^{n}(x_i - \bar{x})^2)}}, \tag{5.9}$$

where

$$s_{y|x} = \sqrt{\left[\left(\frac{n-1}{n-2}\right) s_y^2(1 - r_{xy}^2)\right]} \tag{5.10}$$

and $t_{n-2,1-\alpha/2}$ can be read from Table T.1 in [140] with n the number of downloaded pages and $\alpha = 0.05$.

5.5.3 R-factor for VoIP Clients

The perceived quality metric for VoIP calls is the R-factor, and is evaluated using the perceptual quality model presented in [71, 64] and standardised in recommendation ITU-T P.833 [142]. This quality model takes into account the end-to-end packet delay, the speech quality and adaptive playout buffers. The procedure for VoIPquality evaluation is as follows:

1. a speech sample of 8 s duration is encoded with the ITU reference implementation of the G.729 codec;

2. the VoIP server sends CBR traffic corresponding to that codec (see Section 5.3);

3. the receiver saves a trace of the received packet with the timestamps of their arrival at the application buffer;

4. the playout time of each packet is calculated according to the Moon adaptive playout scheme [54], and the mean transmission delay is calculated[2];

5. an ITU reference implementation of the G.729 speech decoder generates a degraded version of the original speech sample, using the calculated playout times;

[2] A delay of 150 ms is added to simulate a distant conversation partner

6. the PESQ calculates the speech quality of the degraded sample;

7. the MOSvalue and the mean delay are used to calculate the R-factor [71].

The procedure is used with traces of the packets received by each VoIP client during the simulation run. Since the VoIP call is longer than the sample duration, the R-factor is calculated for non-overlapping windows of 8 s and the perceived quality of a call is measured on the mean R-factor over the call duration, with accuracy calculated according to Equations (5.7) and (5.8). The encoded voice sample has an R-factor of 73.5168 before transmission—the maximum achievable value for the call quality after transmission.

5.5.4 PSNR for VoD Clients

The DIV of the video stream received during a simulation run was defined as the metric for perceived quality of VoD clients in Section 2.4.2. It is calculated using the free Evalvid framework [143] from packet and frame traces obtained during a simulation run. The same video described in Section 5.3 is used for all clients in all simulation runs.

The quality of the transmitted video is based on the PSNR, which is calculated using the free Evalvid framework [143] as follows:

1. an original video sequence is encoded using the ffmpeg H.264 codec [57, 144] (this happens before the simulations and is done only once);

2. the video server sends the encoded video as described in Section 5.3;

3. the receiver saves a trace of the received packets with the timestamps of their arrival at the application buffer;

4. the received video file is reconstructed from the received packets and decoded using a playout buffer which drops packets according to their arrival timestamps if necessary;

5. the received decoded video file is then compared to the original video file and the PSNR calculated.

The PSNR is calculated for each frame of the video for every VoD client in each run, based on packet traces saved during the simulations. So, after the PSNR evaluation there is a time sequence of PSNR values for each VoD client, which is converted into the DIV after a mapping to MOS as follows:

1. the PSNR time sequence of the video received at the client is converted into the corresponding MOS sequence according to the table in Section 2.4.2;

2. the MOS time sequence of the received video is compared to the MOS sequence of the encoded video that was transmitted by the VoD server: the DIV is the percentage of frames in each (non-overlapping) interval of 500 (20 s at 25 fps) which has a lower MOS than the corresponding interval in the transmitted video.

A video is considered good if the percentage of intervals with DIV higher than 10% is less than 10%, i. e. a video is acceptable if at most 10% of the 20 s intervals have more than 10% of the frames with a worse MOS than the transmitted video.

5.5.5 Total Cell Throughput

Additionally to the application specific metrics described above to evaluate the user perceived quality of each application, the average overall cell throughput—$\bar{\rho}$, is used to evaluated the trade-off between quality provided and resource utilisation.

The samples values of the cell throughput—ρ_k—are gathered by adding up the amount of bits B_k sent on the channel in every period of Δt_{batch} and calculating the corresponding throughput $\rho_k = \frac{B_k}{\Delta t_{\text{batch}}}$, where B_k is the amount of bits sent over the channel during batch interval k. At the end of the simulation run the mean value and its 95% confidence interval are calculated as in Equations (5.7) and (5.8)

$$\bar{\rho} \pm t_{n-1,1-\alpha/2}\sqrt{\frac{S_\rho^2(n)}{n}}. \tag{5.11}$$

To support some reasonings related to the efficiency of channel usage two other metrics shall be used. The time overhead, which is calculated as

$$\frac{\text{channel time occupied with transmission of PHY and MAC headers}}{\text{total busy channel time}}, \tag{5.12}$$

and the average number of bits per symbol used for payload transmission $\bar{k}$.

5.5.6 Average Results over Several Runs

The average quality obtained in one simulation run is significant for varied fading conditions, but to obtain results that are significant over diversified relative positions of WT with respect to each other and to the AP it i necessary to average the results over several runs with different WT positions. The average quality for a certain traffic type is averaged over all clients and runs and the corresponding accuracy calculated from the variance, assuming that the perceived qualities of the WT are statistically independent.

5.6 Summary of Parameters

The following tables show an overview of the parameters used in the performance evaluation in the next chapters.

Table 5.4: System Parameters

Parameter	Description	Value
R_{cell}	Cell radius	500 m
N	Number of WT	varies
d	Distance to AP	varies
α	Pathloss exponent	2.5
A_0	Pathloss at 1 m	46.7 dB
W	Bandwidth	varies
NPwr	Noise power	-115 dBm
TxPwr	Transmission Power	20 dBm
MSS	Maximum segment size	1460 Bytes
MTU	Maximum transfer unit	1500 Bytes
$L_{\text{header, RTP}}$	RTP header	12 Bytes
$L_{\text{header, UDP}}$	UDP header	8 Bytes
$L_{\text{header, TCP}}$	TCP header	20 Bytes
$L_{\text{header, IP}}$	IP header	20 Bytes
$L_{\text{header, MAC}}$	MAC header	36 Bytes
$L_{\text{header, PHY}}$	PHY header	21 Bytes
t_C	Time constant of rate low-pass filter	100 ms

Table 5.5: Traffic Parameters

Parameter	Description	Value
D_{bulk}	Amount of data in a bulk request	1 MByte
N_{www}	Amount of requests in a session	1
$t_{\mathrm{waiting,bulk}}$	Waiting time between two bulk requests in a session	—
$t_{\mathrm{idle,bulk}}$	Idle time between two bulk sessions	0 s
D_{www}	Amount of data per item in a web page	Pareto(1.76,30000)
N_{www}	Amount of requests per web page	logn(25.81,78.75)
$t_{\mathrm{think,www}}$	Waiting time between two requests in a session	γ(8.45,1.04)
$t_{\mathrm{idle,www}}$	Idle time between two web pages	exp(5)
$\bar{r}_{\mathrm{VoIP}}$	Average PHY data rate of a VoIP flow	47.2 kbps
L_{VoIP}	Length of a VoIP packet (on PHY)	944 bits
$\bar{r}_{\mathrm{VoD}}$	Average PHY data rate of a VoD flow	264 kbps
$r_{\mathrm{VoD}}^{\mathrm{max}}$	Maximum PHY data rate of a VoD flow	370 kbps

Table 5.6: FM Parameters for each flow type.

Parameter	Description	Value
$q_{\max,\text{Bulk}}$	Maximum size of queue for Bulk traffic	64 kB
u_{Bulk}	Utility curve for Bulk traffic	Eq. 3.5
θ_{Bulk}	Interval for FM resources tracking for Bulk traffic	2 s
$\text{PER}_{\max,\text{Bulk}}$	Maximum acceptable PER for Bulk traffic	10^{-4}
$q_{\max,\text{WWW}}$	Maximum size of queue for WWW traffic	64 kB
u_{WWW}	Utility curve for WWW traffic	Eq. 3.6
θ_{WWW}	Interval for FM resources tracking for WWW traffic	0.544 s
$\text{PER}_{\max,\text{WWW}}$	Maximum acceptable PER for WWW traffic	10^{-4}
$q_{\max,\text{VoIP}}$	Maximum size of queue for VoIP traffic	Eq 3.7
u_{VoIP}	Utility curve for VoIP traffic	Tab. 3.1
θ_{VoIP}	Interval for FM resources tracking for VoIP traffic	200 ms
$\delta_{\max,\text{VoIP}}$	Maximum acceptable delay per packet	50 ms
$\text{PER}_{\max,\text{VoIP}}$	Maximum acceptable PER for VoIP traffic	10^{-3}
$q_{\max,\text{VoD}}$	Maximum size of queue for VoD traffic	64 kB
u_{VoD}	Utility curve for VoD traffic	Eq. 3.8
θ_{VoD}	Interval for FM resources tracking for VoD traffic	1 s
$\text{PER}_{\max,\text{VoD}}$	Maximum acceptable PER for VoD traffic	$5 \cdot 10^{-3}$

Chapter 6

Performance of the PeLe Scheduler

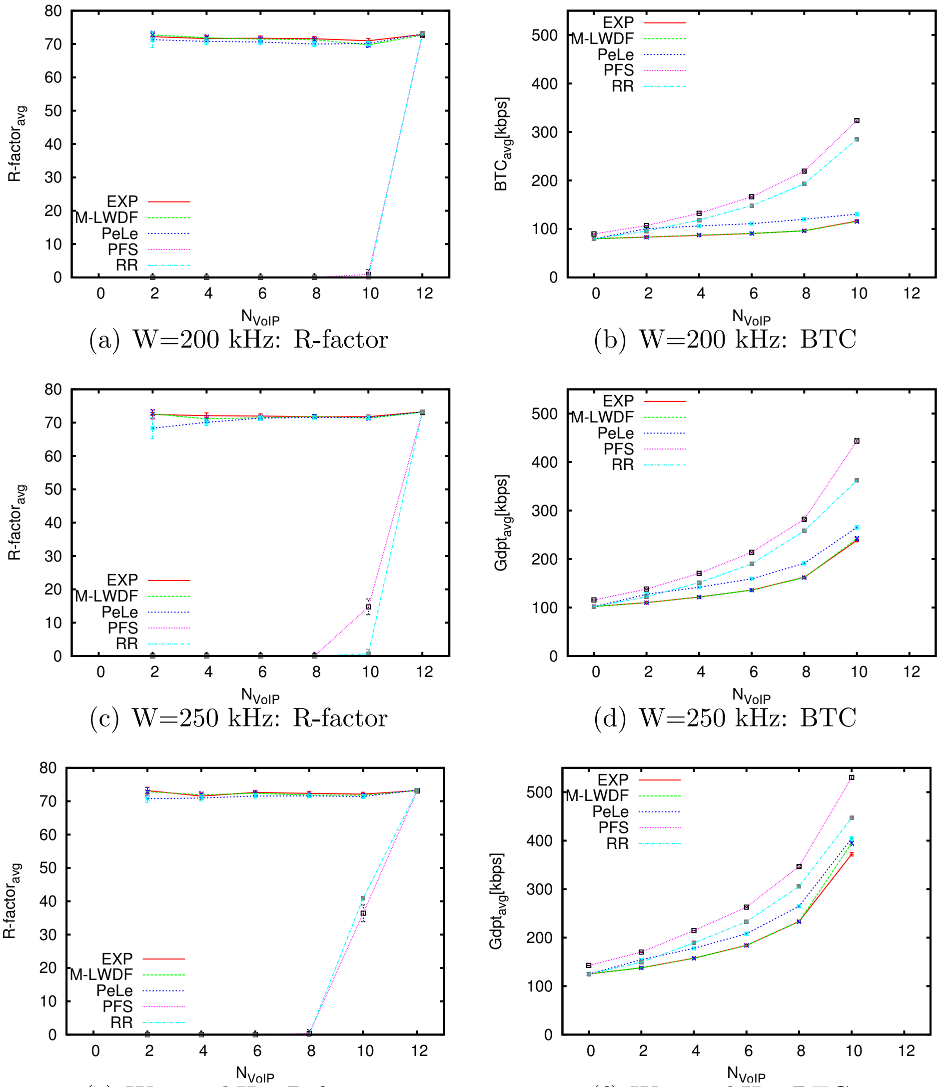
This chapter presents the performance evaluation of the PeLe scheduler that is proposed in this thesis to improvet the perceived quality of service of diversified application types in a multi-user scenario. On one hand, it shows the improvements in user perceived quality achievable by the proposed scheduler when compared to reference schedulers, considering also the cost of providing that higher quality. On the other hand, the functionality of the scheduler is explained and clarified. The influence of the different parameters in its performance is thoroughly studied, so that the solution can be improved and extended to other traffic types.

First, a traffic mix with only VoIP and Bulk traffic is used to explain the behaviour of the scheduler in different load situations and to study how the support of VoIP quality is achieved. Section 6.2 studies the effects of using utility curves of different shape for Bulk traffic. Then, Section 6.3 first shows how the utility curve for WWW traffic is designed and then evaluates the quality of the different applications with varied traffic mixes. Section 6.4 shows how flow weights can be used to change the relative priority of flows. Section 6.5 gives a different perspective by evaluating the performance of the PeLe and reference schedulers in terms of satisfied users, where satisfaction is defined according to the utility curves and application specific metrics. A utility curve for VoD traffic is studied in Section 6.6, and the performance of the PeLe scheduler with a traffic mix containing all traffic types is evaluated in Section 6.7. Finally, Section 6.8 studies the performance of different scheduling paradigms in terms of total achieved throughput and the amount of satisfied users and establishes the trade-offs of the different approaches.

6.1 Performance with VoIP and Bulk Traffic

The first set of experiments studies the support of VoIP traffic, using a mix of VoIP and Bulk clients. For three channel bandwidths $W \in \{200, 250, 300\}$ kHz and 12 WT in the cell, the traffic mix is varied by varying the amount of VoIP clients—N_{VoIP}. The Bulk flows have a requested rate $r_{\mathrm{Bulk}} = 100$ kbps. Figure 6.1 shows the quality metrics for both application types and Figure 6.2 shows the total cell PHY throughput.

Figure 6.1: Average quality of the Bulk transfer and VoIP clients and total cell throughput for $W \in \{200, 250, 300\}$ kHz and $N = 12$.



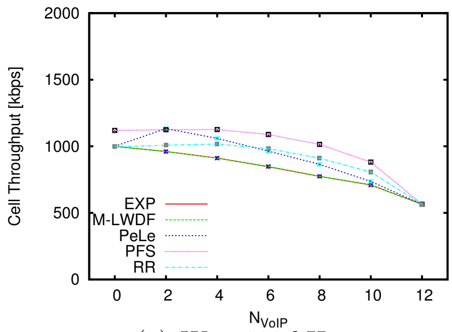
(a) W=200 kHz: R-factor (b) W=200 kHz: BTC

(c) W=250 kHz: R-factor (d) W=250 kHz: BTC

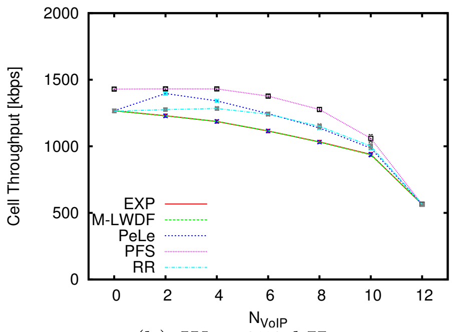
(e) W=300 kHz: R-factor (f) W=300 kHz: BTC

6.1.1 Scheduler Performance

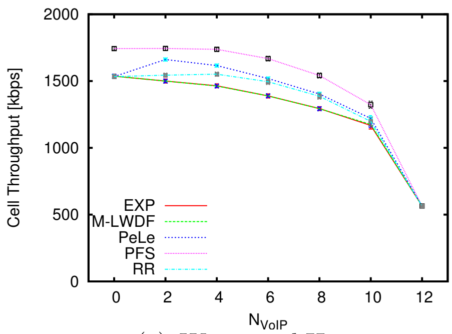
Figure 6.2: Total average PHY throughput in the cell—$\bar{\rho}$.



(a) $W = 200$ kHz



(b) $W = 250$ kHz



(c) $W = 300$ kHz

The evolution of the R-factor in Figures 6.1–a,c,e shows that the schedulers that do not take specific flow requirements into account, e. g. PFS and caRR, cannot support VoIP traffic. For both schedulers, the average R-factor of VoIP calls decreases as soon as the traffic mix includes Bulk transfers. Moreover, for more than 4 active Bulk clients it is 0, although there is enough bandwidth to support both types of traffic, as is achieved by the other schedulers. This happens because those schedulers neither allow the priorisation of VoIP traffic nor any other control over the delay of the packets. Because the scheduler handles only the traffic generated by the Bulk server, only SYN, SYN+ACK, FIN+ACK packets and eventually the last packet of a file are shorter than the MTU, and their transmission occupies the channel for a long time. When Bulk flows compete with the VoIP flows, several large packets belonging to Bulk flows may be served in sequence by the caRR scheduler, occupying the channel for a long time and leading to large delays. The PFS, which serves the flow with the best relative channel quality may not serve a VoIP packet for a long time, if relative link qualities of the other WT are better. These schedulers take the link quality into account but ignore the specific requirements of VoIP flows, thus are not able to deliver proper perceived quality. However, because the PFS and caRR serve less VoIP packets which have a high PHY and MAC overhead, they achieve the two highest average BTC, as shown in Figures 6.1.

The PeLe, EXP and M-LWDF deliver similar and high average R-factors to the VoIP clients, showing only little variation when the number of VoIP clients increases. However, the BTC achieved by the PeLe is higher than for the EXP or M-LWDF. This is a consequence of the PeLe tolerating higher delays of the VoIP packets, as long as they do not lead to excessive packet drops, the measure of excessive being defined by the utility curve. Figure 6.3 shows the tails of the end-to-end packet delays for the EXP, M-LWDF and PeLe schedulers for $W = 200$ kHz. The EXP and M-LWDF keep the delays very low because they strive to balance the queuing delays of the single flows according to the weights of each flow type. The Bulk flows have weights $a_i = 0.001$ which was found to be the highest value that guarantees good VoIP perceived quality in simulations of the performance of the EXP scheduler. The PeLe scheduler works differently: it takes advantage of the tolerance expressed by the scheduling metric of the VoIP flow. It expresses the fact that delays are only a problem if they lead to packet drops and that some packet drops which are due to exceeding the delay are also acceptable. This results in larger queuing delays of the VoIP packets, but has no significant influence in the perceived quality of VoIP flows, as can be seen in Figure 6.1–a,c,e. Since the PeLe scheduler is not making the effort to keep low delays, it has more freedom to schedule the large Bulk packets at better channel conditions, which is why the BTC is higher (Figures 6.1–b,d,f).

The cost of the higher perceived quality achieved by the PeLe scheduler is a decrease in the cell throughput, seen for all channel bandwidths in Figure 6.2. When the offered load consists only of VoIP flows, all schedulers peform similarly because the offered load is very low; differences between the schedulers can only be seen in other traffic scenarios. The PFS has the highest cell throughput in all cases, as expected from the considerations in Sections 4.2 and 5.4. The EXP and M-LWDF have the lowest throughput because giving very low delays to the VoIP packets limits the freedom to take advantage of the best link periods. The total cell throughput of the caRR and PeLe lie in-between. When only Bulk clients are present ($N_{\text{VoIP}} = 0$) the PeLe has a similar throughput to the caRR because all WT are receiving similar high service. When all flows receive the requested service (or more) and have $\xi_i = 1$, the quality variation $\Delta\xi$ due to the transmission or not of a packet is 0, leading to random decisions[1]. For $N_{\text{VoIP}} = 2$ the PeLe cell throughput rises and is higher than that of the caRR because the number of random scheduling decisions decreases and the PeLe can take advantage of good link condition

[1] This might be improved by scheduling the flow with best link condition instead of solving ties randomly.

more often. Then, for an increasing number of VoIP clients, the cell throughput of the PeLe scheduler decreases steadily because of the increasing number of VoIP packets, which have a large overhead: 58 Bytes of MAC and PHY headers at the lowest available transmission rate are added to the 60 Byte payload transmitted at an eventual high rate. I. e. for VoIP transmissions the header overload is very high, as can be seen by the increase of the time overhead shown in the following table for the PeLe and caRR ($W = 200$ kHz).

	time overhead [%]	
N_{VoIP}	PeLe	caRR
0	0.093	0.093
2	0.213	0.112
4	0.312	0.140
6	0.410	0.179
8	0.507	0.240
10	0.602	0.367
12	0.760	0.760

The total throughput achieved by the PeLe decreases with increasing number of VoIP clients because of the time overhead. The channel throughput for the PeLe becomes lower than for the caRR when the overhead caused by the support of VoIP becomes larger than the benefit achieved by taking better advantage of good link periods.

6.1.2 Details of VoIP Support

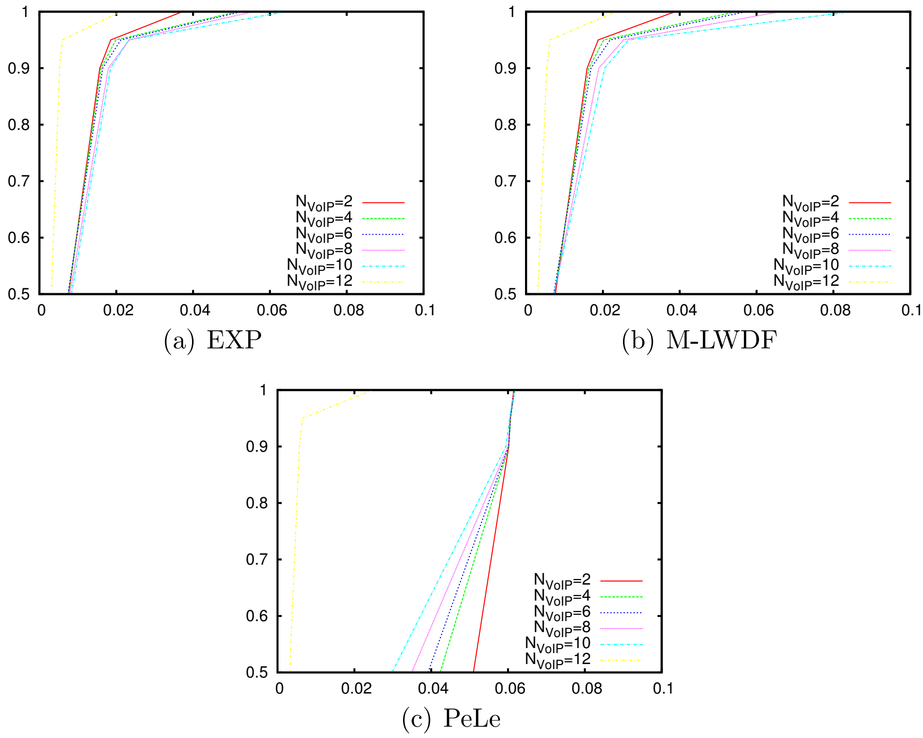
Figure 6.3 shows that the PeLe scheduler keeps the delays of VoIP packets close to the maximum tolerated without leading to a large number of drops. This is achieved by the VoIP specific queueing and by taking advantage the fact that VoIP traffic is CBR and packets are very short. As described in Section 3.5.3, the queue is dimensioned so that packets arriving at rate $r_{\bar{\text{VoIP}}}$ that wait more than the maximum acceptable delay $\delta_{max,\text{VoIP}}$ lead to a full queue and to an imminent packet drop. A drop is estimated to happen if the VoIP flow with a full queue is not served until the expected arrival of its next packet. Accordingly, the future quality of a VoIP flow when another flow j is transmitted $\xi^{\text{j}}_{\text{VoIP}}$ is 0 until a packet drop becomes "expected". This occurs when the next VoIP packet arrival is expected during the transmission of the other flow. If this is the case, the scheduling metric of the competing flow j decreases, because $\xi^{\text{j}}_{\text{VoIP}}$ is decreased by the estimated quality loss produced by the eventual VoIP packet drop.

Thus, VoIP packets close to the deadline have a high priority for two reasons. On the one hand, because short packets occupy the channel for short times, δ_{VoIP} is low. Additionaly, there is a lower error probability for short than for long packets, so a higher modulation can often be used, leading to small δ_{VoIP}. So, the transmission of VoIP packets often causes only little impact on other flows, i. e. ξ_j^{VoIP} is low. On the other hand, when a packet from flow j competes with a VoIP packet close to the deadline, ξ_{VoIP}^j is very high, reducing the priority of that other flow. The combination of these two factors gives VoIP packets close to their deadlines a very high rank, so they rarely exceed the deadlines.

6.1.3 The Role of θ_{VoIP}

The time window θ_{VoIP} plays an important role in the support given to VoIP flows, as it determines the granularity of sampling of the utility curve described by Table 3.1 and, consequently, the magnitude of $\Delta\xi_{\text{VoIP}}^j$ and $\Delta\xi_{\text{VoIP}}^{\text{VoIP}}$. The following table shows the PDR and the corresponding FM quality estimate ξ_{VoIP} for 1 and 2 packet losses for 3 different values of θ_{VoIP}, assuming that one VoIP packet is generated every 20 ms. Recall that each G.729 VoIP packet carries 2 VoIP samples, as explained in Section 5.3, so a packet loss means

Figure 6.3: Tails of the end-to-end VoIP packet delays for the EXP, M-LWDF and PeLe schedulers for $W = 200$ kHz and varying number of VoIP clients.



(a) EXP (b) M-LWDF

(c) PeLe

that 2 voice samples are lost.

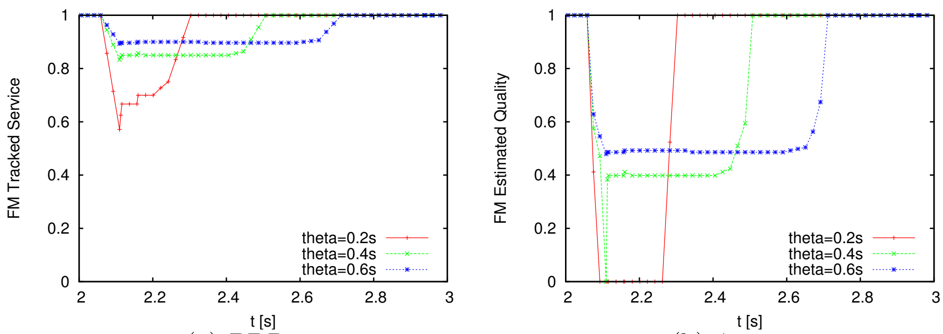
	$\theta = 0.2$ s		$\theta = 0.4$ s		$\theta = 0.6$ s	
Lost Packets	PDR	ξ_i	PDR	ξ_i	PDR	ξ_i
1	0.9	0.256	0.95	0.539	0.96(6)	0.658
2	0.8	0	0.9	0.256	0.93(3)	0.430

According to this table, if a VoIP flow has $\xi_{\text{VoIP}} = 1$ and risks dropping a packet if another flow j transmits, the other flow's ξ^j_{VoIP}=0.256, 0.539 or 0.658, for θ_{VoIP}=0.2, 0.4 or 0.6 s, respectively. Thus, the shorter θ_{VoIP}, the more the service tracked at the FM reflects the impact of a packet drop and the higher the priority of the VoIP flow expressed by the estimated quality metrics. Another way of looking at this is seeing that the utility curve is being enforced over a short time interval.

Figure 6.4 illustrates this in more detail. It shows the evolution of the service tracked and the estimated quality ξ_{VoIP} of a VoIP flow's FM for the 3 values of θ_{VoIP} above. These plots were obtained from a simulation in a controlled environment with only two WT sharing the channel. The first has a Bulk client and has a constant good channel. The second has a VoIP client with a constant good channel that suffers a fade of duration 0.1 s at t=2 s. The decrease of the estimated quality due to waiting for another flow to be served, $\Delta\xi^j_{\text{VoIP}}$, can be seen in the magnitude of the steps of ξ_{VoIP} (the difference between two consecutive points in the curve) at the beginning of the fade. The increase in estimated quality due to transmitting a VoIP packet, $\Delta\xi^{\text{VoIP}}_{\text{VoIP}}$, can be seen in the magnitude of the steps of ξ_{VoIP} at the end of the fade.

During the outage the VoIP queue fills up, leading to the loss of 4 packets that is reflected in the 4 decreasing steps of the tracked service in Figure 6.4–a. Each packet loss causes a larger decrease in the PDR for shorter θ_{VoIP},

Figure 6.4: Evolution of the ξ_{VoIP} and PDR estimated at the FM of a VoIP flow during a fade for $\theta_{\text{VoIP}} \in \{0.2, 0.4, 0.6\}$ s.



(a) PDR (b) ξ_{VoIP}

as expected from the values in the table above, and a corresponding slower decrease of ξ_{VoIP} (Figure 6.4–b). According to the definition of the scheduling metric in Section 3.4.2, the quality increase due to serving a flow is weighted by the quality decrease of the flows that have to wait. Thus, the use of a short θ_{VoIP} causes a higher cost of another flow's transmission if it causes an eventual VoIP packet drop.

The attention should now be focused on the speed of the increase of the quality estimate ξ_{VoIP}. Using large θ_{VoIP} leads to lower $\Delta\xi_{\text{VoIP}}^{\text{VoIP}}$, seen in the smaller steps of ξ_i for increasing θ_{VoIP}, and may lead to consecutive VoIP drops after a starvation. This can have severe consequences for the user perceived quality, since burst drops are more difficult for the decoder to conceal.

In Figure 6.4–a, when many packets get lost the tracked service decreases more for θ_{VoIP}=0.2 s than in the other cases because of the coarser granularity of the service tracker. So, several packets must be sent before the quality estimate starts to increase from 0, as can be seen in Figure 6.4–b. This means that for several packets, the $\xi_{\text{VoIP}}^{\text{VoIP}}$ is 0, giving the flow low priority. However, this is a problem only if a VoIP flow is starved several tens of milliseconds long, what is very unlikely to happen due to fading in a realistic scenario. The extremely long outage was artificially induced to better illustrate the behaviour.

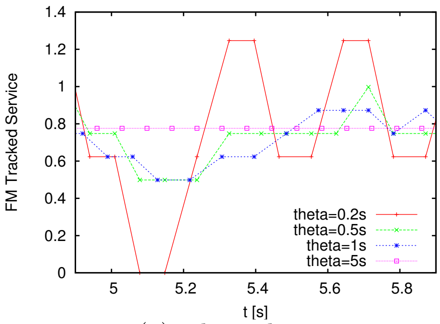
Henceforth θ_{VoIP}=0.2 s is used. Although this gives very high priority to VoIP flows in the imminence of and after packet drops, it also enables a greater flexibility in scheduling other flows, as was seen in the previous section.

6.1.4 The Role of θ_{Bulk}

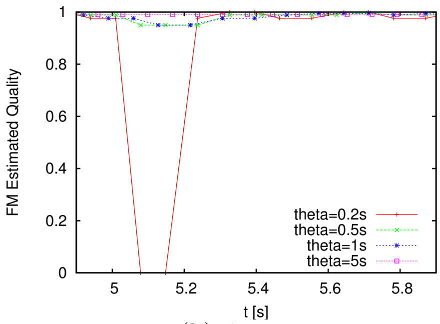
This section studies the effects of varying θ_{Bulk}. Figure 6.5 shows the evolution of the service tracked by the FM and the ξ_{Bulk} during a fade of 100 ms at t=5 s, in the same controlled scenario used in the previous section. As the fade occurs, the tracked service decreases fast for $\theta_{\text{Bulk}} = 0.2$ s, and decreases slower for increasing θ_{Bulk}. This is similar to what was seen for VoIP. For $\theta_{\text{Bulk}} = 5$ s the 100 ms wide fade remains unnoticed by the tracked service. Because of the elastic utility curve of Bulk flows, the estimated quality, ξ_{Bulk}, hardly reflects the fade for $\theta_{\text{Bulk}} \geq 0.2$ s. So, starving the flow for 100 ms would not affect the priority of the Bulk flow except for $\theta_{\text{Bulk}} \geq 0.2$ s, and would even remain unnoticed for $\theta_{\text{Bulk}} = 5$ s. Because TCP would not notice the outage, this is not a problem, but in fact the desired behaviour. In general, θ should take a value that expresses the timescale of the application's sensitivity to the service delivered. That is the timescale in which the priority of the flow should be affected and cause the scheduler to react to the flow's needs. In further performance evaluation, θ_{Bulk} is set to 2 s.

Figure 6.6 shows the R-factor, Bulk goodput and total achieved PHY

Figure 6.5: Evolution of the ξ_{Bulk} and resources tracked at the FM of a Bulk flow during a fade for $\theta_{\text{Bulk}} = \{0.2, 0.5, 1, 5\}$ s.



(a) Throughput



(b) ξ_{Bulk}

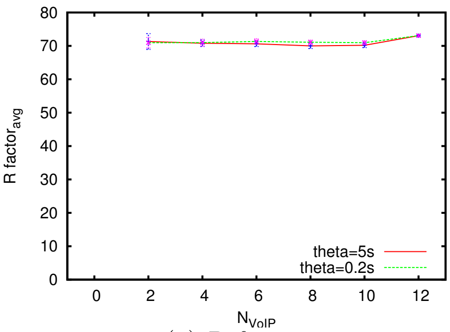
throughput of the PeLe scheduler in the same scenario as described at the beginning of this section for W = 200 kHz, θ_{Bulk}=5 s and θ_{Bulk}=0.2 s (the differences can be better shown using the two extreme values). The average R-factor is not significantly affected by the change in θ_{Bulk} because of the very high priority of VoIP flows, as explained above. When a shorter θ_{Bulk} is used, the average and range of the BTC of Bulk clients is lower than for a large θ_{Bulk} (Figure 6.6–b). This happens because the scheduler tries to meet the service requirements over shorter time windows and has to compensate for starvation more often. As a consequence, Bulk packets are more frequently sent over lower quality links, leading to lower total PHY throughput, as can be seen in Figure 6.6–c. This reasoning is supported by the statistics in this table (taken from the simulations for $N_{\text{VoIP}} = 6$).

	θ_{Bulk}	
	0.2	5
time overhead [%]	0.40	0.41
$\bar{k}$	3.46	3.60
# VoIP packets	521664	519990
# Bulk packets	88038	95860

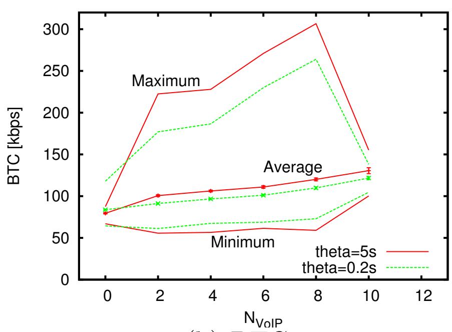
For θ_{Bulk}=0.2 s more VoIP packets are sent than for θ_{Bulk}=5 s, and the time overhead and $\bar{k}$ is expected to be higher. However, this is not the case; the time overhead and $\bar{k}$ are lower because the Bulk payloads are transmitted more often at lower data rates, leading to the lower throughput achieved for θ_{Bulk}=0.2 s.

The only exception to the rationale above is when only Bulk traffic is available ($N_{\text{VoIP}} = 0$). Because there is enough bandwidth to satisfy the requests of all flows, $\xi_{\text{Bulk}} = 1$ most of the time. When the utility curve is enforced over

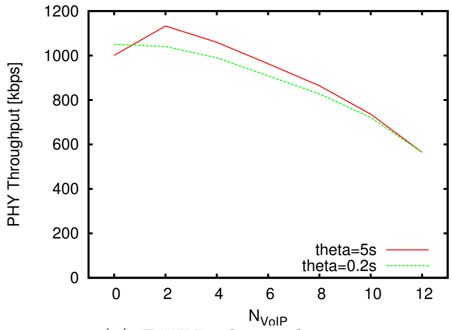
Figure 6.6: Average perceived quality of VoIP calls and the Bulk transfer and total cell throughput for $W \in \{200, 250, 300\}$ kHz and $N = 12$.



(a) R-factor



(b) BTC

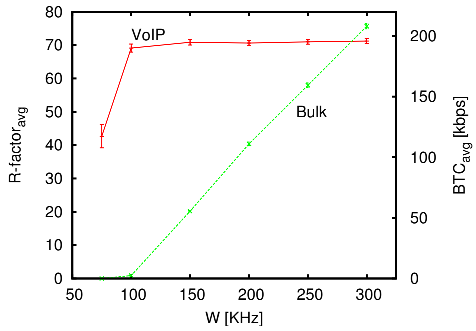


(c) PHY Throughput $\bar{\rho}$

long time intervals, starvation periods remain unnoticed by the estimated quality metrics $\xi_{\text{Bulk}}^{\text{Bulk}}$ and ξ_{Bulk}^{i}. That leads to more random scheduling decisions for $\theta_{\text{Bulk}} = 5$ s (26786 vs. 14335 on average) and, thus, to a lower throughput. The following table supports this showing the amount of random scheduling decisions and the average amount of bit per symbol $\bar{k}$ for both values of θ_{Bulk} for each of the 5 simulation runs.

	Random decisions		$\bar{M}$	
θ_{Bulk}	0.2	5	0.2	5
1	15910	27834	3.21	3.11
2	14722	27728	3.35	3.17
3	11828	24754	2.92	2.79
4	14416	27241	3.20	3.07
5	14797	26373	3.08	2.94

Figure 6.7: Average perceived quality of the Bulk transfer and VoIP clients when the cell is driven into overload by decreasing the channel bandwidth W ($N = 12$ and $N_{\mathrm{VoIP}} = 6$).



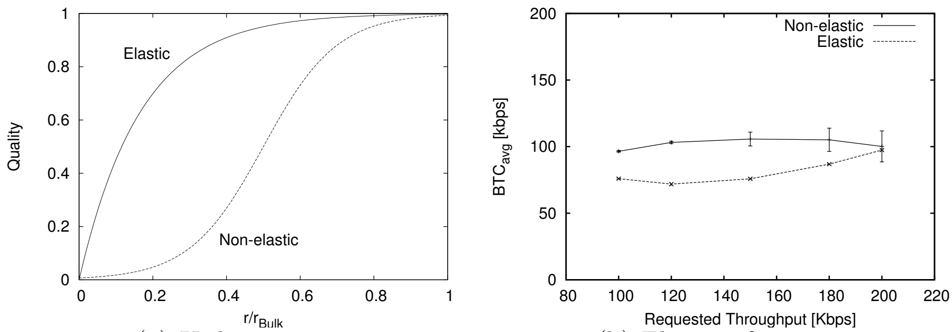
6.1.5 Behaviour in High- and Overload

Figure 6.7 the perceived quality of the VoIP and Bulk flows when the channel bandwidth W is decreased from 200 kHz for $N_{\mathrm{VoIP}} = 6$. As the channel bandwidth decreases, the average BTC of Bulk data transfers decreases and for $W = 100$ kHz the Bulk clients receive almost no data, while the average R-factor is still high and close to the toll value of 70. This is the consequence of the extremely high priority of the VoIP flows, expressed by the strict utility curve and short θ_{VoIP}, together with the high tolerance of the Bulk flows to receiving lower service than requested, expressed by the elastic utility curve and the large θ_{Bulk}.

6.2 Elastic versus Inelastic Utility Curve for TCP Traffic

This section shows how the use of an elastic utility curve changes the behaviour of the PeLe scheduler towards TCP traffic. Figure 6.8–a shows both utility curves. The experiment set-up consists of 12 WT receiving Bulk file transfers, where half of the Bulk flows are supported by an elastic utility curve as described by Eq. 3.5 and the other half by an inelastic utility curve described by Eq. 3.6. The requested service is the same and θ_{Bulk}=2 s for all flows. The cell is slightly overloaded at a requested rate $\bar{r}_{\text{Bulk}}$ of 100 kbps per flow. Increasing the requested rate drives the cell further into overload, enabling a clearer observation of the scheduler behaviour when the available bandwidth is scarce with respect to the requests.

Figure 6.8: Average BTC of TCP Bulk transfer for increasing rate demand ($N = 12$ $W = 200$ kHz).

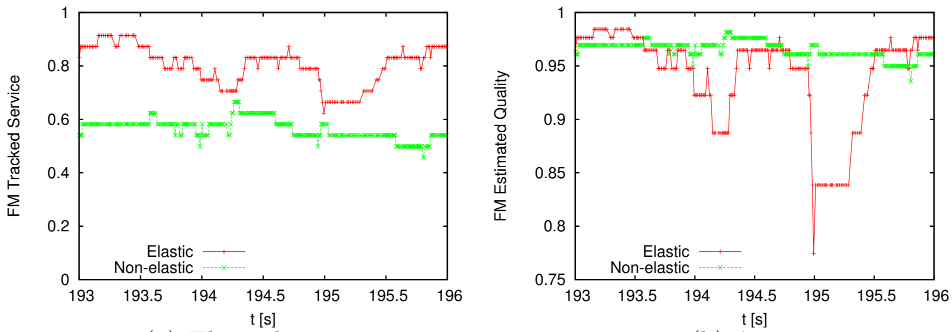


(a) Utility curves (b) Flow performance

Figure 6.8–b shows that the average BTC is higher for the flows with a inelastic utility curve than for flows with an elastic utility curve. For $\bar{r}_{\text{Bulk}}$=100 kbps, the available channel resources are slightly scarce. Inelastic flows receive a higher average BTC because their packets have higher priority due to the higher gradient of the inelastic utility curve. The estimated quality metrics depend on the gradient of the utility curve at the current estimated quality—$u'(\xi_i)$, and it is larger for the inelastic curve when ξ_i is slightly lower that 1. So, although all flows receive less service than requested, packets from inelastic flows have higher ranking metrics and are scheduled more often than packet from elastic flows, whose utility curves have lower gradient at similar estimated quality.

This can be observed in Figure 6.9, which shows the evolution of the service tracked at the FM and the corresponding estimated quality for one elastic and one inelastic flow, obtained from a run with requested rate $\bar{r}_{\text{Bulk}} = 150$ kbps.

Figure 6.9: Evolution of the ξ_i and throughput estimated at the FM of an elastic and an inelastic flow for $\bar{r}_{=}$150 kbps.



(a) Throughput (b) ξ_i

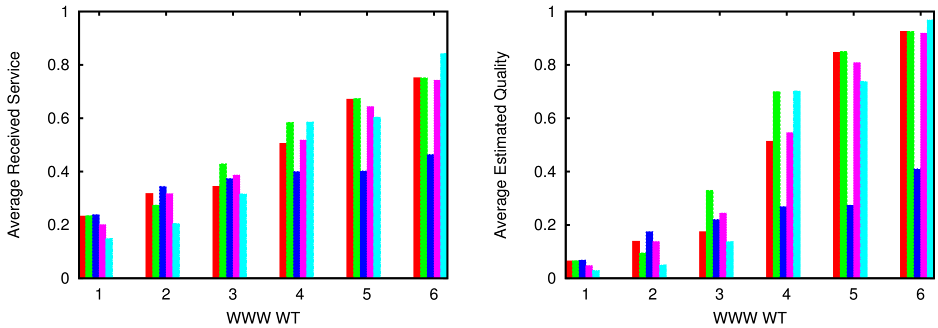
Notice that although the average service delivered to the elastic flow is approximately 20% less than the average service delivered to the inelastic flow (Figure 6.9–a), the quality according to the utility curves is similar (Figure 6.9–b). This means that, from the point of view of the scheduler, both flows are being equally well served: $\xi_{\text{elastic}} \approx \xi_{\text{inelastic}}$.

A decrease in the tracked service can be seen at t=194 or t=195 s in Figure 6.9–a and is similar for both flows. However, the corresponding variations in estimated quality in Figure 6.9–b differ: they are much larger for the inelastic than for the elastic flow because of the higher gradient of the inelastic utility curve at the estimated quality ξ_i. Thus, when the service delivered to both flows decreases, packets from the non-elastic flows receive higher priority until the estimated quality rises again to a level where the gradients of both utility curves are similar, as happens for $t = 195.5$ s in Figure 6.9–b.

As the requested data rate$\bar{r}_{\text{Bulk}}$ increases, the average BCT of elastic flows increases. At the same time the variance of the BTC for non-elastic flows increases as seen in the increasing confidence intervals in Figure 6.8. The higher variance of the average BTC of non-elastic flows is due to a higher variance in each run, i. e. in each simulation run there are big differences in the service given to different non-elastic flows. Why does this happen? The justification is similar to why the elastic flows receive less service than non-elastic flows, only it is due to the competition among non-elastic flows. The elastic flows are not considered in the following paragraph because their have much lower priority than any of the non-elastic flows.

As the requested data rate $\bar{r}_{\text{Bulk}}$ increases, the average received service $r/\bar{r}_{\text{Bulk}}$ decreases, and so does the FM estimated quality ξ. The gradient of the non-elastic utility curve $u'(\xi)$ increases at first when the tracked resources decrease, but when ξ becomes less that 0.5 the gradient starts decreasing

Figure 6.10: Received service on average as seen by the FM and the corresponding estimated quality according to the utility curve for each non-elastic flow after each simulation run for $\bar{r}_{\text{Bulk}}$ =200 kbps. Flows are sorted according to increasing average received service.



(a) $r/\bar{r}_{\text{Bulk}}$ (b) $u(r/\bar{r}_{\text{Bulk}})$

again. Non-elastic flows with that low average tracked service have a lower gradient and a lower priority, causing their ξ and gradient to decrease further. Thus, some non-elastic flows can keep a higher service at the cost of others. Figure 6.10–a shows the service delivered on average to each non-elastic flow—the average received PHY throughput, r, divided by the requested throughput, $\bar{r}_{\text{Bulk}}$; Figure 6.10–b shows the corresponding quality according to the non-elastic utility curve. The average received service of WT 1 to 3 correspond to very low ξ and much lower $u'(\xi)$ than WT 4 to 6 in the curves in Figure 6.8–a.

This behaviour agrees with the scheduling goal, as can be seen when the total quality seen at the FMs achieved by the PeLe scheduler is compared with the case of evenly distributed service. Dividing the total service available to non-elastic flows evenly among them would deliver the following average service and estimated quality in each run

Service	94.052	98.204	74.064	93.659	90.063
Quality	0.426	0.478	0.215	0.421	0.378

The following table shows the sum of the FM estimated quality of the non-elastic flows according to the utility curve for the PeLe scheduler if all flows would receive the same service.

Even rate share	2.557	2.865	1.288	2.528	2.270
PeLe	2.662	2.963	1.415	2.703	2.624

As desired, for each run, the average estimated quality seen at the FM is higher when the PeLe scheduler is used than if all non-elastic flows were evenly sharing the throughput allocated to non-elastic flows.

In Figure 6.8–b it can also be observed that the average BTC received by elastic flows increases as the requested rate r_{Bulk} increases. This is also a consequence of the strong competition for channel access among non-elastic flows. When several non-elastic flows are in the range of large gradient of their utility curve, the transmission of a packet at low rate can produce a high degradation of the estimated quality of the others. In these situations, a packet from an elastic flow that can be transmitted at high rate produces lower overall quality degradation, hence the increase in the BTC of elastic flows. This is a consequence of the high overload artificially induced in the system and does not happen under normal load situations.

The use of an elastic and a non-elastic utility curve enables the PeLe scheduler to deliver differentiated service to TCP traffic. The elastic curve allows the service given to the flow to be much lower than requested without increasing the priority of the flow. Conversely, the non-elastic curve tries to keep the delivered service above a certain level but reduces drastically the priority of the flow when the service becomes too low. The non-elastic curve suggests that the application can make no use of less than a certain amount of service, and in that case it is often a waste of resources to send data to that application anyway.

6.3 Support of Intermittent, Interactive WWW Traffic

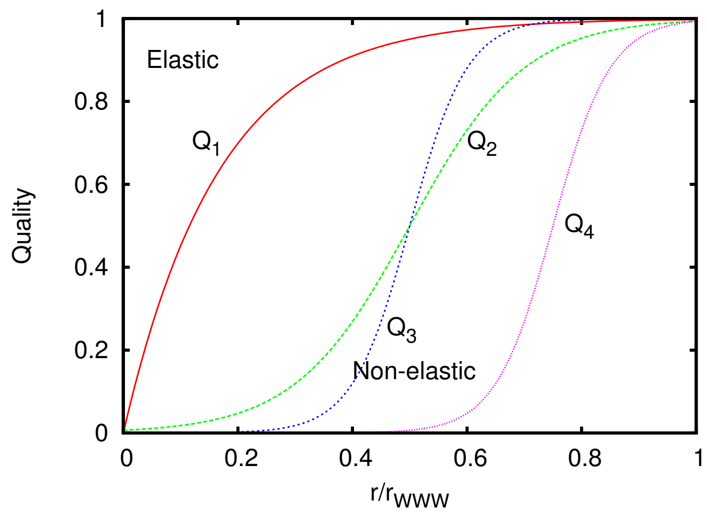
According to the behaviour of a WWW client described in Section 5.3, WWW traffic consists of sequential downloads of files of different sizes with varying intervals between downloads, creating an intermittent traffic pattern. This section studies a utility curve to support WWW traffic by observing how utility curves of different shapes support pages of different sizes. At the end of the section, the performance of the PeLe scheduler is evaluated with traffic mixes that include WWW traffic.

6.3.1 Study of an utility curve

The traffic scenario consists of 4 Bulk clients, 4 VoIP clients and 4 WWW clients in a cell with W=150 kHz. The WWW clients request pages of fixed size per run, with page size $L \in \{2.5, 12, 60, 120\}$ kBytes and the think time fixed to 2 s. The requested rate of the WWW flows is $\bar{r}_{\text{WWW}}$ =100 kbps. Bulk flows request $\bar{r}_{\text{Bulk}}$ =50 kbps each. The four utility curves plotted in Figure 6.11 are studied:

- $Q_1(x) = 1 - \exp^{-6x}$ (the elastic utility curve in the previous section)

Figure 6.11: Possible utility curves for WWW traffic.

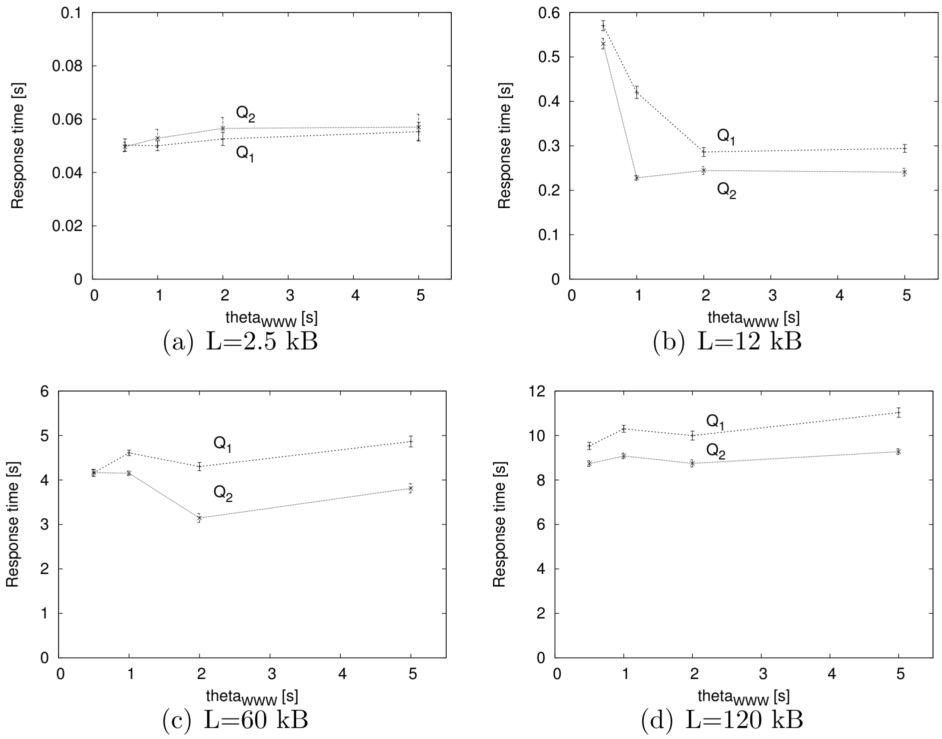


- $Q_2(x) = \frac{1}{1+exp^{-10(x-0.5)}}$ (the inelastic utility curve in the previous section)
- $Q_3(x) = \frac{1}{1+exp^{-20(x-0.5)}}$
- $Q_4(x) = \frac{1}{1+exp^{-20(x-0.75)}}$

The average response times per page for $\theta_{\mathrm{WWW}} \in \{0.5, 1, 2, 5\}$ s are measured for simulation runs of 600 s.

Figure 6.12 shows the mean response times and confidence intervals, calculated over all pages and clients, for each page size for Q_1 and Q_2. It is important to notice that the initial phase of a flow in the FM, which is a transient for the large Bulk file transfers in the previous section, plays a larger role for short WWW pages. The initial phase of the flow, when the estimated quality ξ_i is growing from 0 to 1 along the utility curve will be called the flow's start phase. The service delivered to the flow is calculated as the amount of bits sent divided by the observation interval θ_{WWW}. Thus, sometimes a whole page may not be enough for the flow to reach $r = 1$, and the flow does not leave the start phase. In that case, the longer the θ_{WWW}, the smaller fraction of requested service the whole page represents. The same holds for a single packet: the longer the θ_{WWW}, the smaller fraction of requested service it represents. The following table illustrates that fact, showing the delivered service corresponding to a 1500 Byte TCP packet and to a whole page for the page sizes used.

Figure 6.12: Response times for the WWW clients with fixed packet size L for the elastic and inelastic utility curves Q_1 and Q_2 and varying values of θ_{WWW}.



(a) L=2.5 kB

(b) L=12 kB

(c) L=60 kB

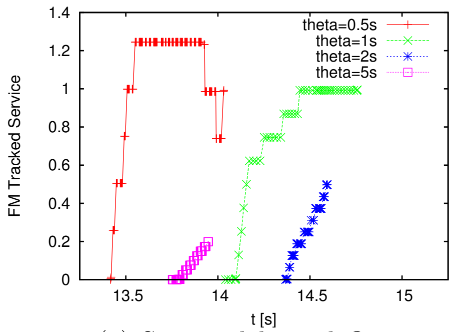
(d) L=120 kB

	L [kB]				
θ_{WWW}	1.500	2.5	12	60	120
0.5	0.252	0.4	1.92	9.6	19.2
1	0.126	0.2	0.96	4.8	9.6
2	0.063	0.1	0.48	2.4	4.8
5	0.025	0.04	0.192	1.2	1.92

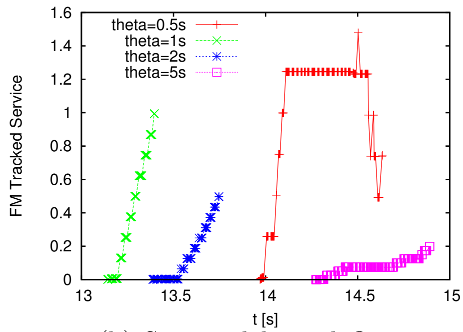
This determines how many packets it takes for the estimated quality ξ_{WWW} to move from left to right along the utility curve at the start of the flow. For smaller θ_{WWW} the estimated quality grows faster, i. e. the fewer packets are needed until ξ_{WWW} reaches 1. It has been shown that the part of the start phase where the flow has the highest priority is where the utility curve has the highest gradient. So the flow has a higher priority for a longer time if it traverses the high gradient part of the curve in small steps. This is relevant for small pages, for which the start phase of the flow represents most part or all of the page.

Figure 6.13 shows examples of the time evolution of the delivered service and estimated quality for different θ_{WWW} for a 12 kB page when Q_1 or Q_2 are

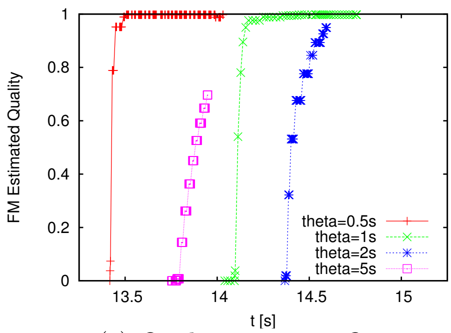
Figure 6.13: Evolution of the FM tracked service and estimated quality of the WWW flow with 12 kB pages for utility curves Q_1 and Q_2 and varying θ_{WWW}.



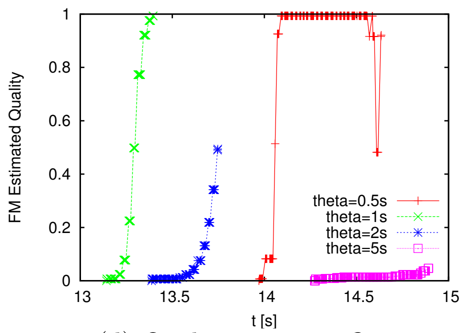
(a) Service delivered Q_1



(b) Service delivered Q_2



(c) Quality estimate Q_1



(d) Quality estimate Q_2

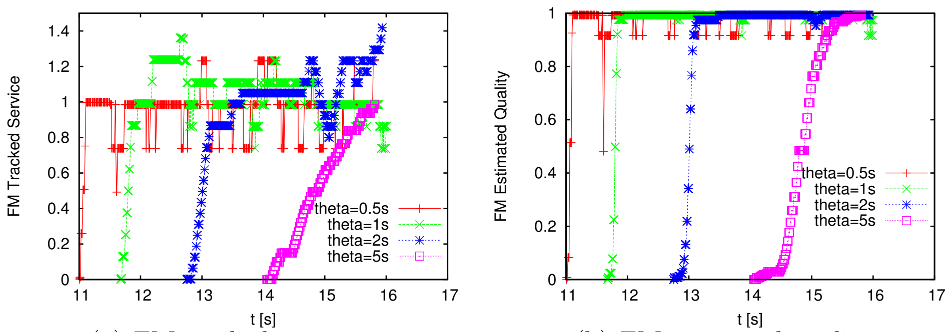
used. In Figures 6.13–a and b it can be seen that for $\theta_{\mathrm{WWW}} > 0.5$ s the whole page represents only a fraction of the requested service, as expected from the table above. For $\theta_{\mathrm{WWW}} = 5$ s, each packet represents only a small service increase (0.025), leading to the slow increase of the delivered service seen in Figures 6.13–a and b.

One big difference between the elastic and inelastic utility curves can be seen at the beginning of the flow start phase. For Q_1 the increase in quality corresponding to the little increase in tracked service at flow start is very high, whereas for Q_2 it is low: for similar tracked service for $\theta = 5$ s in Figures 6.13–a and b the quality ξ_{WWW} in Figure 6.13–c grows more than in Figure 6.13–d. Since the quality increase per packet determines the scheduling metric, the packets enjoy higher priority in the first case than in the latter, which explains why they are scheduled faster for Q_1 than for Q_2. This leads to the better performance of Q_1 for very short pages as seen in Figure 6.12–a.

Another difference between Q_1 and Q_2 is the behaviour after the start phase. For Q_1, as the estimated quality reaches values above 0.3, the gradient of the utility curve decreases and so does $\Delta\xi_{\mathrm{WWW}}^{\mathrm{WWW}}$, expressing lower flow priority. This is more pronounced for smaller values of θ_{WWW}, since the start phase ends after the transmission of fewer packets —curves for $\theta_{\text{WWW}} = 0.5$ and 1 s. The estimated quality increase due to transmitting a packet for service larger than 0.4 is very low, and the estimated quality does not degrade even if the flow is starved for longer periods of time. This can be seen in Figure 6.13–c. For $\theta_{\text{WWW}} = 1$ s it takes more than 0.5 s to transmit the last 4 packets and for $\theta_{\text{WWW}} = 0.5$ s the service even decreases to 0.5 before the last packet is scheduled (Figure 6.13–a), but the quality of the flow does not reflect it (Figure 6.13–c). Although this the behaviour wanted from the scheduler as expressed by an elastic curve, it is not adequate for WWW flows because it can lead to unnecessarily large page response times.

When Q_2 is used the behaviour of the tracked service and estimated quality ξ_{WWW} is different. With $\theta_{\text{WWW}} = 0.5$ s, the start phase of the flow is short (4 packets), but a higher service is enforced by the inelastic utility curve in the steady state phase (0.9 instead of ≈0.3). For $\theta_{\text{WWW}} = 1$ s, the estimated quality is in the high gradient part of the utility curve for most of the page duration and almost all packets have high priority, explaining the low response times that can be seen in Figure 6.12–b. For $\theta_{\text{WWW}} = 2$ s the estimated quality grows in small steps along the steep part of the utility curve, but the flow ends before taking advantage of it all. For $\theta_{\text{WWW}} = 5$ s, it takes long for ξ_{WWW} to cross the initial, convex part of the curve before the high gradient segment and the response time is large.

Figure 6.14: Evolution of the FM quality estimate and tracked service of the WWW flow with 60 kB pages for utility curve Q_2 and varying θ_{WWW}.



(a) FM tracked resources (b) FM estimated quality

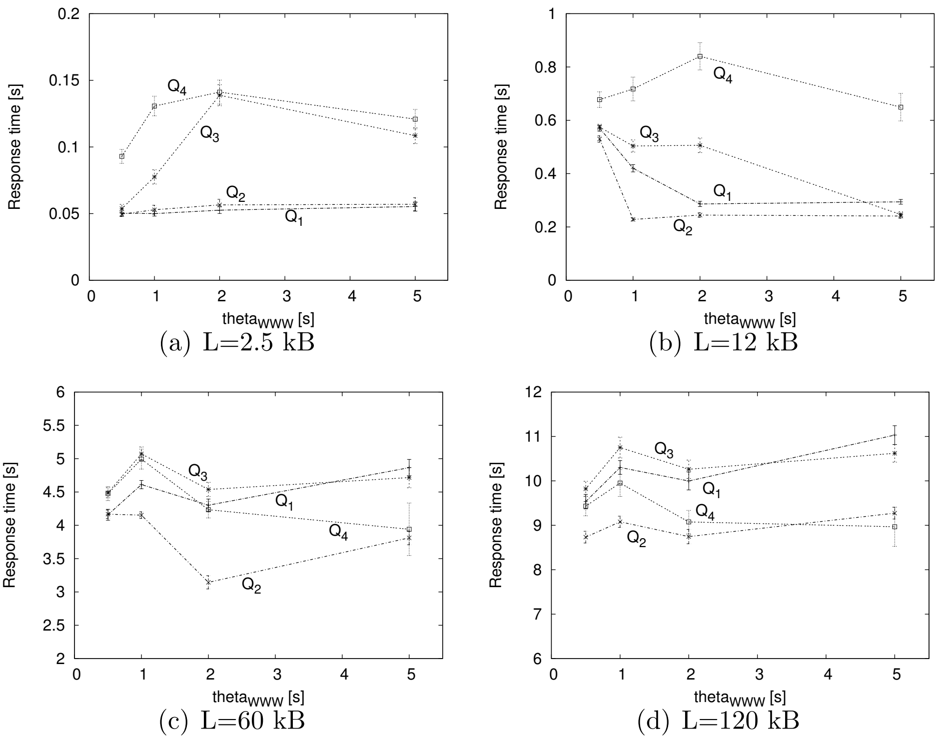
Figure 6.14 shows the evolution of the tracked service and estimated quality at the FM for a 60 kB page. For $\theta = 5$ s, the estimated quality is in the high gradient part of the utility curve for most of the flow duration, so it has high priority and finishes fast. As θ_{WWW} decreases, the estimated quality moves faster across the steep part of the curve. Afterwards, the behaviour is similar to what was seen in Section 6.2 for an inelastic utility curve.

According to these observations, an inelastic curve is more appropriate to support WWW traffic. So now it is time to study different parameters for the curve—steepness a and location b—by comparing Q_2, Q_3 and Q_4. Figure 6.15 shows the average page response time for all utility curves and the packet sizes 2.5, 12, 60 and 120 kB.

For small pages, Q_4 leads to much larger page response times than Q_2 (or even Q_1) because these pages consist of only a few packets and the quality estimate does not reach the part of the utility curve with high gradient. The response time is smaller for low θ_{WWW} because then even a few bytes of the page are enough to reach the convex part of the utility curve and enjoy high priority.

Even for large pages, using Q_4 leads to higher average page response times than Q_2 (Figure 6.15–c and d). Again the problem lies in the transmission of the initial packets: until the tracked service exceeds approximately 0.6 (corresponding to a little less than 7.5 kB for $\theta_{\text{WWW}} = 1$ s at the requested bandwidth of 100 kbps) the gradient of the utility curve is extremely low and the packets have very low priority. However, once the page start phase is over, Q_4 is much

Figure 6.15: Response times for the WWW clients with fixed packet size L for Q_1, Q_2 and Q_3 and varying values of θ_{WWW}.



(a) L=2.5 kB

(b) L=12 kB

(c) L=60 kB

(d) L=120 kB

stricter in keeping the delivered service close to the requested service, because the estimated quality loss due to not scheduling a packet is very high when the tracked service decreases below 0.85. Although this could lead to a better performance of Q_4 for large pages, even in those cases the long time spent to send the first packets of the page is dominant and Q_4 produces on average larger response times than the previously used inelastic curve Q_2.

The average response times achieved using Q_3 lie between those for Q_2 and Q_4 for short pages (Figure 6.15–a and b). This is because it takes longer to reach the part of the utility curve with a high gradient than when using Q_2. The service must be more than 0.2 which is equivalent to 2.5 kB for $\theta_{\mathrm{WWW}} = 1$ s (2 packets) and 1.25 kB for $\theta_{\mathrm{WWW}} = 0.5$ s (1 packet). But it takes less Bytes than when Q_4 is used. For large pages (Figure 6.15–c and d), the performance is worse than for both Q_2 and Q_4 because the utility curve is steep, but only if the service is below 0.6. So, when pages are long enough for the flow to reach the "steady state", Q_3 tolerates more degradation of the service than Q_2 of Q_4 before giving high priority to the flow again.

A steep inelastic utility curve located closer to the left hand side of the plot in Figure 6.11 would benefit short pages because even the first packets could profit from increased priority, similarly to Q_1. But the start phase of WWW pages would also finish after very little data, Furthermore, for the rest of the flow duration such a curve would tolerate even more service degradation than Q_3.

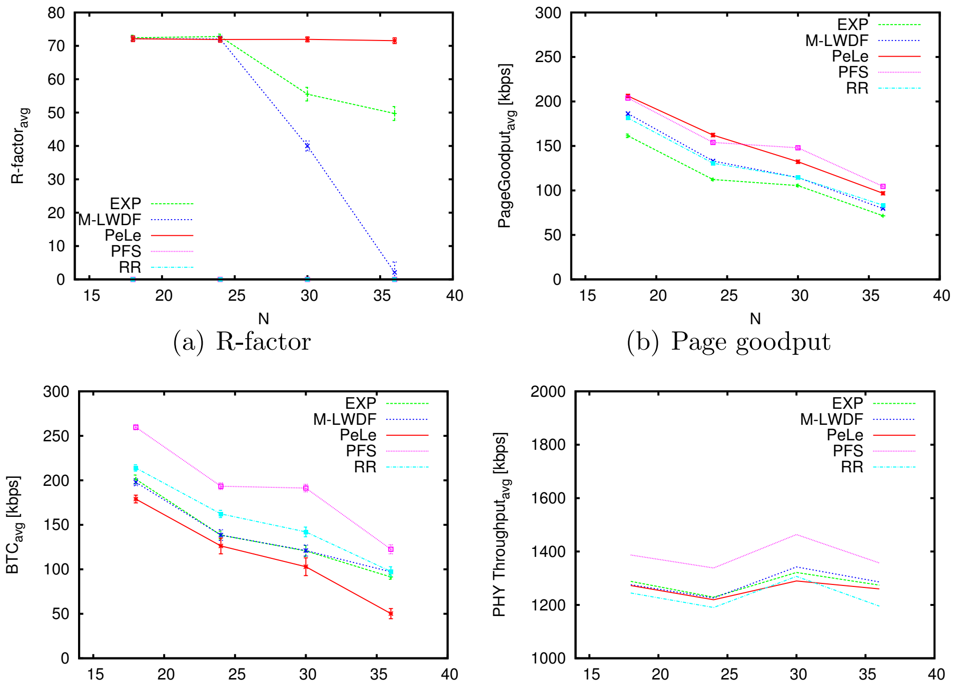
These considerations lead to the conclusion that Q_2 is the best utility curve among those studied here to support WWW traffic. It shall be used in subsequent work, with $\theta_{\mathrm{WWW}} = 0.544$ s, a choice that shall be justified in the next section.

6.3.2 Performance with VoIP, Bulk and WWW Traffic

This section shows the performance of the PeLe scheduler for traffic mixes including WWW traffic. The scenario consists of a wireless cell with W=250 kHz. Initially there are 18 WT in the cell, of which 4 have VoIP clients, 4 have Bulk clients, and 10 have WWW clients. The load in the cell is increased by increasing the amount of WT with WWW clients. The WWW clients request a maximum bandwidth of 200 kbps each one as do Bulk clients. Since the WWW traffic is intermittent with long think and idle times, most of the time only a small amount of WWW clients are actively downloading a page. There is very low probability that all are active at the same time, which is why the cell is not overloaded with that many clients requesting such high bandwidths.

Figure 6.16 shows the average perceived quality of the three applications in the traffic mix for increasing channel load. For 18 and 24 WT the EXP and M-LWDF schedulers deliver satisfying quality to the VoIP flows. However,

Figure 6.16: Average perceived quality of the Bulk transfers, WWW traffic and VoIP calls, and average PHY throughput for increasing WWW traffic.



(a) R-factor

(b) Page goodput

(c) BTC

(d) PHY Throughput $\bar{\rho}$

when the amount of WT in the cell increases, the PeLe scheduler is the single scheduler that can keep a high quality of the VoIP calls. Further, PeLe delivers higher average page goodput to the WWW clients than the EXP or M-LWDF of those schedulers. The price to pay for this is a lower BTC than any of the reference schedulers. The following table shows the difference between the BTC of Bulk flows achieved by the PeLe scheduler and the reference schedulers expressed as a percentage of the BTC achieved by the reference schedulers.

N	EXP	M-LWDF	PFS	caRR
18	11.04	9.55	31.07	16.34
24	8.78	8.83	34.71	22.18
30	14.75	14.99	46.22	27.58
36	45.08	48.35	59.03	48.55

As the load in the cell increases the difference in BTC increases because the PeLe uses more resources to support the WWW traffic. Since the Bulk utility curve expresses high tolerance to lower service than requested, when resources are scarce the Bulk flows are more penalised than the WWW flows. Despite the low BTC, according to the Bulk utility curve the perceived quality of the

Bulk users becomes lower than 0.9 only for N=36. Although the average BTC delivered by the PeLe to the Bulks flows is much lower than that achieved by the caRR or PFS schedulers, none of these can provide any quality to the VoIP calls.

Although the perceived quality metrics of the different applications differ enormously, the PHY throughput achieved by the schedulers does not vary significantly. Moreover, the PHY throughput achieved by the PeLe is at most 12% below that of the PFS and differs at most by 2% from the other reference schedulers.

The results in Figure 6.17 also show that, in contrast to the EXP or M-LWDF, the PeLe scheduler keeps its qualitative performance in changing load situations and across varied traffic mixes without any changes needed to the parameterisation. The flow weights that enable the EXP and the M-LWDF to support VoIP in the scenario in Section 6.1 are also valid for 18 and 24 WT in Figure 6.16, but not for other traffic mixes. Finding the weights that enable the support of VoIP with those schedulers would require new extensive simulations with the new traffic mixes.

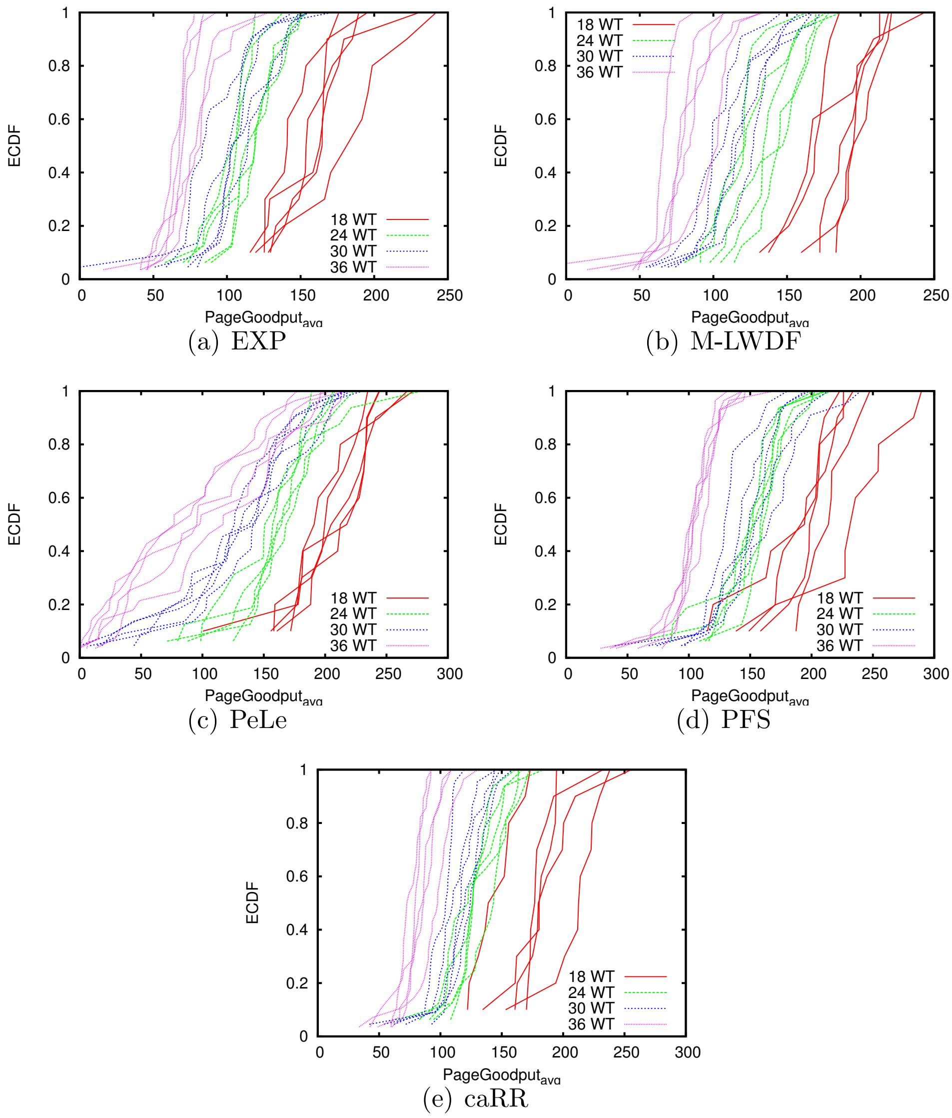
6.3.3 Fairness Among WWW Flows

Figure 6.17 shows the empirical CDF of the average page goodput of all WWW clients per run (there are 5 curves for each traffic scenario). The lower the variations in service delivered to different WWW clients, the more the CDF curve looks like a step. The caRR is the fairest scheduler in all load cases; the performance of the EXPand M-LWDF follows, with fairness increasing as the load increases, as can be seen in the increasing steepness of the ECDF curves for increasing N. The PeLe scheduler is less fair towards the WWW clients than the reference schedulers for all loads; moreover, the fairness in page goodput among the clients decreases as the load increases. This is a consequence of the inelastic utility curve used for the support of WWW traffic, which emphasises the fact that below a certain service there is no point in delivering any service at all to the flows, as seen in Section 6.2. This leads to a scheduler behaviour that benefits some clients at the cost of others: even for 36 WT the clients with best service receive page goodput above 150 kbps, at the cost of some other WWW clients who receive no service at all. This starving of some flows to be able to satisfy others happens only among flows of the same type; the VoIP or Bulk flows are not affected.

6.3.4 Tuning θ_{WWW}

The value of θ_{WWW} has a big influence in the service given to WWW flows since it determines, on one hand, the amount of packets that have low priority

Figure 6.17: ECDF of the average page goodput of all WWW clients in the 5 runs.



(a) EXP (b) M-LWDF (c) PeLe (d) PFS (e) caRR

at flow start and, on the other hand, the priority given to the transmission of a packet when the delivered service departs from the requested after the flow's start phase. These results were obtained in Section 6.3.1 for a specific set of parameters (channel bandwidth, requested bandwidth). This section discusses how to tune the value of θ_{WWW} for the traffic pattern characteristic of WWW traffic.

Section 6.3.1 showed that the amount of data that makes up the page's initial phase has a strong influence upon the response time. Thus, θ_{WWW} can be chosen to limit the percentage of pages that are sent entirely in the intial low priority phase. The number of bytes necessary to reach a certain service ν is $B = \nu \cdot \theta_{\mathrm{WWW}} \bar{r}_{\mathrm{WWW}}$, since $\nu = \frac{B/\theta_{\mathrm{WWW}}}{\bar{r}_{\mathrm{WWW}}} = \frac{B}{\theta_{\mathrm{WWW}} \bar{r}_{\mathrm{WWW}}}$. So, one can pre-define that a maximum x% of the pages (corresponding to B_x Bytes according to the page size distribution) should correspond to less than a service threshold ν_{th}, and

$$\theta_{\mathrm{WWW}} = \frac{8 \cdot B_x}{\nu_{\mathrm{th}} \cdot \bar{r}_{\mathrm{WWW}}} \tag{6.1}$$

can be used to ensure that only that percentage x of pages is served entirely without reaching the steep segment of the utility curve. For example, from the distribution of WWW page sizes given in Table 5.2 it is known that 10% of the pages have less than 1.36 kB. With Q_2 and setting the "low priority"-threshold at a service of $\nu_{\mathrm{th}} = 10\%$, the following holds: $8 \cdot 1.36\,10^3 = 0.1 \cdot \theta_{\mathrm{WWW}} \bar{r}_{\mathrm{WWW}}$. For the previous section, $\nu_{\mathrm{th}} = 10\%$ and $x = 10\%$ for $\bar{r}_{\mathrm{WWW}} = 200$ kbps lead to θ_{WWW}=0.544 s.

Figure 6.18: Average user perceived quality of the WWW, VoIP and Bulk clients for varying θ_{WWW}.

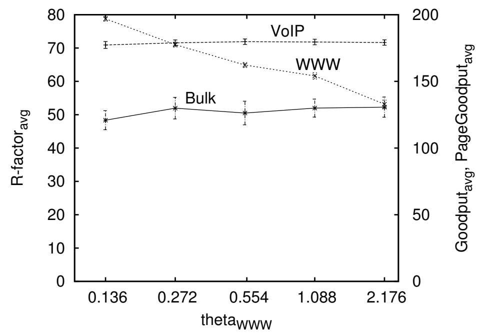


Figure 6.18 shows the performance of the applications in the scenario with 24 WT used in Section 6.3.2 when θ_{WWW} is varied. This corresponds to varying the percentage of pages that are sent entirely at a low priority. The following table shows the values of θ_{WWW} and the corresponding percentage of pages that are completely sent before the service reaches 10% when the requested rate is $r_{\bar{\mathrm{WWW}}} = 200$ kbps[2].

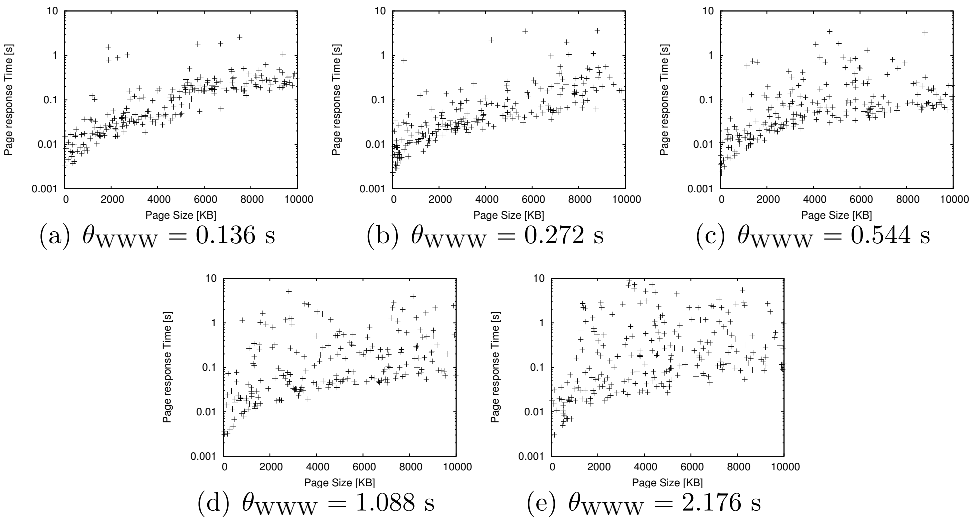
[2]The values were taken from the distribution of the page sizes after the simulation runs, showing which percentage of the pages was actually handled with a low priority.

θ_{WWW}	$B_{10\%}$ [kB]	x%
0.136	0.34	2.4%
0.272	0.68	4.4%
0.544	1.36	7.5%
1.088	2.72	15.8%
2.176	5.44	26.4%

As θ_{WWW} grows, the percentage of pages that receive low priority throughout their duration increases.

The curves in Figure 6.18 show that only the performance of WWW flows is affected by these changes in θ_{WWW}. As expected, as θ_{WWW} grows, the average page goodput obtained by the WWW clients decreases because more pages have low priority. This is illustrated in Figure 6.19, which shows the page size versus page response time for pages of less than 10 kB for $\theta_{\mathrm{WWW}} \in \{0.136, 0.272, 0.544, 1.088, 2.176\}$ and where the increase in response time for short pages for increasing θ_{WWW} can be clearly seen. Figure 6.20 shows the page sizes versus response times for large pages. There the increase in the variations of the page goodput for growing θ_{WWW}, which occurs because the requested rate is enforced over longer periods of time, can be clearly seen.

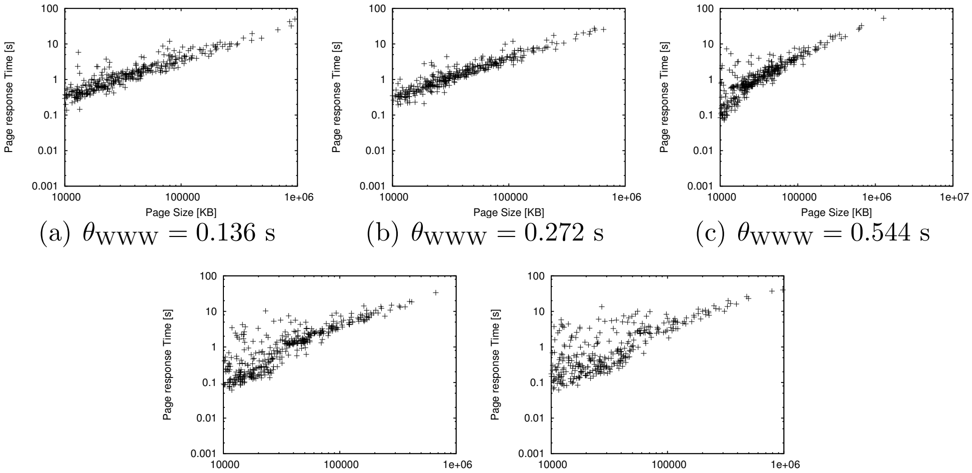
Figure 6.19: Page size versus page response time for pages shorter than 10 kB and varying values of θ_{WWW}.



(a) $\theta_{\mathrm{WWW}} = 0.136$ s (b) $\theta_{\mathrm{WWW}} = 0.272$ s (c) $\theta_{\mathrm{WWW}} = 0.544$ s

(d) $\theta_{\mathrm{WWW}} = 1.088$ s (e) $\theta_{\mathrm{WWW}} = 2.176$ s

Figure 6.20: Page size versus page response time for pages longer than 10 kB and varying values of θ_{WWW}.



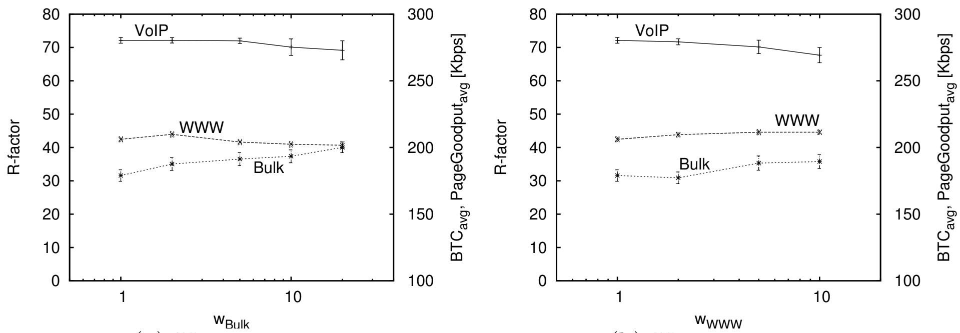
(a) $\theta_{\text{WWW}} = 0.136$ s (b) $\theta_{\text{WWW}} = 0.272$ s (c) $\theta_{\text{WWW}} = 0.544$ s

(d) $\theta_{\text{WWW}} = 1.088$ s (e) $\theta_{\text{WWW}} = 2.176$ s

6.4 Weighting Utility Curves

Up to this point the relative importance of packets from different flow types is determined by the shape of the utility curves alone. But utility curves may be given weights to express different scheduling priority when flows are not satisfied, as described in Section 3.4.4. In this section, the performance of the PeLe scheduler is evaluated in the same scenario with 18 WT used in Section 6.3.2 when varying weights of either the Bulk or the WWW flows. As previously, the requested maximum rate for both flow types, r_{WWW} and r_{WWW}, is 200 kbps; the difference between this scenario and that of Section 6.3.2 is the priority given to packets from a flow when is has received less than the requested resources. In Figure 6.16 the channel is very close to overload. The available bandwidth is enough to satisfy the VoIP and WWW clients and to give the Bulk clients just about the requested BTC.

Figure 6.21: Average perceived quality of all applications when flow weights for WWW and Bulk applications are varied.



(a) Varying w_{Bulk} (b) Varying w_{WWW}

Figure 6.21 shows the perceived quality for varying weight of Bulk flows $w_{\text{Bulk}} \in \{1, 2, 5, 10, 20\}$ and WWW flows $w_{\text{WWW}} \in \{0.5, 1, 2, 5, 10\}$. When the weight of Bulk flows is increased (Figure 6.21–a), the average BTC increases slowly towards the requested 200 kbps. The increase is slow with w_{Bulk} because the gradient of the utility curve of Bulk flows is much lower than that of either other flow type for service above 50%. The cost of the higher priority of Bulk flows is carried mostly by the VoIP flows, whose average perceived quality decreases for $w_{\text{Bulk}} \leq 10$. The WWW flows are intermittent, so that most of the time the competition for the channel happens between VoIP and Bulk flows. Furthermore, the WWW page goodput decreases only slightly when the priority of the Bulk flows increases because almost 50% of the pages are transmitted using the steep part of the WWW utility curve, where even with $w_{\text{Bulk}} = 20$ WWW flows have higher priority than Bulk flows. Only some very large pages might be affected by the lower relative priority of the WWW utility curve at high service.

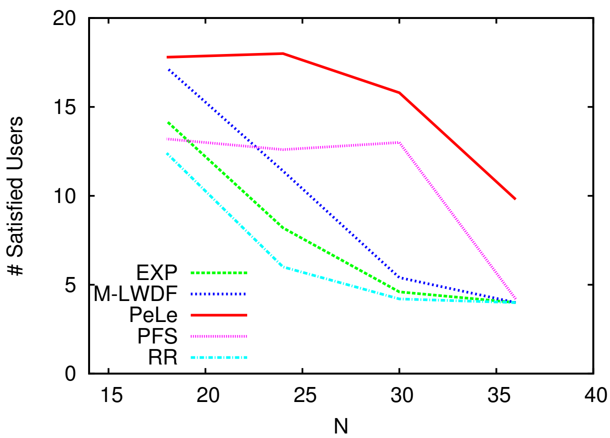
In Figure 6.21–b, w_{WWW} is increased but the page goodput of WWW flows remains unchanged. This is due in part to the fact that 50% of the page downloads are sent with high priority in the steep part of the utility curve because they have very high priority anyway. Further, the WWW flows received almost the requested service anyway, such that increasing their priority does not lead to a better performance, as was seen in Section 3.4.4. However, the higher weight of WWW leads to the decrease of the VoIP perceived quality because more often WWW packets have higher priority than VoIP packets close to the deadline. The slight increase in Bulk quality is due to its taking advantage of resources left over from VoIP flows in periods of bad quality.

6.5 Improved User Quality using PeLe

This section shows the results previously obtained from a different perspective: the scheduling algorithms are evaluated according to the number of users that they can satisfy. Whether a user is satisfied depends on whether his perceived quality metric exceeds a certain threshold according to the following rules

Bulk	$u_{\mathrm{Bulk}}(\overline{\mathrm{BTC}}) \geq 0.9$
WWW	$u_{\mathrm{WWW}}(\bar{g}_{\mathrm{WWW}}) \geq 0.9$
VoIP	R-factor $\geq$70
VoD	$P[\mathrm{DIV}_{500\mathrm{s}} > 10\%] < 10\%$

Figure 6.22: Amount of satisfied users for increasing number of WT in the cell (additional users run WWW clients).



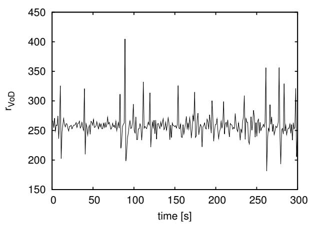
For the audiovisual applications, the thresholds are taken in accordance with the considerations in Sections 5.5.3 and 5.5.4. The thresholds for Bulk and WWW application satisfaction are rules of thumb since there are no reference values for this types of traffic. Furthermore, those rules of thumb use the estimated utility curves that are defined for the FM, and not for performance evaluation. However, it has been argued thus far that those curves do express the expected sensitivity of the user perceived quality to the delivered service.

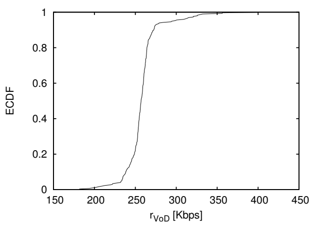
Figure 6.22 plots the number of satisfied users for increasing number of WWW clients, in the same scenario as used in Section 6.3.2. It shows the advantage of using the PeLe scheduler: with the same bandwidth this scheduler provides satisfying perceived quality to a larger number of users than any of the reference schedulers. The caRR satisfies the least users, as it gives equal channel access opportunity to all flows regardless of the application. The EXP scheduler satisfies fewer users than the M-LWDF. The latter provides better service to more Bulk and WWW flows, whereas the EXP provides better quality to VoIP calls. The PFS satisfies the most users after the PeLe because it achieves the highest throughput and distributes it fairly among all users, satifying most WWW and Bulk clients; however, no VoIP client is satisfied because delays are not controlled at all, as has been seen previously.

6.6 Support of VBR Video on Demand

This section studies the support of on-demand video streaming (VoD) and shows the steps towards the definition of a utility curve for a variable bit rate (VBR) video stream. The video chosen for use in the simulations is encoded

Table 6.1: Data rate of the VoD stream used (calculated over non-overlaping 1 s windows) and its empyrical CDF.





percentile	r_{VoD} [kbps]
50	257.01
90	272.70
95	298.85
99	356.15
100	404.77

as VBR with an average bitrate of 256 kbps, but has several rate peaks, as can be seen in the evolution of the video data rate calculated over 1 s intervals shown in Figure 6.1. Accordingly, most of the time the estimtated quality

ξ_{VoD} should fluctuate around the value that corresponds to the average flow data rate $u(\bar{r}_{\mathrm{VoD}})$. Whenever rate peaks occur, the utility curve of the VoD flow should express the importance of supporting them. If it fails to do so, the rate peak has no influence in the scheduling metric, what can lead to large delays and packet drops during the peak and, as a consequence, to reduced user perceived quality. Thus, the parameters of the utility curve must be chosen so that the transmission of packets during a rate peak produce an increase in estimated quality. To achieve this, most of the rate peak should correspond to a part of the utility curve with large gradient.

These considerations are important for designing the utility curve of a VoD stream. If many rate peaks are much larger than the mean rate, then $u(\bar{r}_{\mathrm{VoD}}/r_{\mathrm{VoD}})$ should be lower, so that a large part of the frequent peaks receives high priority. Oppositely, if there are few peaks much larger than the mean rate, they may not need such high priority because even if large delays occur (with eventual packet drops) they are not frequent. The CDF of the data rate of the VoD stream can be used to parametrise the utility curve. Table 6.2–a shows the parameters of the utility curve described by Eq. 3.8 when $r_{\mathrm{VoD}}^{\max}$ is set to the 99-percentile of the 1 s moving window rate for different values of $u(1)$ (corresponding to $r_{\mathrm{VoD}}^{\max}$) and $u(X)$, $X = \frac{\bar{r}_{\mathrm{VoD}}}{r_{\mathrm{VoD}}^{\max}}$. The corresponding utility curves can be seen in Figure 6.2–b, and Figure 6.23 shows the behaviour of the tracked service and estimated quality ξ_{VoD} at the FM. The cell load is low, so that the variations in the tracked service at the FM are due to the VBR nature of the VoD flow and not a consequence of scarce channel resources.

Because for u_1 and u_2 $u(X)$ is located in a steeper part of the utility curve, $\Delta\xi_{\mathrm{VoD}}$ is larger than for u_3 or u_4 for the same service variations. So, setting $u(X)$ at a point of high gradient gives the VoD flow higher priority if the rate

Table 6.2: Possible utility curves for VoD.

	$u(X)$	$u(1)$	a	b
u_1	0.6	0.99	15.18	0.697
u_2	0.6	0.999	23.55	0.707
u_3	0.8	0.99	11.63	0.605
u_4	0.8	0.999	20.00	0.655

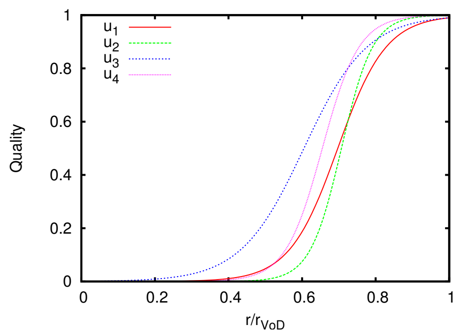
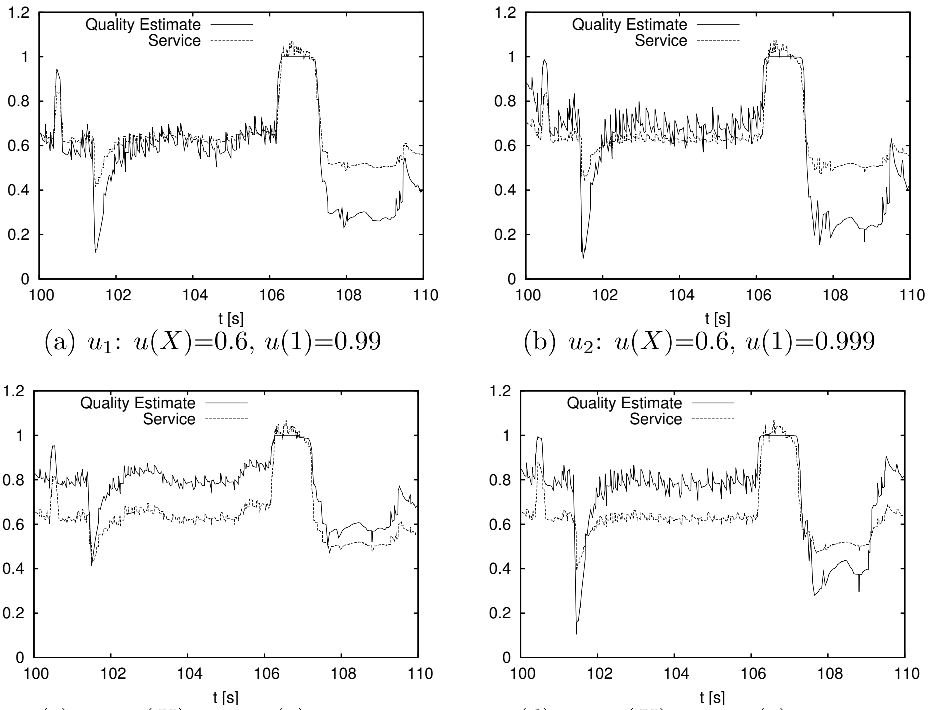


Figure 6.23: Evolution of the service and estimated quality ξ_{VoD} at the FM for an exemplary VoD flow for different parameter settings of the VoD utility curve.



(a) u_1: $u(X)$=0.6, $u(1)$=0.99

(b) u_2: $u(X)$=0.6, $u(1)$=0.999

(c) u_3: $u(X)$=0.8, $u(1)$=0.99

(d) u_4: $u(X)$=0.8, $u(1)$=0.999

diverges from the average. Comparing u_1 and u_3 with u_2 and u_4, the latter curves give higher priority to the flow at the rate peaks. In all four cases, the large peak at $t \approx 106$ s is cut-off from the quality estimate. Since the 99-percentile was chosen as the maximum requested rate, the flow is satisfied at less than the actual maximum rate generated by the application. In case of scarce channel resources, those peaks are not supported by the scheduler since they are not expressed as a quality increase.

For the further support of the VoD flow in this work, the utility curve u_2 shall be used because the video chosen contains a lot of movement and the loss of packets at rate peaks severely impairs the perceived quality.

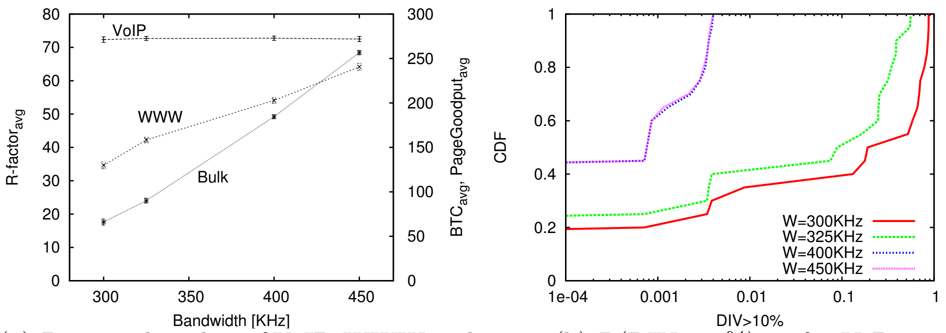
6.7 Performance with Bulk, WWW, VoIP and VoD

This section evaluates the performance of the PeLe scheduler with a traffic mix containing all 4 studied traffic types. The scenario consists in 16 WT sharing a wireless channel of variable bandwidth $W \in \{300, 325, 350, 400, 450\}$: 4 Bulk clients, 4 VoIP clients, 4 WWW and 4 VoD clients. Recall that the main metric for the perceived quality of VoD flows is the Distortion in Interval (DIV), defined in Section 2.4.2: the percentage of 20 s intervals in which more than 10% of the frames have a lower MOS than the original video. When too many intervals have a large percentage of frames with MOS lower than the original video, the perceived quality is bad; so, it is desirable to have low percentage of intervals with DIV>10%.

Figure 6.24–a shows the average perceived quality of the VoIP, WWW and Bulk clients and Figure 6.24–b shows the CDF of the percentage of time that the DIV of each VoD client is above 10%. The PeLe can provide high quality to all flows when W=450 kHz. Because all clients receive the requested service the Bulk flows receive more than requested as TCP can take advantage of the excess resources. At W=400 kHz, all flow types receive their requested service: the VoIP clients maintain a high quality, the Bulk have a BTC close to 200 kbps, the WWW clients also see a page goodput close to 200 kbps, the VoD clients see no change in the distribution of the DIV. When the channel bandwidth is further decreased, the BTC of the Bulk clients is affected the most since their quality is the least sensitive to low received service. As a consequence, the Bulk flows' throughput is reduced by the scheduler in periods of congestion, for example when many WWW flows are simultaneously active. The VoIP clients keep their high quality as the bandwidth continues to decrease due to their high priority, as was previously explained. The WWW also see their quality decrease, but not as strongly as the Bulk clients because much data is sent with high priority on the steep part of the utility curve.

With regard to the VoD traffic, as the bandwidth decreases and it become unfeasible to support all clients some of them receive bad quality whereas other

Figure 6.24: Perceived quality of VoIP, WWW and Bulk clients and CDF of the DIV for VoD clients when the channel bandwidth is varied.



(a) Perceived quality of VoIP, WWW and Bulk

(b) P(DIV>10%)<x for VoD

keep a low DIV, as can be seen in Figure 6.24–b. For W≥400 kHz all VoD clients had DIV> 10% less than 5% of the time. In contrast, for W=325 kHz only 50% and for W=300 kHz only 38% of the users have less than 10% of the time with less than 10% of the frames with MOS lower that the original encoded video. Due to the low available bandwidth the PeLe can no longer support all VoD clients and, because of the inelastic VoD utility curve, some VoD clients are no longer supported. This table shows the correlation coefficient between the average attenuation of each WT receiving a VoD flow and its percentage of time with DIV>10%. The third column shows the value used to test the hypothesis of no correlationas as described in Chapter 4 of [141]: if the value is not in the interval [-2.086;2.086] the inputs can be considered correlated with a confidence level of 95%.

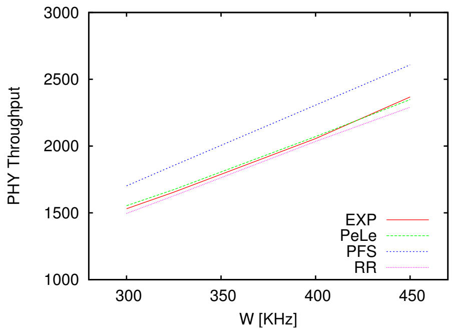
W	r_{xy}	$w = \frac{\sqrt{(N-3)}}{2} \ln\left[\frac{1+r_{xy}}{1-r_{xy}}\right]$
300	0.649407	3.19241
325	0.584497	2.75946
400	-0.131725	-0.546292
450	-0.129024	-0.534963

The results in the table show that when resources are scarce the quality of the VoD clients on WT with lower average link quality is degraded. Because VoD packets are often large, they must often be transmitted using a robust modulation and occupy the channel for longer time. This decreases their priority because of the estimated impairments in the quality of competing flows, both VoD flows and others. Therefore, when VoD packets compete for scarce resources the ones with better link quality are served preferentially. The inelastic utility curve eventually leads the scheduler to “drop” the flows whose WT have worse links because it costs too much to transmit to them.

6.8 Different Paradigms for Wireless Multi-Flow Scheduling

Finally, this section compares the different scheduling approaches studied according to their support of multiple applications and to the total cell throughput. For this purpose, scenarios with application mixes containing all 4 flows types are present. The scenario is the same as in the previous section, with 16 WT in a cell whose bandwidth is varied between 300 and 450 kHz. Because the EXP and the M-LWDF perform similarly but the first has previously proven to give better support to audiovisual flows, the M-LWDF is not included in the study. Figure 6.25 shows the overall physical layer throughout in the cell for the four schedulers. The PFS achieves the highest cell throughput

Figure 6.25: PHY throughput obtained with the PeLe, EXP, PFS and caRR schedulers for varying channel bandwidth.

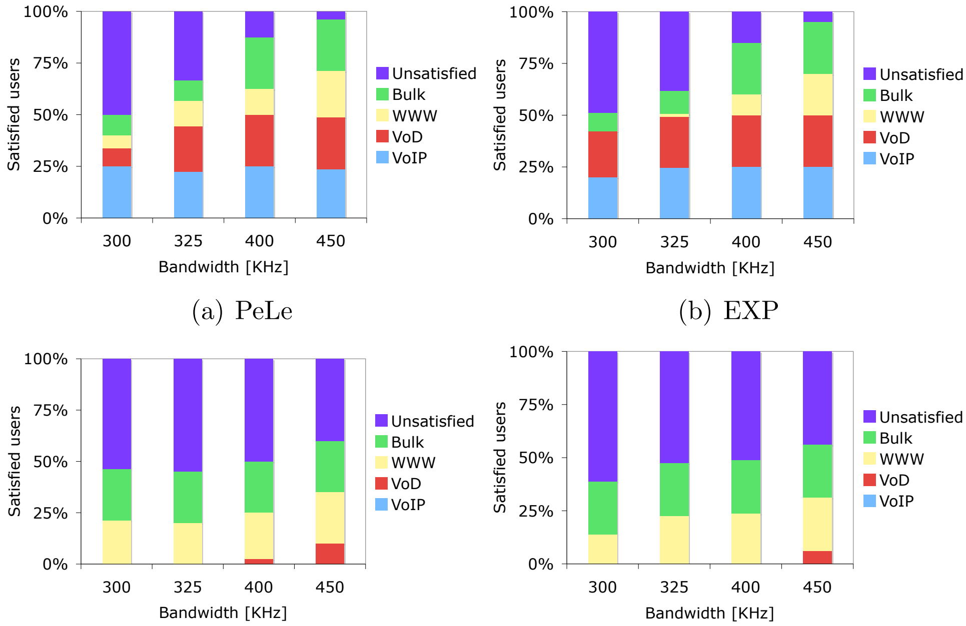


which is around 8% better than the next best scheduler. The lowest throughput is achieved by the caRR scheduler, at around 12% below the PFS. The EXP and PeLe have overall throughputs differing from eachother by at most 1.5%. The EXP is better than the caRR by 1–3.3% and worse than the PFS by 9.2–10.8%, whereas the PeLe lies above the caRR by 1.8–3.8% and below the PFS by 8.7–10.2%. The best overall throughput achieved by the PFS scheduler represents a gain of only 12% with respect to the worst scheduler studied in the scenario.

Figure 6.26 shows the number of satisfied users according to the criteria defined in Section 6.5 discriminated per application type. For the PFS and the caRR, the best and worse in terms of throughput, the amount of satisfied users does not differ much. Even though the PFS satisfies more users, neither scheduler can provide quality to audiovisual flows because the schedulers do not take any flow requirements into account. So, what happens to the extra 12% data trasnmitted by the PFS scheduler? Figure 6.27 shows the CDF of the application specific metrics for VoD, WWW and Bulk flows per scheduler. The results for VoIP are not shown as, on the one hand all VoIP calls experience good quality with either PeLe or EXPand, on other hand, all experience bad quality with PFS or caRR. Comparing Figures h with k and i with l it becomes clear that the PFS delivers more data to both WWW and Bulk clients than the caRRdoes. However, since these clients request only 200 kbps, the higher received data rate does not lead to an increase in perceived quality and the clients do not benefit from the extra service delivered by the PFS.

Figures 6.26–a and b show that the amount of satisfied users for the EXP and PeLe schedulers is similar. For lower channel load—W $\geq$400 kHz—the schedulers perform similarly, delivering satisfying quality to the audiovisual flows and dividing the rest of the bandwidth among the TCP-based flows. This is also seen by comparing the curves for W $\geq$400 kHz in Figures 6.27

Figure 6.26: Number of users satisfied with the service delivered by the PeLe, EXP, PFS and caRR schedulers for varying channel bandwidth.



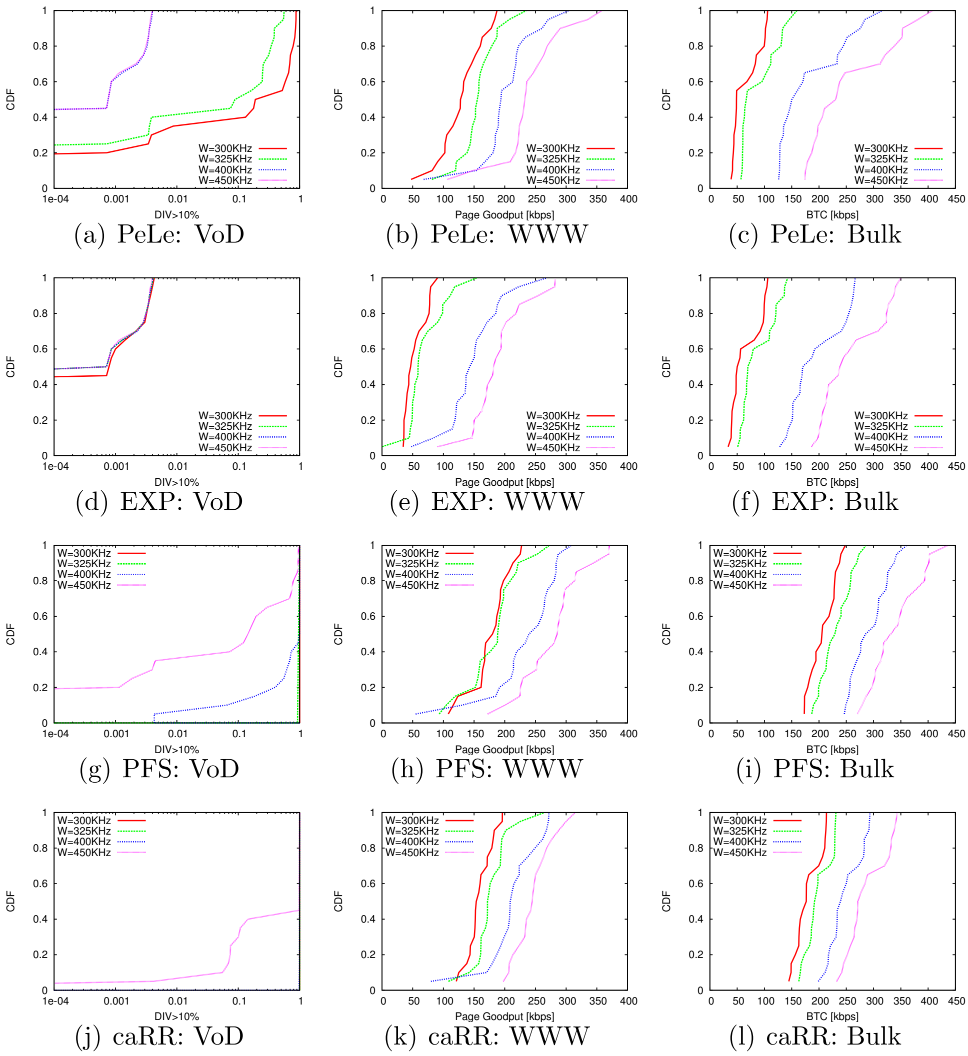
(a) PeLe (b) EXP (c) PFS (d) caRR

a with d, b with e, and c with f. The main difference lies in the tails and range of the page goodput and BTC for W=400 kHz: although the minima are similar, the PeLe is unfairer, having a larger range and maxima of the page goodput and BTC. When flows are satisfied (the FM estimated quality is 1), giving them more service than they requested does not produce any increase in quality, so the allocation is made randomly and some flows end up receiving more resources than others.

The two schedulers differ when resources become scarcer. While the EXP keeps up a good quality delivered to all audiovisual flows, the PeLe keeps some clients of each application type satisfied. The EXP keeps the same service to audiovisual flows when the available bandwidth decreases (Figure 6.27–d) because they have delay weights 1000 times larger than the TCP-based flows. This leads to the lower quality of WWW and Bulk flows (Figures 6.27–e and f). Also if the weights were lower or the traffic mix different, the EXP might not be able to satisfy the VoIP clients even at high bandwidth.

The PeLe weighs the effect of sending packets of a flow on other flows, and when the available channel bandwidth decreases, it is no longer possible to support all VoD flows without impairing several other flows. Thus, it is better for the overall quality in the cell to support less VoD flows, because that enables the support of several other flows. Due to this weighing in the scheduling metric, the PeLe can support several flows of each application type. It also differentiates between the two types of TCP-based flows according to the elastic and inelastic utility curves, enabling WWW clients to see a higher goodput than the Bulk clients. This is achieved without changing any FM or scheduler parametes, i. e. the parametrisation of the scheduler is done for a certain traffic type, and remains constant independently of the traffic mix.

Figure 6.27: CDF of the application specific quality metrics obtained with the PeLe, EXP, PFS and caRR schedulers for varying channel bandwidth.



(a) PeLe: VoD (b) PeLe: WWW (c) PeLe: Bulk

(d) EXP: VoD (e) EXP: WWW (f) EXP: Bulk

(g) PFS: VoD (h) PFS: WWW (i) PFS: Bulk

(j) caRR: VoD (k) caRR: WWW (l) caRR: Bulk

6.9 Conclusions

This chapter evaluated the performance of the PeLe scheduler, showing how it takes advantage of application characteristic traffic patterns and soft QoS perception to improve user perceived quality. The design of utility curves for the example applications was described in detail and relevant issues highlighted. Namely, that the utility curve should express the sensitivity of the application to the delivered network service and that the interval θ over which that delivered service is calculated should express the timescale of the user perception of service changes. Further, application characteristic traffic patterns should be taken into account, as was seen for VBR VoD flows. In summary, this chapter described how an FM and a utility curve can be designed for an application whose traffic pattern and sensitivity to network service are known and what aspects should be taken into account.

The PeLe scheduler was thoroughly evaluated in terms of user perceived quality provided to the different application types in a wide range of load scenarios. The load was varied by changing both the channel bandwidth and the representation of the different application types in the traffic mix. The results show that the PeLe scheduler behaves as expected. It allocates the wireless channel so that the resulting service seen by the application leads to high perceived quality. Furthermore, when resources are scarce, the degradation of the service delivered takes place according to the sensitivity of each application to service differing from that requested, as expressed by the utility curves.

Using simulations, several state-of-the-art schedulers were compared to the proposed PeLe scheduler in terms of application perceived quality and of the total channel throughput in many load scenarios and with varied traffic mixes. The results show that QoS support can only be achieved at the cost of total cell throughput. However, under realistic traffic conditions, the best overall cell throughput (achieved by the PFS) is at most 12% better than the worst other scheduler—not a large variation. However, the PFS does not support audiovisual applications since it does not enable differentiated support and cannot cope with the different service requirements.

Chapter 7

Accuracy of Wireless Link Quality Prediction

Channel adaptive techniques require a prediction of the behaviour of the wireless link in order to adapt the transmission accordingly.

Section 7.1 introduces the problem of predicting wireless fast fading, presents a state-of-the-art overview and introduces three heuristics for the prediction of signal strength. Afterwards, the results of a simulative evaluation of the accuracy of those heuristics using WLAN channel traces (Section 5.1) are presented. An evaluation of the effect of prediction errors in the performance of a threshold-based adaptive modulation scheme is presented in Section 7.3. Based on statistics of the effects of prediction errors, an improvement is studied proposed and evaluated in Section 7.4.

7.1 Wireless Link Prediction

7.1.1 Related Work

Reference [145] introduces the Long Range Predictor (LRP): a prediction algorithm based on an auto-regressive model (AR) for the fading process that uses the maximum entropy method (MEM) for the estimation of the AR model coefficients. The link is undersampled to achieve a longer prediction horizon with the same AR model order and complexity. The undersampled predicted values are then interpolated to get prediction values at the data sampling rate. The metric used for accuracy of the prediction is the mean squared value of the prediction error—MSE. The performance of the algorithm is very good for a stationary Jakes linkbut degrades for non-stationary link parameters. In [146] the authors remark that prediction accuracy is reduced by noisy samples, short observation intervals and mismatch in the AR coefficients (due to non-stationarity and short prediction windows). However, the degradation is not quantised.

Reference [147] proposes to model the link as an FIR adaptive filter

$$s(n) = \sum_{k=1}^{n_k} h_k(n)u(n-k) + w(n)$$

with time-varying taps $h_k(n)$. The authors argue that it is easier to predict the taps because they vary in a slower timescale than the link fading [148] and propose two predictors for the complex taps $h_k(n)$ one based on an AR model using least squares estimation and another one based on Kalman filters. The power of each tap can be calculated from the squared magnitude of the tap and the signal power by adding the power of all taps. As above, the authors identify the need for noise reduction and propose Wiener smoothers, despite the delay that they introduced. They also use sub-sampling to extend the achievable prediction range. The metric used to evaluate the performance of the predictors is the normalised mean squared error (NMSE) of the signal power. The accuracy of the prediction algorithms according to this metric is evaluated on channel sounding measurements from a cellular provider in an urban environment. Although the algorithms presented in [145] and [147] are evaluated in different scenarios and under different channel conditions (according to the data presented in the references) the second outperforms the first in terms of NMSE for the same fraction of wavelength. Unfortunately, these algorithms could not be reproduced from the information available in the papers, thus they cannot be used as reference for comparing the heuristics presented here.

Reference [149] studies a wireless link predictor based on neural networks that performs better than the Modified Covariance algorithm [150], which is used for reference here. However, neural networks are far more complex than the auto-regressive model or adaptive filters and the comparison would be unfair.

The literature in this are often proposes to use the following heuristic for link prediction [96, 98, 97]: assume that the link will not change from the last available sample, i. e. use the last reported link value as a prediction for the future link value. Refernce [98] evaluates the accuracy of this heuristic using a good/bad Gilbert-Elliot link model. However, that link model is inappropriate for a system that enables the adaptation of transmission parameters like modulation or coding.

In the single existing comparative study known at the beginning of this work [150], Semmelrodt et al. propose several methods for estimating the coefficients of an AR model and adaptive filters, and evaluate the prediction results using theoretical Rayleigh models of the wireless link. The metric used is the normalised mean squared value of the prediction error (NMSE) for a certain fraction of wavelength. Comparing the results shown in references [150], [145] and [147], the prediction of the signal envelope obtained with the best algorithms in [150] are more accurate than LRP [145] or [147] for one wavelength.

The predictors presented are complex and, except for the neural networks, require different calibration for specific environments. Further, they are sensitive to the time-variant nature of the wireless channel and to noise, but it has not been studied to which extent. In this scenario, heuristics are an attractive low-complexity alternative, but an evaluation of their accuracy and a comparison to existing predictors is missing. Hence the importance of evaluating the accuracy of the OS link prediction heuristic [96, 98, 97] and comparing it to a reference scheduler—the Modified Covariance (MC) that performed best in the comparative study [150]. Additionally to the OS, two other slightly more complex prediction heuristics are also studied.

7.1.2 Heuristics for Wireless Link Prediction

This section formally describes the signal strength prediction heuristics that are evaluated in the next section. Let $s(i)$ be the received signal amplitude at discrete time instant i and $\widehat{s(i)}_h$ the predicted received signal amplitude at time $i + h$, as predicted at time i. At time i, $s(i)$ is the newest available link sample and $\widehat{s(i)}_1$ is the earliest value to be predicted and the prediction horizon $h > 1$. The following three heuristics for wireless link prediction are compared in the next section:

One Step (OS)

$$\hat{s}(i)_h = s(i) \tag{7.1}$$

Moving Average (MA)

$$\hat{s}(i)_h = \frac{1}{N}\sum_{k=0}^{N-1} s(i-k), \tag{7.2}$$

i. e. the predicted value is the average of the last N link samples.

Linear Prediction (LP)

$$\hat{s}(i)_h = (i-h) * \frac{\sum_{k=0}^{N-1} s(i-k) * k - \sum_{k=0}^{N-1} s(i-k) \sum_{k=0}^{N-1} k}{\sum_{k=0}^{N-1} k^2 - \left(\sum_{k=0}^{N-1} k\right)^2}, \tag{7.3}$$

i. e. the predicted value is given by a linear regression over the last N link samples.

Whenever perfect prediction is mentioned, a “God’s view” is assumed, i. e. it is assumed that the future can be foreseen and $\hat{s}(i)_h = s(i + h)$.

7.1.3 Reference Predictor

As a reference, the prediction algorithm that performed best in the comparative study [150]—Modified Covariance (MC)—is used. The MC models the received signal $s(i)$ as an AR process of order p:

$$s(i) = \sum_{k=1}^{p} a_k \cdot s(i-k) + e(i)\, i = 0 \ldots N-1 \tag{7.4}$$

where $e(i)$ is a complex white Gaussian noise process and a_k the coefficients of the AR polynomial. A linear predictor is used to extrapolate the behaviour of the process beyond the available link samples [150, 151]:

$$\hat{s}(i) = \sum_{k=1}^{p} a_k \cdot s(i-k) \tag{7.5}$$

To obtain a predicted value it is necessary to estimate the coefficients of the AR model a_k. Since the wireless link varies in time due to movement in the environment and of the sender or receiver, the model is also time-variant. However, assuming that the model remains invariant over short periods of time, the coefficients can be estimated by solving a system of linear equations. Thus, the time-variant wireless link is modeled as an AR process with time-variant coefficients $a_k(i)$, that are recalculated each time that prediction is required. The MC prediction algorithm uses the least squares method to solve the system of linear equations for the calculation of the model coefficients $a_k(i)$. Reference [150] proposes the use of model order p =15, but here p is varied and the value with the best results is chosen. The prediction of future samples $\hat{s}(i)_h$, $h \geq 1$, is obtained by cascading linear predictors, each using the previously extrapolated signal samples as input, as shown in Figure 7.1. Since the input to predictors down the chain are extrapolated samples (themselves inaccurate), the prediction error propagates for increasing horizons.

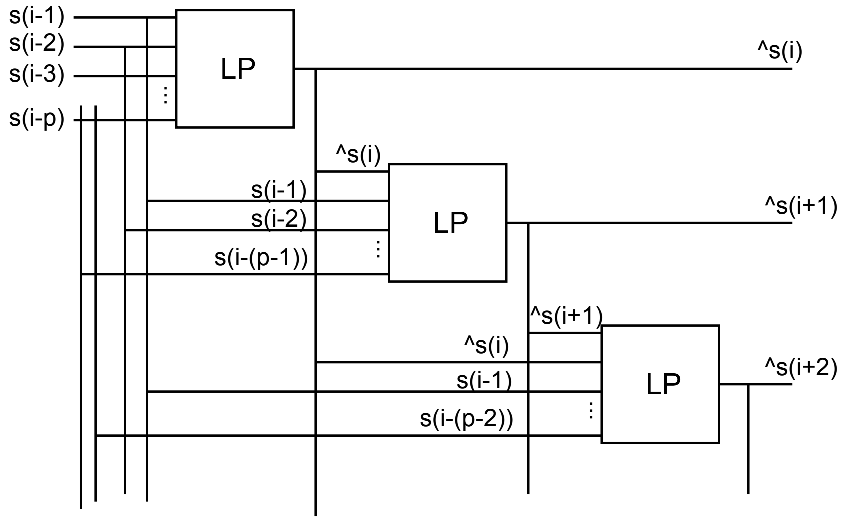
7.2 Accuracy of Signal Prediction

7.2.1 Simulation of Link Prediction and Metric

At each discrete time instant i, the values of the amplitude of the received signal for prediction horizons $h \in [1; 15]$ ms are predicted as defined in Section 7.1. Then, the prediction error—the difference between the predicted and the actual signal amplitudes $e(i)_h = \hat{s}(i)_h - s(i+h)$—are calculated for each horizon h.

The metric used to compare the performance of the channel predictors is the the mean squared value of the normalised prediction error (NMSE),

Figure 7.1: Cascaded linear predictors for prediction horizons beyond h = 1 ms. $\hat{s}(i)$ serves as input to the linear predictors for $h > 0$, $\hat{s}(i)_1$ is input to the linear predictors for $h > 1$ and so on. As a consequence, the inaccuracies in $\hat{s}(i)$ cause inaccuracies in $\hat{s}(i)_1$, which increases further in $\hat{s}(i)_2$, i. e the prediction error grows down the chain of linear predictors—error propagation.



similarly to [145, 147, 150]:

$$\mathrm{NMSE}(h) = \frac{1}{K}\sum_{i=0}^{K-1}\frac{\left(e(i)_h\right)^2}{\sqrt{\frac{1}{K}\sum_{j=0}^{K-1}|s(j)|^2}}, \tag{7.6}$$

where K is the number of channel samples. The metric is expressed in dB

$$\mathrm{NMSE}(h)[\mathrm{dB}] = 10 \cdot \log(\mathrm{NMSE}(h)). \tag{7.7}$$

The more negative the values, the higher the accuracy of the predictor. An NMSE of 0 dB expresses errors with power similar to the signal power.

The significance of the results is calculated in terms of the 95% confidence interval of the NMSE

$$\mathrm{NMSE}(h) \pm t_{\infty,0.975}\sqrt{\frac{S_{e_h}^2}{K}}, \tag{7.8}$$

where $t_{\infty,0.975} = 1.960$ from Table T.1 in reference [140], and $S_{e_h}^2$ is the variance of the normalised squared error

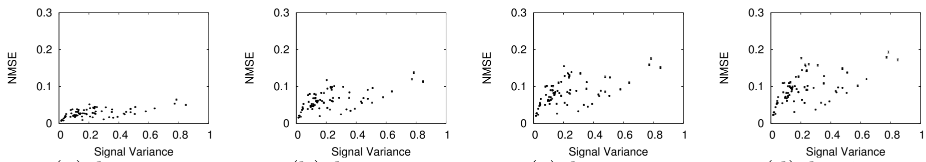
$$S_{e_h}^2 = \frac{1}{K}\sum_{i=0}^{K-1}\left(\frac{e(i)_h}{\sqrt{\frac{1}{K}\sum_{j=0}^{K-1}|s(j)|^2}} - \mathrm{NMSE}(h)\right)^2. \tag{7.9}$$

7.2.2 Comparison of the NMSE

This section shows the MSE of the predicted signal with its 95% confidence interval plotted against the variance of the measured received signal. Figure 7.2 shows this for short horizons ($h \in \{1, 2, 3, 4\}$ s) and Figure 7.3 for large horizons ($h \in \{1, 5, 10, 15\}$ s) for the 4 predictors studied. The figures show the scattering of the MSE for different predictors and horizons for different variance of the received signal. More details of the causes of prediction errors can be found in the technical report [152].

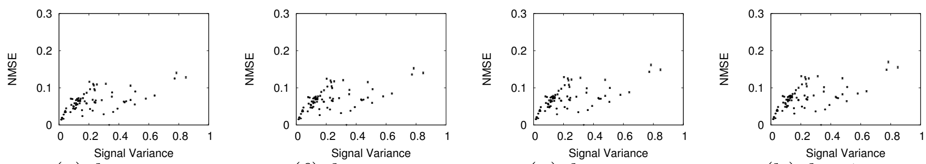
Figure 7.2: NMSE (and 95% confidence interval of the squared error) of the signal strength prediction error for the predictors studied and short prediction horizons. OS: One Step; MA: Moving Average; LP: Linear Prediction; MC: Modified Covariance

OS



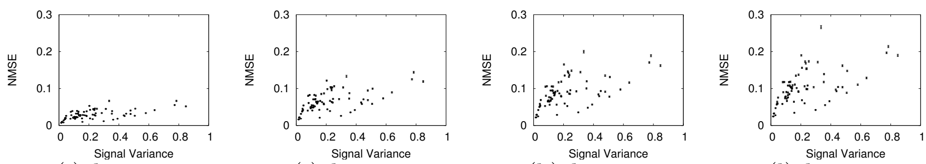
(a) h=1ms (b) h=2ms (c) h=3ms (d) h=4ms

MA N=10



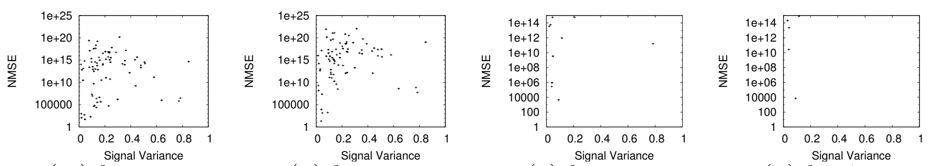
(e) h=1ms (f) h=2ms (g) h=3ms (h) h=4ms

LP N=30



(i) h=1ms (j) h=2ms (k) h=3ms (l) h=4ms

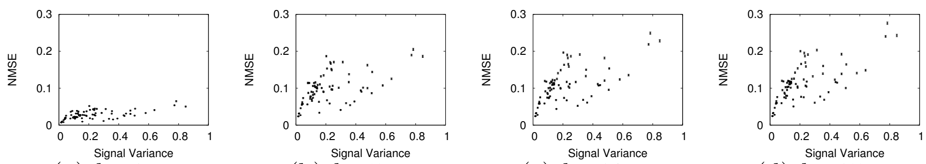
MC p=30



(m) h=1ms (n) h=2ms (o) h=3ms (p) h=4ms

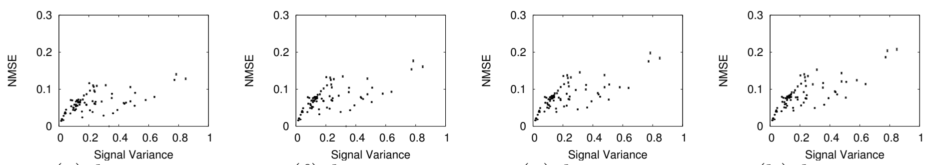
Figure 7.3: NMSE (and 95% confidence interval of the squared error) of the signal strength prediction error for the predictors studied and large prediction horizons. OS: One Step; MA: Moving Average; LP: Linear Prediction; MC: Modified Covariance

OS



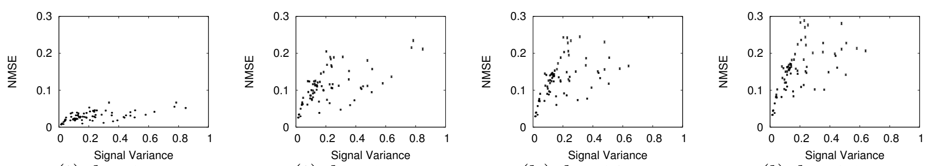
(a) h=1ms (b) h=5ms (c) h=10ms (d) h=15ms

MA N=10



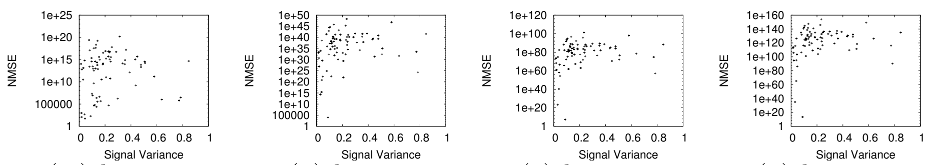
(e) h=1ms (f) h=5ms (g) h=10ms (h) h=15ms

LP N=30



(i) h=1ms (j) h=5ms (k) h=10ms (l) h=15ms

MC p=30



(m) h=1ms (n) h=5ms (o) h=10ms (p) h=15ms

First of all, it is clear that for all horizons and variances, the reference predictor MC is inadequate for the prediction of the signal strength as represented by a WLAN card at the link layer. This is due to the wrong tracking of the coefficients of the AR model due to noisy samples, which lead to an inaccurate AR model and consequent inaccurate prediction for $h = 1$. Since prediction for farther horizons is obtained by cascading linear predictors (see Section 7.1.3), the error propagates, leading to huge errors for larger horizons.

For all three heuristics, the NMSE increases with the prediction horizon and increasing variance of the signal strength, i. e. the heuristics perform better for short prediction horizons and for links with little variations. These behaviours are expected as the heuristics rely on the assumption of minor link variation, which is less valid for high signal variance and for larger intervals between the prediction instant i and the time of the predicted value.

The influence of the number of samples used in the prediction, N for the MA and LP predictors, has also been studied in technical report [152]. Because they are noisy, the measured signal samples change more sharply than they do for Rayleigh links; thus, the LP with fewer samples leads to high gradients that lead to big prediction errors for $h > 2$ ms. In contrast, when the regression uses many samples, the output of the LP varies only little around the average signal strength. In general, the MA and LP produce lower NMSE the more samples N they use, but they only predict the average signal strength and not the variations. The low NMSE is then due to the variations being of small amplitude because of the low mobility environments (Section 5.1). Since the predicted signal strength is close to the average signal strength, the NMSE does not increase with the prediction horizon.

For $h \leq 2$ ms, the OS and LP perform best due to the small channel variations within this time interval. This is only true for this shortest prediction horizon. Otherwise the highest accuracy is achieved by the MA heuristic with N=30. Since the link does not change much in the interval considered, and because signal samples are noisy, the MA heuristic using many samples is intuitively the best way to predict the link behaviour. However, this also means that for $h > 2$ ms, the heuristics studied only predict the average signal strength.

7.3 Effects of Link Prediction Errors on an Adaptive Rate Scheme

Although the metric used above, the NMSE, is frequently used for comparison of prediction algorithms, its meaning for link adaptive mechanisms is hard to grasp, i. e. it does not express how severely the prediction errors affect the performance of a link adaptive mechanism. This section studies the effects of the prediction errors on the performance of the threshold-based adaptive modulation scheme described in Section 5.2.1.

The scenario consists of a single transmitter-receiver pair, where the average received SNR is 20 dB and the channel bandwidth $W = 200$ kHz. The adaptive rate scheme is used with packets of different lengths $L \in \{400, 800, 1200\}$ bits. It also considers varying feedback delay of the link signal samples $d \in \{1, 2, 4, 6, 8\}$, since in a realistic scenario the signal strength is measured at the receiver and then fed back to the transmitter. Table 7.1 shows the modulation thresholds for $\mathrm{PER}_{\max} = 10^{-3}$ and the packet durations for each modulation. The performance is measured as packet loss rate (PLR):

Table 7.1: SNR thresholds for changing the modulation ($\mathrm{PER}_{\max} = 10^{-3}$) and duration of the packets on the channel for $W = 200$ kHz.

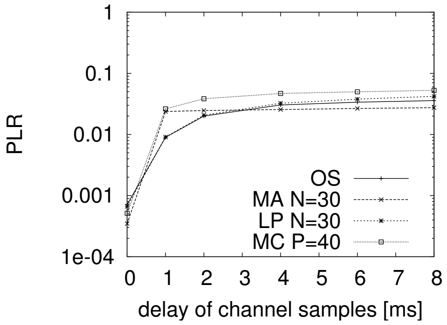
		SNR threshold (per Symbol) [dB]			packet duration [ms]		
Modulation	M	L=400	L=800	L=1200	L=400	L=800	L=1200
BPSK	1	10.18	10.46	10.60	2	4	6
QPSK	2	13.21	13.47	13.63	1	2	3
8PSK	3	18.37	18.65	18.81	0.67	1.33	2
16PSK	4	24.09	24.39	24.55	0.5	1	1.5
32PSK	5	29.98	30.28	30.44	0.4	0.8	1.2
64PSK	6	35.92	36.20	36.38	0.33	0.67	1

$$\mathrm{PLR} = \frac{\#\ \text{transmitted packets} - \#\ \text{correctly received packets}}{\#\ \text{transmitted packets}}. \tag{7.10}$$

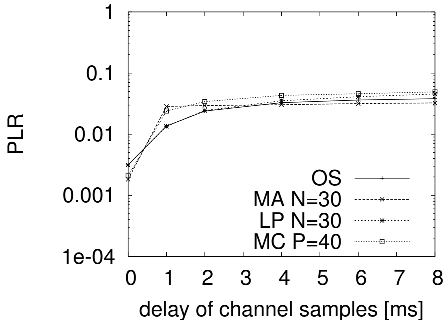
Figure 7.4 shows the PLR for adaptive modulation with all studied predictors, where $N = 10$ and 30 was used for the MA and LP, respectively, and $p = 40$ for the MC reference predictor. A feedback delay of 0 stands for instantaneous feedback of the link samples.

For instantaneous prediction, the PER target of 10^{-3} is met only for the shortest packets ($L = 400$). The maximum duration of short packets (see Table 7.1) makes only short prediction horizons necessary for the adaptation,

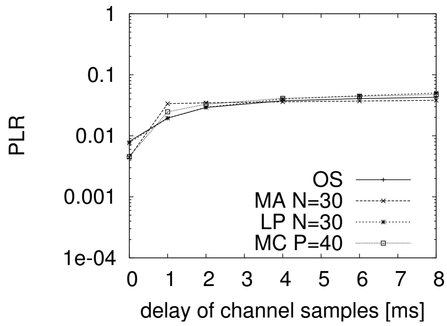
Figure 7.4: Packet loss rate (PLR) on the wireless link for delayed feedback of signal strength information for different packet lengths.



(a) L=400 bits



(b) L=800 bits



(c) L=1200 bits

whereas long packets require the adaptive scheme to “look” at predicted signal values farther in the future. According to the results in the previous section, prediction errors are much larger for $h \leq 2$ ms but these prediction horizons are not needed for $L = 400$ Bytes if the feedback of signal strength values is instantaneous. For larger packets the large prediction errors lead to an increased PLR, which no longer meets the requested target of 10^{-3}.

There is a big increase in the PLR when the signal strength values are delayed. Since the PLR for instantaneous feedback of signal strength values is smaller for shorter packets, it suffers a larger degradation when the prediction is delayed. If the feedback values are delayed more than 2 ms, the PLR for the MA and the reference predictor, MC, has values that do not change with the packet length or feedback delay. Thereby, the MA has the lowest PLR and MC the highest (in accordance with the results in Section 7.2.2). The LP and OS perform similarly with a PLR between that of the MA and MC. For prediction delays below 2 ms, the adaptive modulation performs better with the LP N=30 and the OS than with the other predictors, as expected from the results in Section 7.2.2.

Figure 7.4 also shows that there are only small differences in the best performance obtained with the different predictors if the feedbacked link values are delayed. Furthermore, the PLR hardly depends on the feedback delay for the LP and OS for delays larger than 2 ms. For the MA and MC there is no increasing loss due to an increase in the feedback delay because the errors do not increase with increasing prediction horizon, as was seen in Section 7.2.2. Recalling that the OS predictor is the simplest to implement and performs only slightly differently from the other heuristics, it can be concluded that the adaptive modulation scheme does not benefit from using a more complex heuristic. Therefore, in the following only the OS predictor shall be used.

7.4 Reducing the Effects of Link Prediction Errors

The results in the previous section show that errors in the prediction of the received signal can lead to a large number of packet errors and, consequently, to a performance degradation of link-adaptive mechanisms. Furthermore, the transmission rate chosen by the adaptive modulation scheme is often used to drive the decisions of a packet scheduler, in which case the effects of a wrong modulation choice can be even more devastating.

This section studies changes to the previously used threshold-based adaptive modulation scheme so that the packet errors due to wrong choice of modulation is reduced. The modified adaptive modulation scheme chooses a more robust modulation than the one corresponding to the predicted SNR value if the probability is high that the latter modulation scheme would lead to a packet loss. This is because the link is actually in worse condition than predicted. Therefore it is necessary to consider the following trade-off: whenever a conservative modulation is chosen, the total amount of data transmitted is lower and the channel is inefficiently utilised.

The increased amount of packet losses due to link prediction errors is caused by the use of the wrong modulation. If a prediction error leads to the choice of the same modulation as the perfect prediction scheme there is no difference in the outcome. Consequently, the following aspects need to be studied:

- how often is the wrong rate chosen (this depends on the prediction horizon and average SNR relative to modulation changing threshold), covered in Section 7.4.1;

- how often does choosing the wrong rate lead to a packet error, covered in Section 7.4.2.

7.4.1 Wrong Rate Choice

For the OS predictor ($\hat{s}(i)_h = s(i)$) a wrong modulation occurs as a consequence of a prediction error whenever $s(i)$ (the predicted value) and $s(i+h)$ (the actual value) are on opposite sides of the modulation changing threshold. How often that occurs depends on the difference between the average signal value $\bar{s}$ and the modulation changing threshold. The modulation changing threshold is set at 0 dB, and a constant value $\bar{s}$ added to $s(i)$ and $\hat{s}(i)_h$. Assuming constant noise power, the threshold and the signal can be expressed in s instead of SNR. Then the following amounts are counted and expressed as fractions of the total number of modulation choices:

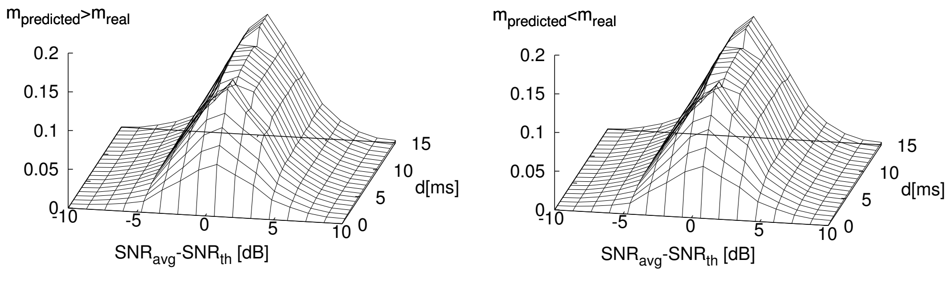
- $\hat{m} > m$: $20 \cdot \log(\bar{s}) + 20 \cdot \log(s(i)) > 0 \wedge 20 \cdot \log(\bar{s}) + 20 \cdot \log(s(i+h)) < 0$
- $\hat{m} < m$: $20 \cdot \log(\bar{s}) + 20 \cdot \log(s(i)) < 0 \wedge 20 \cdot \log(\bar{s}) + 20 \cdot \log(s(i+h)) > 0$

Since the subject of the study is the chosen modulation, henceforth m and $\hat{m}$ are used instead of the signal: $\hat{m}$ stands for the modulation chosen according to the prediction delivered by the OS and m stands for the modulation chosen according to perfect prediction. Figure 7.5–a shows the percentage of cases when $\hat{m} > m$ and Figure 7.5–b the percentage of cases when $\hat{m} < m$.

First of all, it is striking that the fraction $\hat{m} < m$ is very similar to the fraction $\hat{m} > m$ (the difference is in the order of 10^{-7}). However, this is intuitive, as it means that it is as likely that $s(i-h)$ is on the opposite side of the threshold as $s(i+h)$.

The rate of wrong modulation choices increases with the prediction horizon: for $d = 1$ ms it lies around 5%, but for $d = 4$ ms it is already 15%. The largest amount of wrong modulation decisions occurs when $\mathrm{SNR}_{\mathrm{avg}} = \mathrm{SNR}_{\mathrm{th}}$ and

Figure 7.5: Rate of modulation choices according to the OS prediction that differ from the modulation which would have been chosen if the prediction were accurate. The x-axis shows the difference between the average SNR of the signal and the modulation changing threshold.

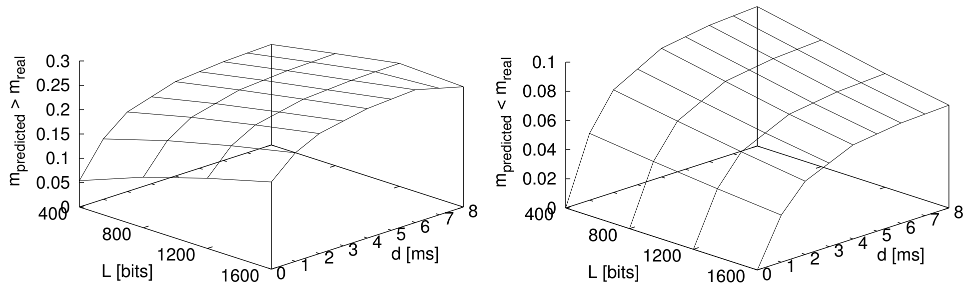


(a) $\hat{m} > m$ (b) $\hat{m} < m$

decreases as the difference between $\mathrm{SNR}_{\mathrm{avg}}$ and the threshold increases. For a difference of 10 dB between the average SNR and the threshold, the percentage of wrong modulation choices is insignificant for all prediction horizons. For this reason, the adaptive rate scheme should behave conservatively and choose a lower rate than predicted only if the average SNR is close to the threshold.

7.4.2 Causes of Packet Errors

Figure 7.6: Percentage of erroneous choices of transmission rate.



(a) $\hat{m} > m$ (b) $\hat{m} < m$

Knowing the probability of choosing the wrong modulation, it is now necessary to study how often that leads to packet errors at the receiver. The same adaptive rate scheme as in Section 7.3 is used with packet lengths $L \in \{400, 800, 1200\}$. Varying the feedback delay of the link quality samples d, the following amounts counted:

- the number of packets with errors,
- the number of times that the rate chosen is too high,
- the number of times that the rate chosen is too low,
- the number of times that one of the previous leads to packet errors.

Figure 7.6 shows that too high a modulation is chosen between 6 and 25% of the times, depending on the packet size. For short packets the variation with the feedback delay is higher than for long packets because of the shorter required prediction horizon.

It happens less often that too low a modulation is chosen. This is because the modulation changing thresholds are not symmetrically arranged around $\mathrm{SNR}_{\mathrm{avg}} = 20$ dB (see Table 7.1). Since most actual SNR values lie in the interval [18;24] dB, most predicted values $\hat{s}$ are also in that interval (OS prediction:$\widehat{s(i)}_h = s(i)$). Most of the wrong modulation choices occur

Figure 7.7: Percentage of erroneous modulation choices for an average SNR of 20 dB and thresholds at differences of 2 and 4 dB.

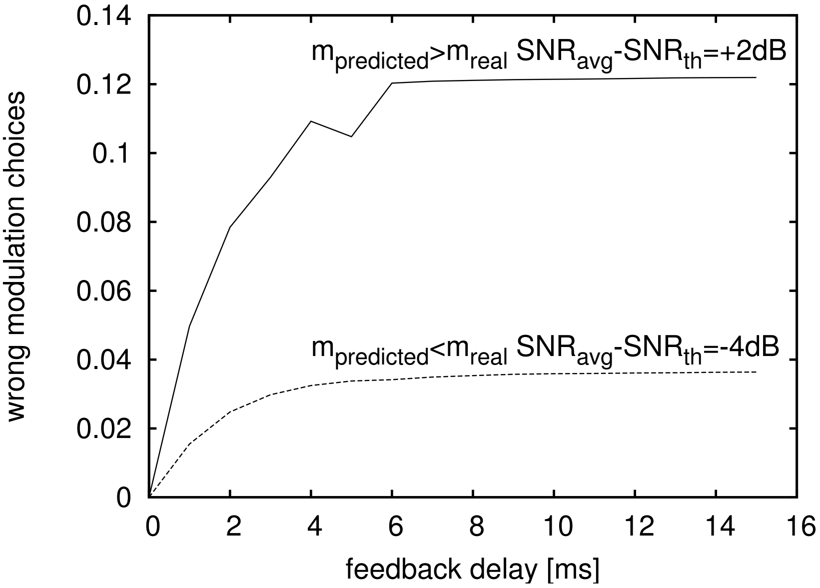
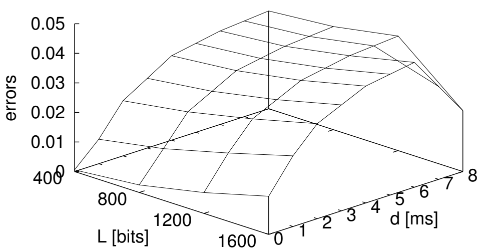
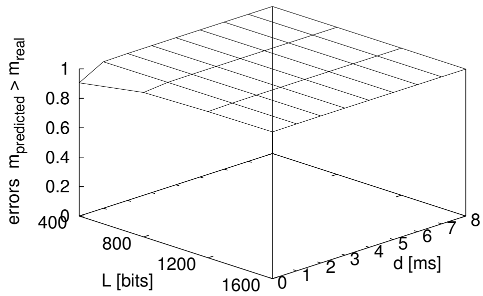


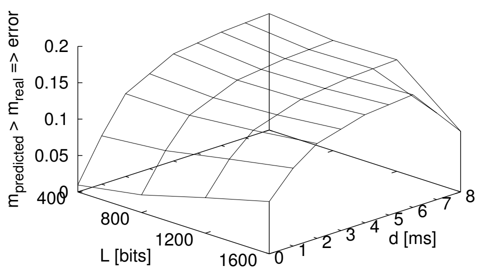
Figure 7.8: Percentage of packets with errors, percentage of packets with errors which were transmitted with too high a data rate, percentage of too high a rate choices that lead to packet errors.



(a) packet errors



(b) $\hat{m} > m$ | packet errors



(c) packet errors | $\hat{m} > m$

for the rate thresholds closest to the $\mathrm{SNR}_{\mathrm{avg}}$—a consequence of the results in 7.4.1—which are at 18 and 24 dB. Figure 7.7 shows the results obtained in the previous section for the corresponding difference between $\mathrm{SNR}_{\mathrm{avg}}$ and the modulation thresholds. There is a higher percentage of wrong choices made relative to the lower threshold than to the upper because the SNR difference is lower. If $\hat{m}$ is in the interval [18;24], a wrong modulation choice relative to the lower threshold means that $\hat{m} > m$ and a wrong choice relative to the higher threshold means that $\hat{m} < m$. According to Figure 7.7 the first case happens more often than the second, which is why there are more cases of the chosen modulation being too high than cases of being too low.

The adaptive rate mechanism chooses the modulation based on the lowest signal value for the whole packet duration (Section 5.2.1). The longer the packet, the higher the probability is that there is at least one actual signal value below the rate threshold. This leads to the choice of the modulation in case of perfect prediction. But for OS prediction, the amount of predicted $\hat{s}$ above the threshold is independent of the packet size. Consequently, m is lower for longer packets, whereas $\hat{m}$ is independent of the packet size because it depends only on the latest available signal value. Thus, for increasing packet length, the percentage of times that too low a modulation is chosen decreases (Figure 7.6–b).

Figure 7.8 shows the percentage of packets received with errors, the percentage thereof which were sent with too high a rate ($\hat{m} > m$), and the percentage of packets sent with too high a rate that actually lead to packets errors. In Figure 7.8—a, the percentage of packet errors grows with the packet length because farther prediction horizons are used in the choice of the rate $\hat{m}$, and that leads to choosing the wrong rate more often, as seen above. Figure 7.8–b shows that packet errors are almost always due to transmission with too high a rate. Since the main goal of the changes to the adaptive modulation scheme is to reduce packet losses it is necessary to reduce the amount of times when too high a modulation is chosen. But, although transmission at too low a rate does not lead to packet errors, it does lead to less efficient channel utilisation. Figure 7.8–c shows that only a percentage lower than 18% of the packets sent at too high a rate lead to packet errors. According to these results, always being conservative and transmitting at a rate lower than that corresponding to the prediction of the OS would lead to almost no packet errors. However, in at least 82% of the cases it would be unnecessary and would cause a low channel utilisation. Thus, it is important to determine a rule to decide which packets require a lower modulation than that predicted by the OS predictor to be used. To be able to do so, it is necessary to know which of the too high a modulation choices $\hat{m} > m$ actually lead to packet errors. It seems likely that the difference between the predicted signal value $\hat{s}$ and the modulation changing threshold s_{th} would be relevant in that distinction. So, the values of the difference $\hat{s} - s_{\mathrm{th}}$, where s_{th} is the first modulation changing threshold lower than the signal value, were gathered for all packets for which $\hat{m} > m$. Figure 7.9 shows the 95-percentile of the distribution of that difference. Unfortunately, according to the results, packets sent with too high a modulation that suffer errors are as close to packets sent with too high a rate that do not. So, it is not possible to distinguish between them beforehand to choose which packets to be conservative with when making the modulation choice.

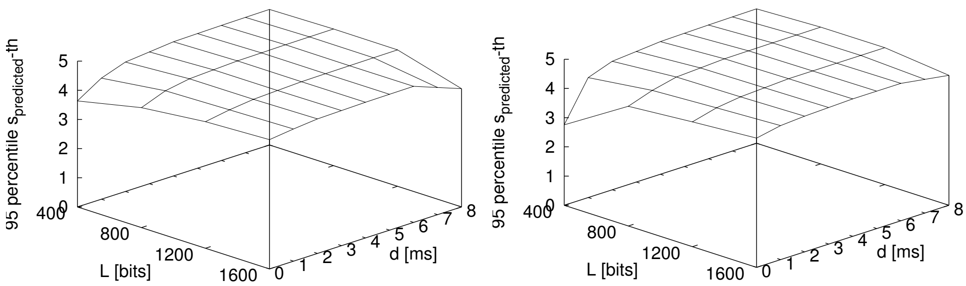
7.4.3 A Stochastic Threshold-based Rate Adaptation Scheme

As it is not possible to differentiate among packets that are sent with too high a modulation, the proposed stochastic adaptive modulation scheme tries to reduce the number of packets sent with too high a modulation blindly. The stochastic adaptive rate scheme works as follows:

- the predicted modulation $\hat{m}$ is chosen corresponding to the SNR calculated from the predicted signal $\hat{s}$ by the OS predictor $\hat{\mathrm{SNR}}[\mathrm{dB}] = 20\log(\bar{s}) - 10\log(\mathrm{NPwr}) + 20\log(\hat{s}))$;
- the packet is sent with rate $\hat{m}$ with probability $1 - P(\hat{m} > m)$;
- or with rate $\hat{m} - 1$ with probability $P(\hat{m} > m)$.

The value of $P(\hat{m} > m)$ is taken from the results obtained in Section 7.4.1, depending on the difference between the average SNR $\hat{\mathrm{SNR}}_{\mathrm{avg}} = 20\log(\bar{s}) - 10\log(NPwr)$ and the modulation changing threshold. For the feedback delay of the signal value d, the duration of the packet is used. The average signal $\bar{s}$ is calculated as an exponentially weighted low-pass filter of the recently measured

Figure 7.9: 95-percentile of $\hat{s} - s_{\mathrm{th}}$ for all packets with $\hat{m} > m$ and for those where $\hat{m} > m$ leads to packet errors.



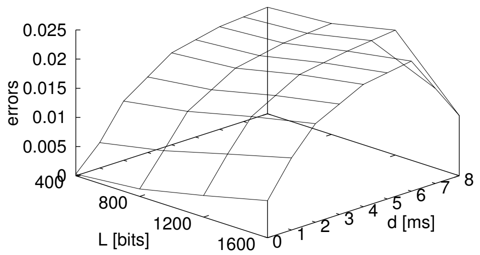
(a) $\hat{m} > m$ (b) $\hat{m} > m \wedge$ packet errors signal values:

$$\bar{s}_i = (1-\alpha)\bar{s}_{i-1} + \alpha s_{i-1}, \tag{7.11}$$

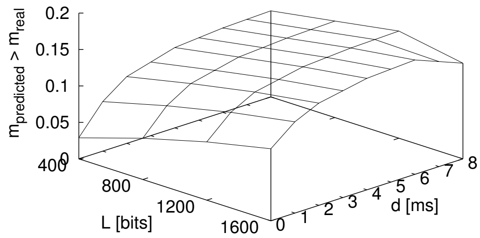
where the time constant of the low-pass filter α is a tuning parameter.

The total PLR obtained using this rate adaptation scheme is plotted in Figure 7.10 together with the rate of packets which are transmitted with too high a modulation and the percentage of too high a modulation choices that lead to channel errors. Using the proposed scheme, the number of packets sent with too high a modulation is reduced by 10% compared to the results in Figure 7.7–a. The number of these that result in packet errors is now 2% lower than before (compared with Figure 7.8–c). This results in a packet error rate that is halved but still 2% where the target PER is 10^{-3}. Variations of the value of the parameter $\alpha \in [0.6; 0.9]$ showed no impact on the results.

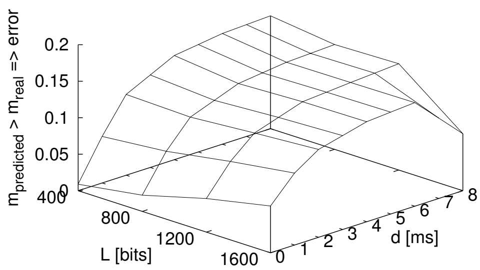
Figure 7.10: Percentage of packets with errors, percentage of packets transmitted with too high rate and percentage of too high rate choices that lead to packet errors for the stochastic adaptative MCS.



(a) packet errors



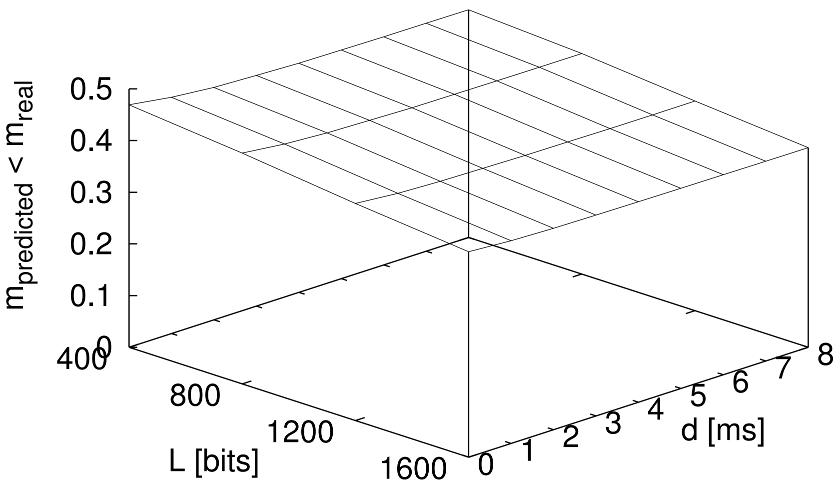
(b) $\hat{m} > m$



(c) packet errors | $\hat{m} > m$

The proposed change is not enough to mitigate the effects of the prediction errors made by the OS predictor. Furthermore, the side-effect of the proposed scheme is an increased number of packets sent at too low a modulation, as can be seen in Figure 7.11. This means that the channel is often inefficiently used.

Figure 7.11: Percentage of packets transmitted with too low rate ($\hat{m} < m$) for the stochastic adaptive MCS.



7.5 Conclusions

This chapter initially compares the accuracy of the OS heuristic for link prediction with a moving average and linear prediction heuristics and the MC algorithm, which is based on an AR model of the wireless link. Based on the NMSE of the prediction errors, the OS offers the best prediction for low mobility WLAN channels up to 2 ms. Beyond 2 ms, the best prediction is achieved by a moving average but the OS is the second best within 1.5 dB of the lowest NMSE for horizons up to 15 ms. A study of the performance of adaptive MCS with packetised data confirms those results. Because the OS is also the simplest heuristic, the rest of the chapter studies the details if how its inaccuracy influences the rate adaptation mechanism described in Section 5.2.1.

The results show that errors in link prediction at the transmitter cause packet transmission errors because too high a data rate is used: up to 20% of the decisions lead to choosing too high a MCS and up to 20% of those lead to packet errors. The amount of times that a wrong decision is made depends on how close the average SNR of the link is to the threshold used for changing the transmisson rate. Choosing a too low data rate for transmission leads to capacity losses, but it is not as harmful to application performance as packet losses.

A simple stochastic rate adaptation scheme that tries to reduce the amount of packet transmission errors by reducing the amount of packet transmitted with too high rate is proposed and evaluated. Although the packet error rate is reduced by 50%, this comes at the cost of an increase of at least 20% in the amount of packets that are sent at too low a modulation and inefficiently use the channel.

Chapter 8

Wireless Scheduling with Inaccurate Link Knowledge

Chapter 6 showed the functionality of the PeLe scheduler. It achieves higher user perceived quality than the reference schedulers, assuming that the AP has perfect information about the behaviour of the wireless links. Chapter 7 showed that assuming the link stays constant is the best low complexity prediction method, but it is not perfect. As a consequence, packet errors occur more often than desired because of wrong modulation choices. This chapter studies how the inaccuracy in prediction of link behaviour affects the performance of the PeLe and the reference schedulers.

8.1 Scenario and Metrics

The scenario used consists of a cell with a bandwidth of W=250 kHz, 4 VoIP clients, 4 Bulk clients and 12 WWW clients (18 WT). Instead of assuming perfect knowledge of link behaviour, as was the case in Chapter 6, the modulation is adapted according to the received signal strength measured at the time of the scheduling decision. This assumes no feedback delay for the transmission of link quality values from the WT to the AP.

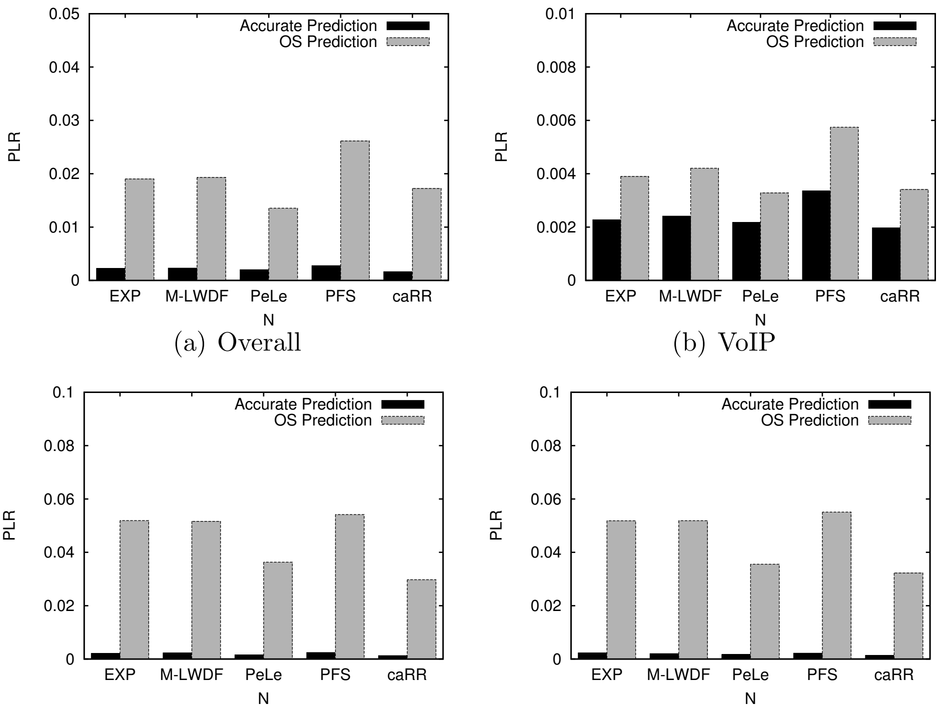
The evaluation in the next section compares the results of Chapter 6 with the perceived quality of the applications when the choice of transmission rate is made by a OS predictor (Section 7.1.2). Special attention is devoted to the Packet Loss Rates (PLR) on the wireless links, which are calculated according to Eq. 7.10. Both the overall PLR on the channel and the PLR per application type are measured, since the PLR depends on traffic type as a consequence of the different packet lengths.

8.2 Effects of Prediction Errors in Scheduler Performance

Figure 8.1 shows the PLR on the wireless channel when the link behaviour is perfectly known before transmission and when it is predicted using the OS method. In general, the PLR increases by almost one order of magnitude when using the OS instead of perfect prediciton, whereby not all applications are affected in the same way. For VoIP flows the PLR more or less doubles because the short packets require only short prediction horizons and are, therefore, subject to fewer and smaller prediction errors according to the results in Section 7.3. Bulk and WWW traffic suffer more from the inaccuracy in link prediction because of the large packets, which require a farther prediction horizon. The link prediction errors lead to an increase of an order of magnitude in the PLR for those flow types.

Comparing now the influence of the link prediction errors on the scheduler behaviour, Figure 8.1 shows that the PFS suffers the largest increase in packet

Figure 8.1: Comparison of the PLR for all studied schedulers, when the prediction of link behaviour is perfect versus when it is obtained by the OS prediction heuristic. (The ranges of the y-axis are different.)



(a) Overall (b) VoIP

(c) WWW (d) Bulk

errors followed by the EXP and M-LWDF. This is because the scheduling decision depends strongly on current and past link prediction for the choice of the transmission rate . These schedulers consider not only the data rate for the current transmission but also the data rates used in previous transmissions. Previous results have shown that packet errors occur mainly because too high a modulation is chosen. Since these schedulers choose the flow that can transmit at the highest relative rate, they are more prone to choose flows that overestimate link quality, and thus are more sensitive to prediction errors. The caRR also considers link condition in its scheduling decision, but only to decide whether the link is good enough to meet a certain target PER. This makes it less sensitive to prediction errors than opportunistic schedulers.

The PeLe scheduler takes the link quality into consideration in the scheduling decision but in an indirect way (as was explained in Chapter 3.4.2): the possible modulation is used to calculate the duration of the packet in the channel and the estimated cost in terms of quality for the flows that do not transmit. As the scheduling decision is not directly coupled to the transmission rate, it is less sensitive to link overestimation, and prediction errors lead less often to transmission errors.

Figure 8.2: Comparison of the transmission rate used on average by the studied schedulers when the prediction of link behaviour is perfect versus when it is obtained by the OS prediction heuristic.

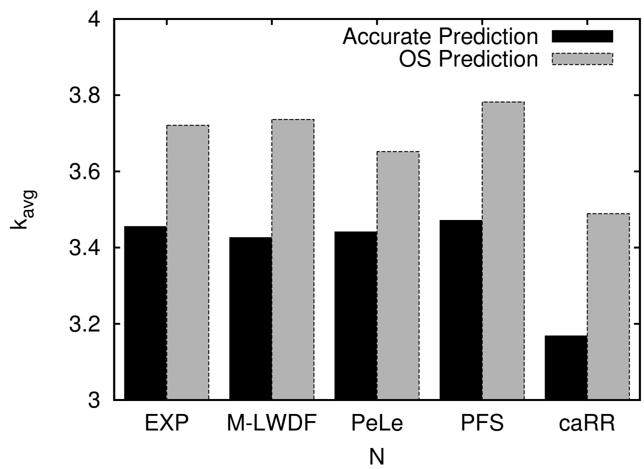


Figure 8.2 shows the average modulation depth k used by each scheduler with perfect and OS prediction. For all schedulers a higher average transmission rate is used with the OS prediction than with perfect knowledge of the link behaviour. The lowest difference occurs for the PeLe scheduler for the reason stated above. At the other extreme, the caRR has the largest difference because, on one hand, positive prediction errors do not influence the schedul-

ing decision (in which case the used rate is the same or higher as for perfect prediction); on the other hand, negative prediction errors only influence the decision if the actual SNR is very low, in which case the negative prediction error leads to postponing transmission. Because prediction errors of both signs are equally probable, as was seen in Section 7.3, and negative errors do not lead to transmissions, the caRR is more strongly biased by positive prediction errors than the other schedulers. For the PFS, the EXP and the M-LWDF, negative prediction errors mostly lead the scheduler to choose another flow, which has either positive or no error. So, opportunistic or channel-aware schedulers are more sensitive to positive prediction errors, i. e. to the cases when $\hat{m} > m$.

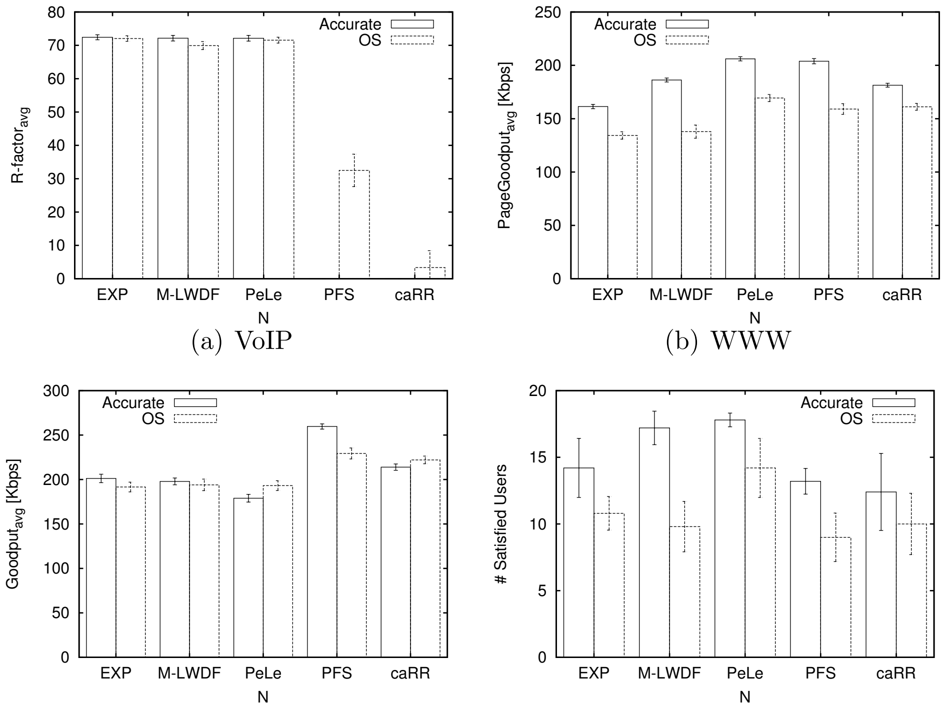
Figure 8.3 shows the effects of using OS link prediction on the user perceived quality (a–c) and on the amount of satisfied users (d). As expected from the results above, the VoIP quality does not change significantly for the schedulers that deliver satisfactory service (R-factor>70). Also, from the results above, it is expected that both TCP-based applications receive lower goodput because of more packet errors. This leads to lower received service and lower quality of both applications, as can be seen in Figure 8.3-b and c.

The difference in average quality is larger for WWW applications because the utility curve is stricter[1]. As a consequence, the number of satisfied users also decreases. Thereby, the PeLe still provides the largest amount of satisfied users since it less sensitive to prediction errors than the EXP and M-LWDF. In general, the PeLe and caRR are the least sensitive to prediction errors, but the PeLe provides satisfying quality to more users, some of them being VoIP clients.

As seen above, link prediction errors lead to increased PLR depending on the necessary prediction horizon, which is determined by the packet length and by the channel bandwidth. Thus, if the channel bandwidth W is increased, the necessary prediction horizons decrease proportionally. For $W = 400$ kHz a packet of 1500 Bytes occupies the channel for 15 ms and VoIP packets for 1.18 ms; but for $W = 2$ Mbps the first occupy only 3 ms and the latter a few hundred microseconds. As a consequence, for larger bandwidths the effects of prediction inaccuracies are expected to be smaller because of the shorter prediction horizons needed. However, this study did not consider the delay necessary to feedback the link quality measurements from the receiver to the transmitter and which can be seen as a packet length-independent, constant increase in the necessary prediction horizon.

[1] See Section 6.5 for the calculation of the satisfied users.

Figure 8.3: Comparison of the average perceived quality of each flow type for all studied schedulers when the prediction of link behaviour is perfect versus when it is obtained by the OS prediction heuristic.



(a) VoIP (b) WWW

(c) Bulk (d) # satisfied users

8.3 Conclusions

This chapter describes a study of the effects of inaccuracies in the prediction of link quality when the OS prediction method described in Section 7.1.2 is used. The performances of the schedulers studied in this thesis are compared when the transmission rate adaptation assumes perfect knowledge of the link quality and when it is based on inaccurate prediction by the OS predictor.

The results show increased packet error rates, some flow types being more affected than others because of the different packet length. VoIP traffic does not suffer an increase in the PLR because of the short packets, which need only very short prediction horizons. However, the goodput seen by TCP-based applications is much impaired due to the larger packets which require far prediction horizons. Although VoD traffic is not studied in the scenario simulated, it is expected that it suffers a lot from prediction errors because the very important I-frames are transmitted in large packets and, similarly to TCP traffic, see an increased packet error rate.

As for the schedulers, the stronger the ranking metric depends on the link quality, the larger the influence of the link prediction errors in the performance. The caRR and PeLe schedulers show the lowest decrease in number of satisfied users, but the PeLe still provides good quality to the most users, including to VoIP flows.

Chapter 9

Conclusions

This thesis proposes a novel approach to packet scheduling over a time shared wireless chanel. The PeLe scheduler takes advantage of the fact that the user perceived quality of an application has a soft dependence on the network service delivered. Moreover, that sensitivity to variations in the received network service is application dependent. A utility curve is used to expresse the relationship between the user perceived quality and the network service received by each application. The utility curve and the recent history of service delivered to a flow are used to calculate the relative importance of packets from different flows. Packets competing for the channel are ranked according to a metric that weighs the benefit of allocating the channel to a flow against the damage that this allocation causes to the other flows. The packet of highest rank is then transmitted, i. e. the scheduler chooses the packet that maximises the sum of the user perceived quality of all flows after its transmission. The link quality that a flow experiences is reflected by the duration of transmission of its packets, depending on the usable transmission rate; a low link quality requires the use of a robust modulation and coding scheme, which has a higher cost in terms of channel occupation time.

The performance of the PeLe scheduler proposed is thoroughly evaluated. The selection of possible utility curves for example applications highlighted relevant issues, namely how the utility curve may express different sensitivities of the application to network service. Furthermore, it was shown that it is advantageous to take application characteristic traffic patterns into account. In summary, this thesis demonstrated how application specific support, including a utility curve, can be designed for an application whose traffic pattern and sensitivity to network service are known.

Using simulations and realistic traffic models, the proposed PeLe scheduler was compared to state-of-the-art schedulers in terms of application perceived quality and of the total channel throughput in many load scenarios and with varied traffic mixes. The results show a difference of at most 12% in overall cell throughput between any two compared schedulers under realistic traffic conditions. Nevertheless, within this small throughput range, there is a stonger differentiation in terms of user perceived quality. The PFS, which always achieves the highest channel throughput, is not able to support audiovisual applications since it does not enable differentiated application support and cannot cope with the different service requirements. Furthermore, the caRR neither impresses with high throughput nor supports flow-specific requirements.

The proposed PeLe and the EXP (an extension of the PFS that enables some support of delay constraints) both deliver differentiated QoS support to the four traffic types studied (bulk transfer, web traffic, VoIP and video streaming). Although the difference in overall cell throughput between the two does not exceed 3%, the two schedulers have different goals and lead to different performance in terms of user perceived quality. The EXP handles audiovisual flows strictly preferentially even if it means that the quality of other flows is strongly degraded. The PeLe scheduler distributes the available bandwidth according to the improvement in user perceived quality as expressed by the utility curves, thus achieving a larger number of satisfied users accross all applications in a wide range of traffic and load scenarios.

Because the schedulers studied assume the existence of a prediction of the wireless link behaviour at the transmitter, the effects of that assumption were studied. The accuracy of the One Step heuristic for prediction of the received signal strength was compared with a moving average and linear prediction heuristics and the MC algorithm. According to the results, the OS delivers the best prediction up to 2 ms for low mobility WLAN channels; although beyond 2 ms the best prediction was achieved by a moving average, the OS was the second best for horizons up to 15 ms. The prediction errors cause packet transmission errors because too high a data rate is used, increasing the packet loss rate by up to 2 orders of magnitude. The amount of times that a wrong modulation is chosen depends on how close the average SNR of the link is to the threshold used for changing the modulation. A simple stochastic rate adaptation scheme that tries to reduce the amount of packet transmission errors by reducing the amount of packet transmitted with too high a rate was proposed and evaluated. Although the packet error rate is reduced by 50%, this comes at the cost of an increase of at least 20% in the amount of packets that are sent at too low a modulation and that therefore inefficiently use the channel.

Finally, the performance of PeLe and the reference schedulers was compared when the transmission rate adaptation is done based on perfect knowledge of the link quality and when it is based on inaccurate prediction by the OS predictor. The results show increased packet error rates, with some flow types being more affected than others because of the different packet lengths. The packet loss rate of VoIP traffic does not significantly change because the packets are short and need only very short prediction horizons. However, TCP based applications suffer a large increase in the amount of packets with transmission errors because the larger packets require farther prediction horizons. Regarding the scheduler performance, when the ranking metric depends on the link quality, the influence of the link prediction errors in the scheduler performance is large. As a consequence, the PeLe scheduler proposed in this thesis showed the lowest decrease in amount of satisfied users. Actually, the PeLe provides good quality to most users from all application types, even when OS prediction is used.

In summary, the results obtained in this work make it possible to better understand the features of individual schedulers. The newly developed PeLe scheduler improves the user perceived performance of wireless communications under realistic traffic and system conditions.

Bibliography

[1] ITU-T, "Recommendation g.113 appendix 1: Provisional planning values for the equipment impariment factor ie," Oct 2001.

[2] A. Reyes-Lecuona, E. González-Parada, E. Casilari, and A. Díaz-Estrella, "A page-oriented www traffic model for wireless system simulations," in *Proc of the 16th International Teletraffic Congress—ITC'16*, pp. 1271–1280, Jun 1999.

[3] A. Aguiar and J. Gross, "Wireless channel models," Tech. Rep. TKN-03-007, Telecommunication Networks Group, Technical University Berlin, Apr 2003.

[4] W. Lee, *Mobile Cellular Telecommunications*. McGraw-Hill International Editions, 1995.

[5] J. K. Cavers, *Mobile Channel Characteristics*. Kluver Academic Publishers, 2000.

[6] A. Neskovic, N. Neskovic, and G. Paunovic, "Modern approaches in modeling of mobile radio systems propagation environment," *IEEE Communications Surveys and Tutorials*, vol. 3, pp. 51–82, Sep 2000.

[7] S. Seidel, T. Rappaport, and R. Singh, "Path loss and multipath delay statistics in four european cities for 900 mhz cellular and microcellular communications," in *Electronics Letters*, vol. 26, pp. 1713–1715, IEEE, Sep 1990.

[8] S. Seidel and T. Rappaport, "900 mhz path loss measurements and prediction techniques for in-building communication system design," in *Proc. of the 41st IEEE Vehicular Technology Conference*, pp. 613–618, 1991.

[9] X. Zhao, J. Kivinen, P. Vainikainen, and K. Skog, "Propagation characteristics for wideband outdoor mobile communnications at 5.3 ghz," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 507–514, Apr 2002.

[10] R. Steele and L. Hanzo, eds., *Mobile Radio Communications.* J. Wiley & Sons Ltd, 2000.

[11] J. Zander and S.-L. Kim, *Radio Resource Managements for Wireless Networks.* Mobile Communications Series, Artech House Publishers, 2001. page 330.

[12] S. Yang, *CDMA RF System Engineering.* Mobile Communications Series, Artech House Publishers, 1998. page 19.

[13] Y. Oda, R. Tsuchihashi, K. Tsunekawa, and M. Hata, "Measured path loss and multipath propagation characteristics in uhf and microwave frequency bands for urban mobile communications," in *Proc. of the 53rd IEEE Vehicular Technology Conference* (IEEE, ed.), vol. 1, pp. 337–341, 2001.

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

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

[16] W. Jakes, *Microwave Mobile Communications.* IEEE Press, Wiley Interscience, 1994.

[17] H. Brehm, H. Stammler, and W. Werner, "Design of a highly flexible digital simulator for narrowband fading channels," in *Signal Processing III: Theory and Applications* (I. Y. et al., ed.), pp. 1113–1116, Elsevier Science Publishers, 1986.

[18] M. Pätzold, R. Garcia, and F. Laue, "Design of high-speed simulation models for mobile fading channels by using table look-up technique," *IEEE Transactions on Vehicular Technology*, vol. 49, pp. 1178–1190, Jul 2000.

[19] J. G. Proakis, *Digital Communications.* McGraw-Hill International, 4 ed., 2001.

[20] A. Kamerman and L. Monteban, "Wavelan-ii: A high-performance wireless lan for the unlicensed band," *Bell Labs Technical Journal*, pp. 118–133, Summer 1997.

[21] D. Eckhardt and P. Steenkiste, "A trace-based evaluation of adaptive error correction for a wireless local area network," *Journal on Special Topics in Mobile Networking and Applications*, vol. 4, pp. 273–287, Dec 1999.

[22] S. Vishwanath and A. Goldsmith, "Adaptive turbo coded modulation for flat fading channels," *IEEE Transactions on Communications*, vol. 51, pp. 964–972, jun 2003.

[23] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *Proc. of the 7th An. Int. Conf. on Mobile Computing and Networking—Mobicom*, pp. 236–251, Jul 2001.

[24] A. Goldsmith and S.-G. Chua, "Variable-rate variable-power mqam for fading channels," *IEEE Transactions on Communications*, vol. 45, pp. 1218–1230, Oct 1997.

[25] S. Nanda, K. Balachandran, and S. Kumar, "Adaptation techniques in wireless packet data services," *IEEE Communications Magazine*, pp. 54–64, Jan 2000.

[26] "Higher speed physical layer (phy) extension in the 2.4 ghz band." Supplement to 802.11-1999,Wireless LAN MAC and PHY specifications, Oct 1999.

[27] "Specific requirements—part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications—amendment 4: Further higher-speed physical layer extension in the 2.4 ghz band." ISO/IEC 8802-11, Jun 2003.

[28] "Ieee standards for information technology—telecommunications and information exchange between systems—local and metropolitan area network—specific requirements—part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications." Standard ISO/IEC 8802-11: 1999, Oct 1999.

[29] B. R. Badrinath and P. Sudame, "To send or not to send: Implementing deferred transmissions in mobile hosts," in *16th International Conference on Distributed Computing Systems (ICDCS)*, pp. 327–333, May 1996.

[30] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi, "Using channel state dependent packet scheduling to improve tcp throughput over wireless lans," *Wireless Networks Journal, vol. (3)1*, 1997.

[31] J. Postel, "Transmission control protocol." RFC 793, Sep 1981.

[32] S. Shenker, "Fundamental design issues for the future internet," *IEEE Journal of Selected Areas in Communications*, vol. 13, pp. 1176–1188, Sep 1995.

[33] M. El-Gendy, A. Bose, and K. Shin, "Evolution of the internet qos and support for soft real-time applications," *Proceedings of the IEEE*, vol. 91, pp. 1086–1104, Jul 2003.

[34] A. Bouch, M. Sasse, and H. DeMeer, "Of packets and people: A user-centred approach to quality of service," in *Proc. Int. Workshop on Quality of Service*, pp. 189–197, Jun 2000.

[35] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications." RFC 3550, Jul 2003.

[36] H. Schulzrinne, A. Rao, and R. Lanphier, "Rtsp: A transport protocol for real-time applications." RFC 2326, Jan 1998.

[37] "Cooperative association for internet data analysis." URL, 2001–02.

[38] M. Allman, V. Paxson, and W. Stevens, "Tcp congestion control." RFC 2581, April 1999.

[39] A. Kumar, "Comparative performance analysis of versions of tcp in a local network with a lossy link," *IEEE/ACM Trans. on Networking*, 1998.

[40] H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, "Tcp over second (2.5g) and third (3g) generation wireless networks." RFC 3481, Feb 2003.

[41] G. Xylomenos and G. Polyzos, "Tcp and udp performance over a wireless lan," in *Proc. Infocom'99*, 1999.

[42] N. Bhatti, A. Bouch, and A. Kuchinsky, "Integrating user-perceived quality into web server design," *Computer Networks*, vol. 33, Jun 2000.

[43] W. Noureddine and F. Tobagi, "Improving the performance of interactive tcp applications using service differentiation," *Computer Networks*, vol. 40, pp. 19–43, Sep 2002.

[44] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving tcp performance over wireless links," *IEEE/ACM Trans. on Networking*, vol. 5, pp. 756–768, Dec 1997.

[45] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving tci/ip performance over wireless networks," *ACM Wireless Networks*, vol. 1, Dec 1995.

[46] C. Parsa and J. J. Garcia-Luna-Aceves, "Tulip: A link-level protocol for improving tcp over wireless l inks," in *Proc. Wireless Communications and Networking Conference—WCNC'99*, vol. 3, 1999.

[47] M. Matis and M. Allman, "A framework for defining empirical bulk transfer capacity metrics." RFC 3148, Jul 2001.

[48] M. Allman, "Measuring end-to-end bulk transfer capacity," in *Proc. 1st ACM SIGCOMM Workshop on Internet Measurements*, pp. 139–143, 2001.

[49] J. Postel, "User datagram protocol." RFC 768, Aug 1980.

[50] Ç. Aras, J. Kurose, D. Reeves, and H. Schulzrinne, "Real-time communication in packet switched networks," *Proceedings of the IEEE*, vol. 82, Jan 1994.

[51] H. Sanneck, W. Mohr, L. Le, C. Hoene, and A. Wolisz, *Wireless IP and Building the Mobile Internet*, ch. 10. Artech House, 1st ed., 2002.

[52] R. Steinmetz and K. Nehrstedt, *Multimedia Systems.* Springer, 2004.

[53] C. Sreenan, J.-C. Chen, P. Agrawal, and B. Narendran, "Delay reduction techniques for playout buffering," *IEEE Trans. on Multimedia*, vol. 2, pp. 100–112, Jun 2000.

[54] S. Moon, J. Kurose, P. Skelly, and D. Towsley, "Packet audio playout delay adjustmenst: performance bounds and algorithms," *ACM/Springer Multimedia Systems*, Jan 1998.

[55] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetised audio applications in wide-area networks," *Proc. Infocom*, pp. 680–688, 1994.

[56] MPEG, "Mpeg-4." ISO/IEC Standard, Oct 1998.

[57] "H.264 : Advanced video coding for generic audiovisual services." ITU-T Recommendation, Mar 2005.

[58] D. Petr, J. D. Silva, and V. Frost, "Priority discarding of speech in integrated packet networks," *Journal of Selected Areas in Communications*, vol. 7, pp. 644–656, May 1989.

[59] J. D. Martin, "Source-driven packet marking for speech transmission over differentiated services networks," in *Proc. IEEE Int. Conf. on Audio, Speech and Signal Processing—ICASSP*, pp. 753–756, May 2001.

[60] H. Sanneck, N. Le, and A. Wolisz, "Intra-flow loss recovery and control for voip," in *Proc. 9th ACM Int. Conf. on Multimedia*, pp. 441–454, 2001.

[61] J.-R. Li, X. Gao, L. Qian, and V. Bharghavan, "Goodput control for heterogeneous data streams," in *Proc. of the 10th Int. Workshop on Network and Operating System Support for Digital Audio and Video—IWNOSSDAV*, Jun 2000.

[62] J. Klaue, J. Gross, H. Karl, and A. Wolisz, "Cross-layer optimisation of ofdm transmission systems for mpeg-4 video streaming," *Computer Communications*, vol. 27, pp. 1044–1055, Sept. 2004.

[63] Q. Li and M. van der Schaar, "Providing adaptive qos to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. on Multimedia*, vol. 6, pp. 278–290, Apr 2004.

[64] C. Hoene, *Internet Telephony over Wireless Links.* PhD thesis, Berlin University of Technology, Dec 2005.

[65] I. Aad and C. Castelluccia, "Differentiation mechanisms for ieee 802.11," in *Proc. Infocom'00*, pp. 209–218, 2000.

[66] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Trans. on Multimedia*, vol. 8, pp. 390–404, Apr 2006.

[67] M. Manousos, S. Apostolacos, I. Grammatikakis, and D. Mexis, "Voice-quality monitoring and control for voip," *IEEE Internet Computing*, vol. 9, pp. 35–42, Jul 2005.

[68] ITU-T, "Methods for subjective determination of transmission quality." Recommendation P.800, Aug 1996.

[69] ITU-T, "Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs." Recommendation P.862, Feb 2001.

[70] "Subjective video quality assessment methods for multimedia applications." ITU-T Recommendation P.910, Aug 1996.

[71] C. hoene, H. Karl, and A. Wolisz, "A perceptual quality model for adaptive voip applications," in *Proc. of Int. Symp. on Performance Evaluation of Computer and Telecommunications Systems*, 2004.

[72] "The e-model, a computational model for transmission planning." ITU-T Recommendation G.107, Mar 2005.

[73] C. Hoene, S. Wiethölter, and A. Wolisz, "Predicting the perceptual service quality using a trace of voip packets," in *Proc. of Fifth International Workshop on Quality of future Internet Services (QofIS'04)*, (Barcelona, Spain), Sept. 2004.

[74] M. Riley and E. Richardson, *Digital Video Communications.* Artech House, 1997.

[75] L. Hanzo, P. Cherrimann, and J. Streit, *Wireless Video Communications.* IEEE Press, 2001.

[76] J.-R. Ohm, *Multimedia Communication Technology.* Springer, 1st ed., 2004.

[77] G. Hauske, "Design and performance of image quality models," in *Proc. 7th International Conference on Image Processing and its Applications*, pp. 321–325, Jul 1999.

[78] Y. Wang, "Survey of objective video quality measurements," tech. rep., EMC Corporation Hopkinton, MA, USA, 2006.

[79] J. W. S. Liu, *Real-Time Systems*, ch. 2. Prentice Hall, 200.

[80] "High speed downlink packet access (hsdpa); overall utran description." 3GPP TS 25.308, Rel-7, RP 35, Mar 2007.

[81] "Universal mobile telecommunications system (umts)—umts phase 1 release 99." 3G TS 22.100 version 3.6.0 Release 1999, 1999.

[82] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Communications*, vol. 12, pp. 3–11, Feb 2005.

[83] D. Reininger and R. lzmailov, "Soft quality-of-service with vbr+video," in *Proc. of 8th Int. Workshop on Packet Video (AVSPN'97)*, Sep 1997.

[84] R.-F. Liao, P. Boukelee, and A. Campbell, "Dynamic generation of bandwidth utility curves for utility-based adaptation," in *Proc. Packet Video '99*, Apr 1999.

[85] A. Demers, "Analysis and simulation of a fair queueing algorithm," *Journal of Internetworking Research and Experience*, Oct 1989.

[86] D. D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: architecture and mechanism," in *Proc. ACM SIGCOMM*, pp. 14–26, Aug 1992.

[87] A.Parekh and R. Gallager, “A generalized processor sharing approach to flow control—the single node case,” *IEEE/ACM Trans. on Networking*, vol. 2, pp. 137–150, Apr 1994.

[88] L. Zhang, “Virtual clock: a new traffic control algorithm for packet switching networks,” *ACM SIGCOMM Computer Communication Review*, vol. 20, pp. 19–29, Sep 1990.

[89] J. Bennett and H. Zhang, “Wf^2q: Worst-case fair weighted fair queueing,” in *Proc. Infocom'96*, IEEE, Mar 1996.

[90] I. Stoica, H. Abdel-Wahab, K. Jeffay, S. Baruah, J. Gehrke, and C. Plaxton, “A proportional share resource allocation algorithm for real-time, time-shared systems,” in *Proc. 17th Real-Time Systems Symposium*, pp. 288–299, Dec 1996.

[91] S. Golestani, “A self-clocked fair queueing scheme for broadband applications,” in *Proc. Infocom '94*, vol. 2, pp. 636–646, IEEE, Jun 1994.

[92] H. Zhang, “Service disciplines for guaranteed performance service in packet-switching networks,” *Proc. of the IEEE*, vol. 83, pp. 389–412, Oct 1995.

[93] R. Ramanathan and P. Agrawal, “Adapting packet fair queueing algorithms to wireless networks,” in *Proc. Mobicom*, Oct 1998.

[94] T. S. E. Ng, I. Stoica, and H. Zhang, “Packet fair queueing algorithms for wireless networks with location-dependent errors,” in *Proceedings of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies — INFOCOM '98*, vol. 3, pp. 1103–1111, Mar 1998.

[95] S. Lu, V. Bharghavan, and R. Srikant, “Fair scheduling in wireless packet networks,” in *Proc. ACM SIGCOMM*, Sep 1997.

[96] S. Lu, T. Nandagopal, and V. Bharghavan, “Design and analysis of an algorithm for fair service in error prone channels,” *ACM/ Baltzer Wireless Networks Journal*, vol. 6, pp. 323–343, Aug 2000.

[97] C. Fragouli, V. Sivaraman, and M. Srivastava, “Controlled multimedia wireless link sharing via enhanced class-based queueing with channel-state-dependent packet scheduling,” in *Proc. Infocom* (IEEE, ed.), Mar 1998.

[98] J. Gomez, A. Campbell, and H. Morikawa, “The havana framework: Supporting application and channel dependent qos in wireless networks,” in *Proc. of the 7th Int. Conf. on Network Protocols—ICNP*, Nov 1999.

[99] G. Bianchi, A. T. Campbell, and R.-F. Liao, "On utility-fair adaptive services in wireless networks," in *Proc. of 5th International Workshop on Quality of Service—IWQoS* (IEEE, ed.), May 1998.

[100] D. Hayes, M. Rumsewicz, and L. Andrew, "Quality of service driven packet scheduling disciplines for real-time applications: Looking beyond fairness," in *Proc. Infocom*, Mar 1999.

[101] D. Eckhardt and P. Steenkiste, "Effort-limited fair (elf) scheduling for wireless networks," in *Proc. of INFOCOM 2000*, Mar 2000.

[102] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. on Information Theory*, vol. 48, pp. 1277–1294, Jun 2002.

[103] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of cdma-hdr a high efficiency-high data rate personal communications wireless system," in *Proc. of the Vehicular Technology Conference Spring 2000* (IEEE, ed.), 2000.

[104] E. F. Chaponniere, P. J. Black, J. M. Black, and D. Tse, "Transmitter directed cdma system using path diversity to equitably maximize throughput." U.S. Patent 6449490, Sep 2002.

[105] J. Holtzmann, "Asymptotic analysis of proportional fair algorithm," in *Proc. of IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications—PIMRC* (IEEE, ed.), vol. 2, pp. F33–F37, Sep 2001.

[106] H. Kushner and P. Whiting, "Convergence of proportional-fair sharign algorithms under general conditions," *IEEE Trans. on Wireless Communications*, vol. 3, pp. 1250–1259, Jul 2004.

[107] J. Yang, Z. Yifan, W. Ying, and Z. Ping, "Average rate updating mechanism in proportional fair scheduler for hdr," in *Proc. IEEE Globecom* (IEEE, ed.), pp. 3464–3466, 2004.

[108] M. Andrews, "Instability of the proportional fair scheduling algorithm for hdr," *IEEE Trans. on Wireless Communications*, vol. 3, pp. 1422–1426, Sep 2004.

[109] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting, "Cdma data qos scheduling on the forward link with variable channel conditions," tech. rep., Bell Laboratories, Lucent Technologies, Apr 2000.

[110] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing qos over a shared wireless link," *IEEE Communications Magazine*, pp. 150–154, Feb 2001.

[111] S. Shakkottai and A. L. Stoylar, "Scheduling algorithms for a mixture of real-time and non-real-time data in hdr," in *Proc. of Int. Teletraffic Conferece 2001*, Sep 2001.

[112] H. Koto, M. Fukushima, S. Nomoto, and F. Takahata, "Scheduling algorithm based on sender bufer backlog for real-time application in mobile packet networks," in *Proc. Wireless Communications and Networking Conference* (IEEE, ed.), vol. 3, pp. 1696–1701, 2005.

[113] M. Hu, J. Zhang, and J. Sadowsky, "Traffic aided opportunistic scheduling for downlink transmissions: Algorithms and performance bounds," in *Proc. Infocom'04*, vol. 3, pp. 1652–1661, 2004.

[114] O.-S. Shin and K. Lee, "Packet scheduling over a shared wireless link for heterogeneous classes of traffic," in *Proc. Int. Conf. on Communications—ICC*, vol. 1, pp. 58–62, Jun 2004.

[115] S. Abedi, "Efficient radio resource management for wireless multimedia communications:a multidimensional qos-based packet scheduler," *IEEE Trans. on Wireless Communications*, vol. 4, pp. 2811–2822, Nov 2005.

[116] S. Abedi, "Improved stability of qos provisioning for 3g systems and beyond: Optimum and automatic strategy selection for packet schedulers," in *Proc. Int. Conf. on Communications — ICC* (IEEE, ed.), Jun 2004.

[117] G. based pricing, resource allocation for multimedia broadband networks Hongbin Ji, J. Hui, and E. Karasan, "Gos-based pricing and resource allocation for multimedia broadband networks," in *Proc. Infocom* (IEEE, ed.), vol. 3, pp. 1020–1027, Mar 1996.

[118] Z. Cao and E. Zegura, "Utility max-min: an application-oriented bandwidth allocation scheme," in *Proc. Infocom*, vol. 2, pp. 793–801, Mar 1999.

[119] F. P. Kelly, "Charging and rate control for elastic traffic," *European Trans. on Communications*, 1997.

[120] T. Harks and T. Poschwatta, "Priority pricing in utility fair networks," in *Proc. of IEEE Int. Conf. on Network Protocols* (IEEE, ed.), IEEE, 2005.

[121] X. Gao, T. Nandagopal, and V. Bharghavan, "On improving the performance of utility-based wireless fair scheduling through a combination of adaptive fec and arq," *Journal of High Speed Networks*, vol. 10, no. 1, pp. 19–36, 2001.

[122] X. Gao, T. Nandagopal, and V. Bharghavan, "Achieving application level fairness through utility-based wireless fair scheduling," in *Proc. Globecom*, vol. 6, pp. 3257–3261, Nov 2001.

[123] Z. Jiang, Y. Ge, and Y. Li, "Max-utility wireless resource management for best-effort traffic," *IEEE Trans. on Wireless Communications*, vol. 4, pp. 100–111, Jan 2005.

[124] X. Liu, E. Chong, and N. Schroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE Journal of Selected Areas in Communications*, vol. 19, pp. 2053–2064, Oct 2001.

[125] R. Liu, R. Berry, and M. Honig, "Delay-sensitive packet scheduling in wireless networks," in *Proc. Wireless Communications and Networking Conference*, vol. 3, pp. 1627–1632, Mar 2003.

[126] M. Alasti, F. Farrokhi, M. Olfat, and K. R. Liu, "Service level agreement /sa) based scheduling algorithm for wireless networks," in *Proc. Int. Conf. on Communications—ICC*, vol. 2, pp. 1028–1032, Jun 2004.

[127] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multiuser throughput allocation subject to throughput constraints," in *Proc. Infocom'05*, vol. 4, pp. 2415–2424, Mar 2005.

[128] K. B. Johnsson and D. C. Cox, "An adaptive cross-layer scheduler for improved qos support of multiclass data services on wireless systems," *IEEE Journal of Selected Areas in Communications*, vol. 23, pp. 334–343, Feb 2005.

[129] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, "Application-driven cross-layer optimization for video streaming over wireless networks," *IEEE Communications Magazine*, pp. 122–129, Jan 2006.

[130] S. Shakkottai, T. Rappaport, and P. Karlsson, "Cross-layer design for wireless networks," *IEEE Communications Magazine*, vol. 41, pp. 74–80, Oct 2003.

[131] A. Aguiar and J. Klaue, "Bi-directional wlan channel measurements in wlan environments with mobility," Tech. Rep. TKN-04-002, Telecommunication Networks Group, Technische Universität Berlin, Apr. 2004.

[132] Intersil Corporation, *PRISM Driver Programmer's Manual (Version 2.10)*, Aug 2001.

[133] J. Malinen, *Host AP driver for Intersil Prism2/2.5/3*. http://hostap.epitest.fi/.

[134] A. Varga, "Omnep++—discrete event simulation system." `www.omnetpp.org`.

[135] E. Casilari, A. Reyes-Lecuona, F. Gonzalez, A. Diaz-Estrella, and F. Sandoval, "Characterisation of web traffic," in *Proc of. IEEE Global Telecommunications Conference—GLOBECOM*, vol. 3, pp. 1862–1866, Nov 2001.

[136] ITU-T, "Recommendation g.729: Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (cs-acelp)," Mar 1996.

[137] "Mpeg-4 part 10: Advanced video coding." ISO/IEC 14496-10 Standard, Oct 2005.

[138] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, no. 6, 2003.

[139] R. Gardner and F. Garcia, "Bulk transfer capacity estimation in ipv6 networks," in *Proc. Int. Multi-Conf. on Computing in the Global Information Technology 2006—ICCGI '06*, 2006.

[140] A. M. Law and W. D. Kelton, *Simulation and Modelling Analysis*. Industrial Enigineering Series, McGraw-Hill International, 3 ed., 2000.

[141] J. Bendat and A. Piersol, *Random Data Analysis and Measurement Procedures*. Wiley Series in Probability and Statistics, 3rd: John Wiley & Sons, 2000.

[142] C. Hoene, "Methodology for derivation of equipment impairment factors from subjective listening-only tests." ITU-T Recommendation P.833, Feb 2001.

[143] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid - a framework for video transmission and quality evaluation," in *Proc. of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, (Urbana, Illinois, USA), Sept. 2003.

[144] "Mpeg-4 ffmpeg codec." `http://ffmpeg.mplayerhq.hu/`.

[145] S. Hu, A. Duel-Hallen, and H. Hallen, "Long-range prediction of fading signals," *IEEE Signal Processing Magazine*, vol. 17, pp. 61–75, May 2000.

[146] T. Eyceoz, S. Hu, A. Duel-Hallen, and H. Hallen, "Adaptive prediction, tracking and power adjustment for frequency non-selective fast fading channels," in *Communication Theory Mini-Conference*, pp. 1–5, 1999.

[147] T. Ekman, M. Sternad, and A. Ahlén, "Unbiased power prediction on broadband channels," in *Proceedings of IEEE Vehicular Technology Conference 2002 — VTC 2002 Fall*, Sep 2002.

[148] M. Sternad, T. Ekman, and A. Ahl¡E9¿n, "Power prediction on broadband channels," in *Proceedings of the 53rd IEEE Vehicular Technology Conference — VTC 2001 Spring*, vol. 4, pp. 2328–2332, May 2001.

[149] J. Klaue and A. Aguiar, "Robust real-time channel prediction based on inaccurate instantaneous measurements: an approach," in *Proc. of 6th IEEE International Workshop on Signal Processing Advances for Wireless Communications (SPAWC)*, (New York, NY, USA), June 2005.

[150] S. Semmelrodt and R. Kattenbach, "Performance analysis and comparison of different fading forecast schemes for flat radio channels." COST Technical Report, Jan 2003.

[151] S. Haykin, *Adaptive Filter Theory, 3rd Ed.* Prentice Hall, 1996.

[152] A. Aguiar, "Heuristic for channel prediction of wlan channels," Tech. Rep. TKN-06-007, Telecommunication Networks Group, Technische Universität Berlin, Oct. 2006.

Appendix A

Acronyms

ACL	Adaptive Cross Layer
AR	Auto-Regressive
AP	Access Point
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BTC	Bulk Transfer Capacity
caRR	channel-aware Round Robin
CBQ	Class-Based Queueing
CBR	Constant Bit Rate
CDF	Cumulative Distribution Function
CRC	Cyclic Redundancy Check
CSDPS	Channel-State Dependent Packet Scheduler
C-IFQ	Chanel-State independent Fair Queueing
DIV	Distortion in Interval
ECDF	Empirical Cumulative distribution Function
ELF	Effort-Limited Fair
EXP	Exponential Scheduler
FIFO	First In First Out
FM	Flow Manager
FQ	Fair Queueing
GMR	Gradient algorithm with Minimum/Maximum Rate Constraints
GOP	Group of Pictures
GPS	Generalised Processor Sharing
HSDPA	High-Speed Downlink Packet Scheduling
IP	Internet Protocol
ISI	Inter-Symbol Interference
ISM	Industrial, Scientific and Medical
ITU	International Telecommunicaiions Union
ITU-T	Telecommunication Standardization Sector of the ITU

IWFQ	Idealised Wireless Fair Queueing
LOS	Line of Sight
LP	Linear Predictor
MA	Moving Average Predictor
MAC	Medium Access Control
PeLe	PeLe Scheduler
MC	Modified Covariance
MCS	Modulation and Coding Scheme
MIGS	Maximum Income Greedy Scheduler
M-LWDF	Modified Largest Weighted Deadline First
MOS	Mean Opinion Score
MPEG	Motion Picture Expert Group
MPQS	Multi-Dimension QoS Scheduler
MSE	Mean Squared Error
MTU	Maximum Transfer Unit
NMSE	Normalised MSE
OS	One Step predictor
PDF	Probability Density Function
PDR	Packet Delivery Rate
PER	Packet Error Rate
PESQ	Perceptual Assessment of Speech Quality
PFS	Proportional Fair Scheduler
PHY	Physical Layer
PL	Path Loss
PLR	Packet Loss Rate
PSK	Phase Shift Keying
PSNR	Peak Signal to Noise Ratio
QoS	Quality of Service
RR	Round Robin
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SBFA	Server-Based Fairness Approach
SCFQ	Self-Clocked Fair Queueing
SH	Shadowing
SNR	Signal-to-Noise Ratio
TAOS	Traffic-Aided Opportunistic Scheduler
TCP	Transport Control Protocol
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System

UWFS	Utility-Based Wireless Fair Scheduling
VBR	Variable Bit Rate
VoD	Video on Demand
VoIP	Voice over IP
WFS	Wireless Fair Scheduler
WFQ	Weighted Fair Queueing
WLAN	Wireless Local Area Network
WT	Wireless Terminal
WWW	World Wide Web

Appendix B

Own Publications

1. A. Aguiar, H. Karl, and A. Wolisz, "Effects of Prediction Inaccuracy on the Performance of Channel-State-Aware Link Layer Schedulers", In Proc. of 1st Int. Conf. on Mobile and Ubiquitous Multimedia, pp. 21-33, Oulu, Finnland, December 2002.

2. A. Aguiar and J. Gross, "Wireless Channel Models", Technical Report TKN-03-007, Telecommunication Networks Group, Technische Universität Berlin, April 2003.

3. A. Aguiar, H. Karl, H. Miesmer, and A. Wolisz, "A Framework for Evaluating Effects of Channel Prediction Inaccuracy on the Performance of Channel Adaptive Techniques", In Proc. of Intl. Conf. on Wireless Networks (ICWN), Las Vegas, Nevada, USA, June 2003.

4. A. Aguiar, C. Hoene, J. Klaue, H. Karl, H. Miesmer, and A. Wolisz, "Channel-aware Schedulers for VoIP and MPEG4 based on Channel Prediction", In Proc. of 8th Intl. Workshop on Mobile Multimedia Communications (MoMuC'03), October 2003.

5. A. Aguiar, H. Karl, and A. Wolisz, "Channel Adaptive Techniques in the Presence of Channel Prediction Inaccuracy", In Proc. of European Wireless 2004, Barcelona, Spain, February 2004.

6. A. Aguiar and J. Klaue, "Bi-directional WLAN Channel Measurements in WLAN Environments with Mobility", Technical Report TKN-04-002, Telecommunication Networks Group, Technische Universität Berlin, April 2004.

7. A. Aguiar and J. Klaue, "Bi-directional WLAN Channel Measurements in Different Mobility Scenarios", In Proc. of Vehicular Technology Conference (VTC Spring), Milan, Italy, May 2004.

8. A. Aguiar, H. Karl, H. Lederer, and A. Wolisz, "Channel-Adaptive Schedulers with State-of-the-Art Channel Predictors", In Proc. of European Wireless 2005, Nicosia, Cyprus, April 2005.

9. J. Klaue and A. Aguiar, "Robust Real-time Channel Prediction Based on Inaccurate Instantaneous Measurements: an Approach", In Proc. of 6th IEEE International Workshop on Signal Processing Advances for Wireless Communications (SPAWC), New York, NY, USA, June 2005.

10. A. Aguiar and H. Karl and H. Lederer, "Verfahren und Vorrichtung zur Versendung von Datenpakete", Patent nr. 10355117 of the German Patent and Trademark Office, filed November 2003, awarded October 2005.

11. A. Aguiar, "Heuristic for Channel Prediction of WLAN Channels", Technical Report TKN-06-007, Telecommunication Networks Group, Technische Universität Berlin, October 2006.

12. A. Aguiar, A. Wolisz, and H. Lederer, "Utility-based Packet Scheduler for Wireless Communications", In Proc. of The 6th IEEE Workshop on Wireless Local Networks, Tampa, FL, USA, November 2006.

13. A. Aguiar and A. Wolisz, "Comparative Evaluation of Prediction Heuristics for Wireless Channels", In Proc. of IEEE Globecom 2006, San Francisco, CA, USA, November 2006.

14. A. Aguiar and A. Wolisz, "Channel Prediction Heuristics for Adaptive Modulation in WLAN", In Proc. of Vehicular Technology Conference (VTC Spring), Dublin, Ireland, April 2007.

By the way, PeLe is Hawaii's goddess of fire.