Collection
des rapports
de Recherche
de l'ENST Bretagne

RR-2007002-RSM

Streaming of H.264 Video over HSDPA: Impact of MAC-Layer Schedulers on User-Perceived Quality

Diffusion de la vidéo codée H.264 sur HSDPA : impact de l'ordonnancement au niveau MAC sur la qualité perçue

Rapport Interne GET / ENST Bretagne

2007

Kamal Deep SINGH
Julio OROZCO
David ROS
Gerardo RUBINO

Streaming of H.264 Video over HSDPA: Impact of MAC-Layer Schedulers on User-Perceived Quality

Diffusion de la vidéo codée H.264 sur HSDPA : impact de l'ordonnancement au niveau MAC sur la qualité perçue

Rapport Interne GET / ENST Bretagne

Kamal Deep Singh
IRISA/INRIA Rennes
Campus de Beaulieu
35042 Rennes Cedex, France
e-mail: ksingh@irisa.fr

Julio Orozco
France Télécom R & D
2 avenue Pierre Marzin
22307 Lannion Cedex, France
e-mail: julio.orozco@orange-ftgroup.com

David Ros
Ecole Nationale Supérieure des Télécommunications de Bretagne
Rue de la Châtaigneraie, CS17607
35576 Cesson Sévigné Cedex, France
e-mail: david.ros@enst-bretagne.fr

Gerardo Rubino
IRISA/INRIA Rennes
Campus de Beaulieu
35042 Rennes Cedex, France
e-mail: rubino@irisa.fr

Abstract

High Speed Downlink Packet Access (HSDPA) is an enhancement of UMTS networks that supports data rates of several Mbit/s, making it suitable for data applications ranging from file transfer to multimedia streaming. In spite of the fairly high data rates that HSDPA offers, the shared downlink radio channel used in HSDPA is a challenging environment for delay- and loss-sensitive applications like video.

This paper focuses on the problem of streaming H.264-coded video over HSDPA. Our goal is to evaluate the impact of several MAC-level scheduling approaches on video quality. The video quality is evaluated using a tool that yields a good estimate of the *user-perceived*, subjective quality. We consider both a *best-effort* streaming scenario, that is, without any QoS support at the IP level, and a *QoS-aware* streaming scenario, in which some video packets, marked by the video server as being the most important for the decoding process, are protected from loss. Our results show the gains achieved by a QoS scheduler in terms of increased cell coverage and increased number of users that can be admitted in the system, for the same subjective quality experienced by the video users. Moreover, with the use of a QoS scheduler, the remaining capacity is fairly divided among the best-effort users.

Keywords

H.264, Video Streaming, Subjective Quality, QoS, 3G/UMTS Networks, HSDPA, Random Neural Networks.

Résumé

Le *High Speed Downlink Packet Access* (HSDPA) est une évolution des réseaux UMTS permettant de supporter des applications très diverses, telles des applications de données et multimédias, car il offre des débits pouvant atteindre plusieurs Mbit/s. Malgré ces débits relativement élevés, la liaison radio sur la voie descendante (de la station de base vers les terminaaux) pose tout de même des problèmes de performance aux applications sensibles aux délais et aux pertes, comme la vidéo, à cause du caractère partagé de cette liaison.

Ce rapport se concentre sur le problème de la diffusion (*streaming*) de la vidéo codée H.264 sur HSDPA. L'objectif de ces travaux est d'évaluer l'impact que diverses méthodes d'ordonnancement au niveau MAC peuvent avoir sur la qualité de la vidéo. Cette qualité est mesurée grâce à un outil offrant une bonne estimation de la qualité *subjective*, perçue par un utilisateur. Nous considérons aussi bien un scénario *best-effort*, sans aucun support pour la QoS au niveau IP, qu'un scénario de diffusion "consciente de la QoS", dans lequel on protège des pertes certains paquets IP ayant été marqués par le serveur comme étant les plus importants pour le décodage du flux vidéo. Les résultats présentés ici montrent l'avantage d'utiliser un mécanisme d'ordonnancement au niveau MAC tenant compte des contraintes de QoS des flux, car il offre une plus grande couverture dans la cellule et il permet de maximiser le nombre d'utilisateurs admis dans celle-ci, tout en conservant une bonne qualité subjective pour la vidéo. En outre, avec un tel mécanisme d'ordonnancement, la bande passante résiduelle est répartie de manière équitable entre les utilisateurs ne demandant pas de garantie de QoS.

Mots-clés

H.264, diffusion de la vidéo, qualité subjective, qualité de service, réseaux 3G/UMTS, HSDPA, Réseaux de neurones aléatoires.

Contents

1 Introduction

Universal Mobile Telecommunications System (UMTS) [1] is a third-generation, wireless cellular network that uses Wideband Code Division Multiple Access (WCDMA) as its radio interface technology. UMTS offers higher data rates with respect to older 2G and 2.5G networks and, with the Release 5 version, is evolving into an all-IP, wireless packet network. The increased bandwidth provided by UMTS allows for the deployment of a wide range of services, like voice, data and multimedia streaming services.

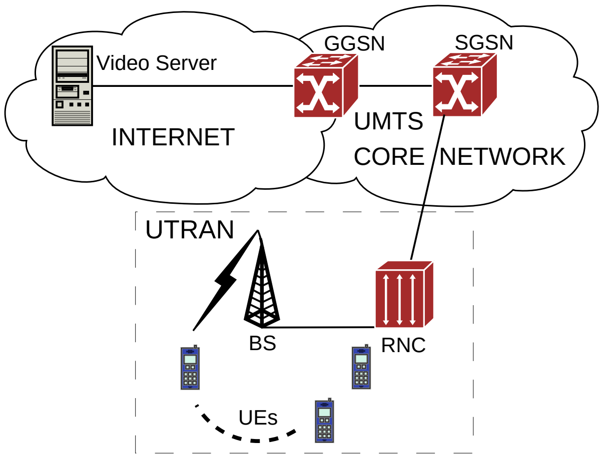


Figure 1: UMTS video streaming scenario.

Video streaming services are becoming popular and will likely be a significant source of revenues for cellular-network operators. Figure 1 shows a typical video streaming scenario and a simplified UMTS architecture consisting of the core network and the UMTS Terrestrial Radio Access Network (UTRAN). The UTRAN consists of Radio Network Controllers (RNC) which control several base stations (BS, or Node B). A mobile user present in the UTRAN can stream video on her User Equipment (UE) from a server in the Internet. The UTRAN is connected to the Internet through the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN) present in the core network. In the UMTS network, all the links in the core network are usually over-provisioned. Such over-provisioning of the core network, together with the fluctuations in radio channel quality that are inherent of wireless links, will often make the UTRAN act as a bottleneck. Therefore, resource management will be required in UTRAN to provide good Quality of Service (QoS) to the users.

High Speed Downlink Packet Access (HSDPA) [2–5] can be regarded as a packet-based enhancement of UMTS[1]. It supports data rates of several Mbit/s, making it suitable for data applications ranging from file transfer to multimedia streaming. Such data rates are

[1]The equivalent new technology in CDMA2000 1xEV-DO systems is called High Data Rate (HDR) [6].

(in principle) high enough for supporting the streaming of multimedia flows to several users at a time in a single cell. Nonetheless, due to its shared nature, the radio channel used to transfer data from the base station to mobile terminals remains a challenging environment for delay- and loss-sensitive applications like video. One of the main characteristics of HSDPA is the use of MAC-layer scheduling to perform resource management (i.e., bandwidth allocation between terminals), taking into account the radio channel conditions of all users. Such channel-adaptive scheduling may result in strong bandwidth fluctuations, causing packet loss and degradation in the quality of the received video. Additional factors like fairness between users, cell throughput or quality-of-service (QoS) parameters are also considered in some scheduling mechanisms proposed in the literature.

This paper focuses on the problem of streaming H.264-coded video over HSDPA, and particularly on the choice of a MAC-level scheduler suitable for video streaming. We are interested in assessing the impact of several schedulers on the *subjective* quality of the received video. Some studies have already been done (see 1.1) that use objective quality parameters like PSNR (Peak Signal to Noise Ratio), packet delay or loss and playout buffer underflow to study video streaming over HSDPA. However, it is well known that objective metrics like those above do not necessarily correlate well with user-perceived quality when a network is involved (see for instance [7]). In order to obtain a quality metric as correlated as possible with human visual perception, we employ a recently proposed technique, PSQA (Pseudo-Subjective Quality Assessment) [8–10] that yields an estimate of the subjective quality. PSQA is based on a specific type of queuing network used as a learning tool called Random Neural Network.

We consider two types of streaming scenarios, differing in the level of QoS support offered at the IP layer: (1) A *best-effort* streaming scenario, that is, without any QoS support at the IP level. (2) A *QoS-aware* streaming scenario, in which the network protects from loss the video packets that are the most important for the decoding process. This protection is achieved through both a packet-marking scheme at the H.264 streaming server, and a QoS-aware queue management in the UTRAN.

The paper is structured as follows. Section 2 provides some background on the H.264 video codec, and on H.264 video streaming over IP networks offering QoS support. Section 3 gives an overview of HSDPA and several MAC layer schedulers, and it also presents the QoS-aware IP queue management in the UTRAN considered in this study. Section 4 focuses on the problem of evaluating the subjective video quality and presents the quality estimation tool that we employed. Sections 5 and 6 present and discuss a performance evaluation study done with the well-known ns-2 network simulator [11]. Finally, Section 7 concludes the paper.

1.1 Related work

Many recent studies have focused on evaluating the performance of HSDPA-enhanced UMTS systems in the context of best-effort traffic [12, 13]. Parkvall et al. [5] provided an overview of key HSDPA technologies and showed the improvements in terms of user throughput, cell capacity and coverage. Jalali et al. [14] discussed the data throughput performance of the channel-adaptive Proportional Fair (PF) scheduler. Channel-adaptive schedulers are reported to take advantage of the multi-user diversity that is inherent of wireless systems. Borst [15] studied the user-level performance of the PF scheduler. A dynamic setting with a random nature of service demands is considered in comparison to many previous works assuming static user population. Bonald et al. [16] extended the results of [15] by using some general scheduling and admission control schemes. A "spatial" component is also considered in the traffic arrival model. Landstrom et al. studied congestion control for Best-Effort traffic in [17] using a new queue management scheme [18] designed for UMTS packet buffers.

Video streaming and providing QoS over HSDPA has also been the subject of many studies. In [19], Lo et al. did a performance evaluation study of H.264 streaming over a pre-Release 5 UMTS network (that is, a 3G network *without* HSDPA). Andrews et al. proposed a scheduler based on providing delay and rate guarantees in [20]. Barriac et al. [21] extended the PF scheduler to equalize the user throughputs for video users. Hosein showed in [22] that "good" schedulers can be designed based upon user utility functions. Kolding [23] proposed a new scheduler, based on similar principles as above, using a better estimation of the required scheduling activity for a given user. Ameigeiras [24] studied the performance of different HSDPA schedulers on CBR video quality. Leibl et al. [25] studied different strategies for joint buffer management and scheduling for H.264 coded video streaming over HSDPA. Lundevall et al. [26] studied the performance of 55 kbps and 90 kbps streaming services over HSDPA, and showed that a QoS-aware scheduler can protect the quality of video during high load conditions. Pedersen et al. [27] discussed the mapping of the schedulers to the QoS parameters specified by the 3GPP project.

2 H.264 Video Streaming over IP Networks

2.1 H.264 Video Coding

H.264 or MPEG-4 Part-10/AVC is the name of a recent standard for video compression that is likely to dominate applications like IPTV and digital video broadcast. This codec is the outcome of a joint activity between the video experts of the ITU-T and ISO. The resulting ITU-T recommendation is known as H.264, while in the ISO standard it is called MPEG-4 Part-10/AVC (*Advanced Video Coding*) [28].

The main goal of H.264 is the efficient and robust compression of rectangular video frames for applications like storage, videoconferencing, video-telephony, broadcast and streaming over a wide variety of networks.

H.264 offers an improved compression efficiency of over 50% against older codecs like MPEG-4/Part-2 or H.263+ [29] [30]. The enhanced coding efficiency of H.264 is not due to a dramatic change in the coding principles; as a matter of fact, H.264 follows the same model of block-based transformation and motion compensated inter-frame prediction used by older standards. The superior performance of H.264 comes from the aggregated effect of a series of algorithmic innovations, each one accounting for an incremental gain. For motion prediction, these enhancements include the use of small and variable block-sizes, quarter-pixel accuracy, motion vectors over picture boundaries and the use of multiple reference pictures [31]. Other enhancements include small block-size transformation and the use of advanced entropy coding methods (CABAC and CVLC) [32].

In addition to an improved coding efficiency, a very important feature of H.264 is that it is specifically designed for the transport of coded video over a variety of existing and future network transport technologies. The design of H.264 addresses this need for flexibility and customizability by means of separating the coded information in two layers: the Video Coding Layer (VCL) and the *Network Adaptation Layer* (NAL) [32]. VCL represents the coded video information in the most efficient manner possible. NAL formats the VCL data and organizes it in elements with the appropriate headers for transport or storage on a wide variety of technologies.

Regarding error robustness, H.264 includes several advanced features that are closely related to the concept of NAL [31]. The key syntax elements of the H.264 structure are flexible-sized *slices*. Each slice is conveyed in a single logical data packet (called NAL Unit or NALU). This flexibility allows for the efficient adaptation of the coded stream to the particular transport technology. Other features include the concepts of Flexible Macroblock Ordering (FMO) and Data Partitioning (DP) [33].

2.2 QoS-Aware Video Streaming

The term *QoS-aware video streaming* refers in general to any streaming architecture in which video data packets receive a differential treatment from the network, according to the relative importance of the coded-video information carried by those packets. The idea is that the underlying IP QoS mechanisms can be leveraged to offer a "better" treatment (for instance, in terms of packet loss rate) to packets that have been "marked" by the source application as containing data which are the most important for decoding purposes. In what follows we will present a QoS-aware H.264 streaming technique based on a particular QoS architecture, the *DiffServ* architecture.

2.2.1 DiffServ Architecture

The IETF (Internet Engineering Task Force) has developed some standards and technologies in order to enable Quality of Service (QoS) support in IP networks. The most widely accepted among them is the Differentiated Services architecture (DiffServ) [34]. In DiffServ, user traffic is separated into different Classes of Service based on their individual requirements. At the edge of the network, each packet is marked according to the treatment that it would like to receive inside the network. Packet marking may be done by edge routers, based on rate-metering or flow identification mechanisms, and also by the source application in order to request a differentiated (enhanced) treatment for some of its data [35]. The mark or tag is a coded value in the DiffServ Code Point (DSCP) field

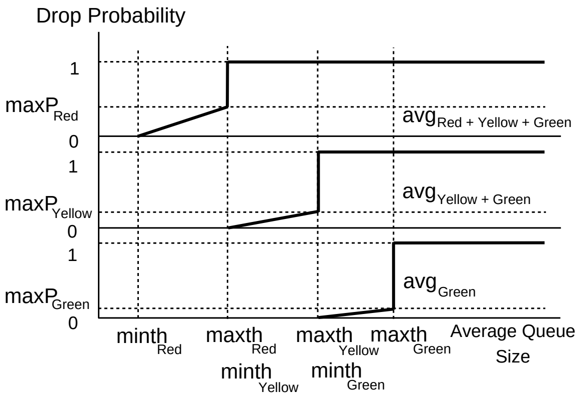


Figure 2: RIO with "staggered" discard probability functions.

of the IP header. Packets with the same DSCP value belong to the same class and will receive the same treatment by core routers inside the network. The different treatments a particular router can implement are called Per-Hop Behaviors (PHB).

One of the standardized PHB is Assured Forwarding (AF) [36]. The AF PHB allows a network provider supporting DiffServ to offer different levels of forwarding assurances, for IP packets accomplishing a target throughput for each network aggregate. It provides four AF forwarding classes with three dropping priorities each that define the relative importance inside an AF class. These drop priorities are usually identified with colors: *green* for the lowest drop precedence (i.e., the highest forwarding priority), *yellow* for the middle drop precedence and *red* for the highest one.

The basic mechanism used in setting up the AF PHB in routers is an *active queue management* (AQM) algorithm that implements differential packet dropping, taking into account the forwarding priorities of packets. RIO (RED with In and Out) [37] is the most common queue management mechanism suitable for AF. RIO is derived from the well-known RED algorithm [38], which aims at avoiding congestion by controlling the average queue size and comparing it against two thresholds, $\min_{th}$ and $\max_{th}$; inside this "congestion-avoidance" interval, packets are discarded with a probability that grows linearly with the average queue size. RIO can be regarded as a set of RED mechanisms running in parallel, where each instance of RED is associated to a given drop precedence. For a given precedence i, the corresponding RED instance monitors the average size of a "virtual queue" holding only packets with drop precedence $\geq i$. Several variants of RIO exist, of which Fig. 2 shows the "staggered thresholds" model [39]. It can be seen that packets of lower priority are more likely to be discarded as compared to higher priority packets. Moreover, incoming higher priority packets are only dropped after discarding all the lower priority packets when queue length starts to increase, i.e., during congestion.

2.2.2 DiffServ-Aware Streaming of H.264 video

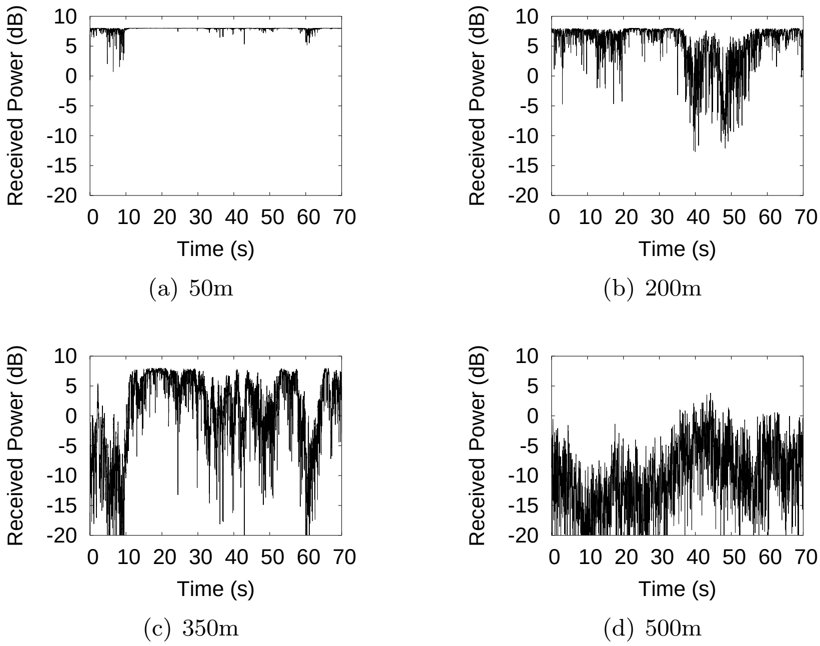
Video codecs perform compression by removing redundant information from the original signal both spatially (within a single frame) and temporally (between consecutive frames). This way, some frames are coded very much like still images and taken as reference for subsequent or previous frames. If a reference frame is lost, the effect on visual quality is large, because a whole group of derived frames can not be decoded correctly without the reference information. Thus, a way of realizing QoS-aware streaming consists of mapping the relative loss relevance (or *semantics*) of the video elements to the different loss priorities in a DiffServ AF network. For example, legacy codecs like MPEG-2 or MPEG-4 define a hierarchy with three types of frames, ordered by their loss effect on visual quality: I (reference), P and B. If the packets that transport I frames are marked with the highest AF priority, they will be the last to be discarded under congestion. Thus, for an equivalent loss rate, this approach should offer a better visual quality than Best Effort.

This type of DiffServ-aware mapping was proposed and evaluated for H.264 streaming in [40]. The authors describe different AF-aware coding and marking strategies based on the network adaptation and error resilience features of H.264. The simplest strategy consists of adapting the video slice size to the network packet size and mark packets containing I, P and B slices as high, middle and low priority, respectively, within a single AF class. A tool for *objective* video quality assessment was used for performance evaluation, and the results show that DiffServ-aware streaming yields better quality than Best-Effort under congestion (improvements of up to 30% were reported in [40]).

3 High Speed Downlink Packet Access (HSDPA)

HSDPA is an enhancement of UMTS that has been designed to meet the increasing demands of data and multimedia services. It provides many improvements as compared to the Release'99 version of UMTS. Higher data rates and lower delays are achieved using fast and channel-adaptive schemes like Adaptive Modulation and Coding (AMC), fast Hybrid-ARQ (HARQ) and fast scheduling based upon a very short time interval of 2 ms. This section presents a brief overview of the main features of HSDPA; for a more thorough presentation, see for instance [2–5].

A new transport channel called High Speed Downlink Shared Channel (HS-DSCH) has been introduced for HSDPA. HS-DSCH is supported by an auxiliary channel called High Speed Shared Control Channel (HS-SCCH). The goal of the latter is to allow for fast monitoring of the radio channel conditions of all users: every 2 ms, a UE can send to the base station a Channel Quality Indicator (CQI) over this control channel. This feedback makes it possible to optimize the transmission by means of channel-adaptive schemes. Thus, the CQI is used by AMC to adapt the coding rate, modulation scheme and number of codes employed, so that users having good channel conditions may be provided with very high data rates. Previously, in UMTS Release'99 systems, a power control mechanism was used in the dedicated channels to counter channel fading; however, power control finds



(a) 50m (b) 200m

(c) 350m (d) 500m

Figure 3: Example of received power by UEs (Ped A, 3km/h) at different distances from the Base Station.

a limited use in a shared channel like HS-DSCH, and so it is not used in HSDPA.

The handling of erroneous data frames has been improved in HSDPA. In UMTS release'99, frames with errors are retransmitted by the RLC protocol layer present in the RNC. In HSDPA, the MAC layer itself can retransmit the erroneous data, and retransmission times are reduced as MAC-layer functionalities have been moved to the base station. This also provides fast response times of the channel-adaptive schemes described above. At the MAC level, fast Hybrid-ARQ is used to retransmit the erroneous frames; UEs in turn do not discard the erroneous frames, but combine them with the successive retransmitted frames using schemes like Chase Combining and Incremental Redundancy.

HSDPA is a shared wireless channel and providing resource management (that is, allocating bandwidth among terminals) is complex due to variable radio channel conditions. For example, in Fig. 3 it can be seen that the channel conditions, especially for the farther users, may change rapidly. Resource management is performed by a scheduler that may adapt to fluctuating radio conditions. The Transmission Time Interval (TTI) in HSDPA has been reduced to 2 ms as compared to the 10, 20, 40, and 80 ms intervals supported by UMTS Release'99. Every TTI, the scheduler chooses the next user to be served based on the channel conditions of *all* the users. This channel-adaptive scheduling can be used, for instance, to maximize the global cell throughput by scheduling users only when their channel condition is good. Note that a given UE can experience strong fluctuations in bandwidth over time, due to such channel-dependent scheduling policies.

3.1 Scheduling in HSDPA

The HSDPA scheduler is the key to resource management in the UTRAN downlink, because it decides which user is to be scheduled at each time slot. In general, the scheduling decision does not only take into account the channel conditions: additional factors like fairness between users, cell throughput or quality-of-service (QoS) parameters are considered in the scheduling mechanism as well. The choice of a scheduler usually involves some trade-offs between these factors.

Numerous MAC-layer schedulers have been proposed and studied in the literature. In the remainder of this paper we are going to focus only on the following four types of schedulers:

- Round-robin (RR) scheduling.
- Maximum C/I (CI) scheduling.
- Proportionally fair (PF) scheduling.
- Rate-Guarantee (RG) scheduling.

We do not claim that this list is exhaustive, however, we believe that it covers reasonably well the different families of schedulers that have been proposed for HSDPA, and that they are representative of different design trade-offs and constraints.

3.1.1 Round-Robin Scheduling

The RR scheduler, which is probably the simplest one in terms of implementation, gives the time slot to users in a round-robin manner. Note that RR is fair with respect to system resources (i.e., time slots), but in general the capacity is *not* shared equally among the UEs, as each one may have different channel conditions. Moreover, this policy is not optimal with respect to maximizing the global throughput.

3.1.2 Maximum C/I Scheduling

Maximum C/I (CI) scheduling[2] gives the channel to the user having the best channel conditions at each given time slot. If $R_i(t)$ is the instantaneous data rate experienced by user i at time t, then the CI scheduler assigns the slot at time t to the user j having the maximum value of the index:

$$p_j = R_j(t) \tag{1}$$

[2]C/I denotes the signal-to-interference ratio that is used to estimate the optimal value of $R_i(t)$, with a given Block Error Rate target ($BLER_{\text{target}}$).

i.e., it gives the channel to the user able to achieve the highest instantaneous rate.

The CI scheduler provides the highest *cell* throughput as it always serves the users that can support the highest data rates. On the other hand, this scheme has the disadvantage of distributing the resources in a very unfair way. In a typical scenario, the users closest to the base station will get all the bandwidth, whereas the users farther away will be starved.

3.1.3 Proportionally Fair Scheduling

The Proportionally Fair (PF) scheduling algorithm [12, 14] assigns the slot at time t to the user i having the maximum relative index:

$$p_i = \frac{R_i(t)}{\lambda_i(t)} \tag{2}$$

where $R_i(t)$ is the instantaneous data rate experienced by user i and $\lambda_i(t)$ is the *average throughput* of user i, which is calculated as follows:

$$\lambda_i(t) = \left(1 - \frac{1}{\tau}\right) \cdot \lambda_i(t - \Delta t) + \frac{1}{\tau} \cdot R_i(t) \tag{3}$$

with $\tau > 1$ and Δt equal to the length of the TTI.

The PF scheduler offers a good trade-off between cell throughput and fairness, as it both gives the channel to the user having "relatively good" channel conditions and results in a fair distribution of resources.

This scheduler provides the so-called proportional fairness criterion defined in [41]. This criterion says that a set of user throughputs $\{\lambda_i\}$ is proportionally fair if it is feasible and, if for any other feasible vector of throughputs $\{\lambda_i^*\}$ the aggregate of proportional changes is zero or negative:

$$\sum_i \frac{\lambda_i^* - \lambda_i}{\lambda_i} \leq 0 \tag{4}$$

3.1.4 Rate-Guarantee Scheduling

QoS guarantees will be required by users of services like video streaming that have requirements in terms of minimum bit rate and maximum packet delay. User satisfaction is the main factor that should be considered. Satisfaction criteria are different for different services, each having their own QoS requirements. Thus, resource management should be aimed at fulfilling these QoS requirements instead of only providing fairness.

QoS classes and parameters have already been defined by 3GPP standards. These standards do not define the scheduler implementation details, and they provide flexibility to the system implementors to choose their own scheduler in order to map these QoS classes and parameters.

Several schedulers that can offer QoS assurances in HSDPA have been proposed in the literature. Pedersen et al. [27] discuss different options for providing QoS using QoS-aware schedulers like those presented in [20, 22]. Most of these schedulers take the slot assignment decision based on an index of the form:

$$p_i = \frac{R_i(t)}{\lambda_i(t)} B_i(t) \tag{5}$$

where $B_i(t)$, called a barrier function, is used to increase or decrease the priority of user i.

In [22] it was shown that, given a user utility $U(\lambda)$, it is possible to design a scheduler that will pick the user j such that:

$$j = \arg\max_i \{R_i(t) \cdot U_i'(\lambda_i(t))\} \tag{6}$$

Note that, in the above formulation, the corresponding utility functions for RR, CI and PF scheduling are $U(\lambda) = 1$, $U(\lambda) = \lambda$ and $U(\lambda) = \log \lambda$, respectively.

Hosein [22] proposed a utility function resulting in a scheduler with a barrier function $B_i(t)$:

$$B_i(t) = 1 + \alpha \exp\{-\beta \cdot (\lambda_i(t) - \lambda_{min}^{(i)})\} \tag{7}$$

with α, β, which corresponds to the case where a minimum throughput $\lambda_{min}^{(i)}$ is required by a user i. Equation (3) is used to estimate $\lambda_i(t)$. Remark that for best-effort users (i.e., users not requiring any guarantees) the barrier function becomes $B_i(t) = 1 + \alpha$, instead of taking $\lambda_{min}^{(i)} = 0$ [23].

In this paper, we will refer to the scheduler described by (5)-(7) as a Rate-Guarantee (GR) scheduler.

3.2 QoS-aware IP Queue Management in the UTRAN

The use of a QoS-aware video streaming technique, as the one discussed in Section 2.2.2, needs support from the underlying IP network. In particular, it is necessary that the network support some DiffServ-like functionality, in the form of a QoS-aware active queue management mechanism like the one used in the AF per-hop behavior (section 2.2.1): QoS-aware marking by the video server can then be combined with an AQM algorithm in network nodes, in order to prioritize the important data and drop the least important data during congestion.

In the UTRAN, IP packets are queued at the Radio Network Controller (RNC) before being fragmented and sent to the MAC layer. The RNC keeps *per-user* packet queues, in order to optimize the channel adaptation as each user has different channel conditions. The use of a flow control mechanism, between the MAC layer (at the base station) and the RNC, ensures that no loss occurs in the MAC queues due to congestion or bandwidth drops. This makes the (IP) queues at the RNC the bottleneck in our configuration, since

IP packets are dropped if the RNC queue limit is reached. Therefore, for the purposes of the QoS-aware video streaming architecture we assume that RNC buffers implement a QoS-aware AQM scheme; in particular, the RIO algorithm is employed for differential packet dropping.

4 Estimation of Subjective Video Quality

To evaluate the perceived quality we use the PSQA tool proposed in [8, 9] for video and audio flows respectively. The technique consists of the following procedure. We take a short video sequence σ and we send it through a network where we can control the values of a vector of J parameters $p = (p_1, \cdots, p_J)$ that we have selected as having the maximal impact on the quality, as perceived by a panel of human observers. We randomly choose $M + N$ different combinations of the values of the J parameters that we put into two different sets denoted by $\mathcal{M}$ and $\mathcal{N}$. Each of them is called a *configuration*. With the kth configuration we associate the value given to sequence σ_k by the panel of humans, following a controlled *subjective testing* experiment according to some well established norm such as [42]. Then, we build a function $Q()$ of the J parameters, such that for any configuration $p \in \mathcal{M}$ the number $Q(p)$ is close to the quality of σ_k given by the human panel. This is done using a specific statistical learning method described below. After obtaining this $Q()$ function, we validate it by comparing its value on the configurations in the second set $\mathcal{N}$ and the quality number coming from the subjective testing experiment. If these values are close enough, the tool is validated.

To build the quality function $Q()$ we use a feed-forward Random Neural Network (RNN), that is, an open Markovian queuing network composed of $J + H + 1$ queues. Nodes 1 to J are input ones receiving customers from outside according to independent Poisson processes with rates ν_1 to ν_J. They send their customers to nodes $J+1$ to $J+H$. Customers leaving a node $J + h$, $1 \leq h \leq H$, are sent to node $J + H + 1$ and from it all customers leave the network. Nodes are independent $\cdot/M/1$ queues. The service rate at node i is μ_i. When leaving input node j, $1 \leq j \leq J$, a customer can be transformed into a "negative" one. In that case, when it arrives at a node $J + h$, $1 \leq h \leq H$, it "kills" itself and it "kills" one of the customers present at the node at that point in time. The routing is independent of everything in the network, and the probability that a customer that leaves node j goes to node $J + h$ as a normal (or positive) customer (respectively as a negative one) is fixed, and denoted here as $r^+_{j,J+h}$ (respectively $r^-_{j,J+h}$). We have $\sum_{h=1}^{H}(r^+_{j,J+h} + r^-_{j,J+h}) = 1$. Things go the same for transfers between node $J + h$ and the only output node $J + H + 1$; here, we simply have $r^+_{J+h,J+H+1} + r^-_{J+h,J+H+1} = 1$.

The number of customers at a node behaves as the *potential* of a neuron, and those flows of negative customers as *inhibiting* signals since they decrease the potential of the neuron. This is what makes this tool behave as a neural network, and [8, 9] and the more recent [10] show that, for this application, it behaves better than classical tools such as standard Neural Networks. When used as a learning tool, we must see the RNN as a black-box implementing a parametric function with J input variables (the rates of the external arriving flows) and one output one (the load of the only output node, assuming the network in equilibrium). In symbols, we see the load ϱ_{J+H+1} of the output node as $\varrho_{J+H+1} = \varrho_{J+H+1}(\nu_1, \cdots, \nu_J)$. The routing probabilities are the parameters of the function. Learning means finding an appropriate set of parameters, such that when the input of the function is the configuration p, the output $\varrho_{J+H+1}(p)$ is close to the quality of a sequence having encountered a network where those selected quality-affecting parameters have the values in p. Finally, remark that the mapping $p \mapsto Q(p)$ implemented in such a black-box is a rational function, having nice mathematical properties (this is worth noting, even if we don't exploit this in the paper). More details about the tool and the PSQA procedure can be found in the references given above.

In the QoS-aware video streaming case, we used the following four parameters as the inputs of the quality metric tool: the loss rate of green packets, the loss rate of yellow packets, the loss rate of red packets and the mean loss burst size of green packets. The first three parameters have a clear meaning. The fourth is there because of the particular impact on quality of the distribution of losses among green packets: it can be expected that loss bursts of green packets (containing the most important information) should have a strong impact on perceived quality. Note that we don't explicitly take delays into account because a late packet is considered as a lost one in our framework.

After the whole PSQA process, the result is a quality function of these 4 variables: after measuring the values of these loss ratios and the average size of the burst of green lost packets at some time t, the function provides an approximation of the *instantaneous* perceived quality of the video at t.

In this study the quality metric is used in the following way: the PSQA function is called every second and the 4 parameters are computed on a time window of length 5 s. After evaluating the quality at every second, the average for the whole sequence is computed.

5 Performance Evaluation

In what follows, we will describe a performance evaluation study done with the well-known ns-2 network simulator [11]. For the sake of clarity, presentation and discussion of results will be deferred to Section 6; in this section we will only explain the methodology and tools we used for this study, as well as the different simulation scenarios and parameters.

The goal of the study is to compare the performance of the four HSDPA schedulers discussed in Section 3.1, in terms of the subjective video quality perceived by end-users. The video is encoded by a H.264 codec whose output is a constant-rate bitstream. We consider two video streaming scenarios: a best-effort scenario (i.e., no QoS support at the IP layer), and a DiffServ-aware scenario implemented according to the principles described in Sections 2.2.2 and 3.2. The resulting subjective quality is estimated thanks to the PSQA tool presented in the previous section.

5.1 Simulation Platform and Models

In order to simulate the UTRAN, we used the EURANE extensions [43] to ns-2. EURANE models the UTRAN in detail, whereas the nodes SGSN and GGSN are just used to route IP packets from the Internet to the UTRAN and introduce some delays. In particular, in EURANE all the functions of the RLC (Radio Link Control Protocol) and MAC-hs (MAC in HSDPA) layers are implemented.

The RLC layer consists of two modes of operation, Unacknowledged Mode (UM) and Acknowledged Mode (AM). There are per-user queues in the RNC and a credit-based algorithm, specified in [44], is used for flow control between MAC and RNC. The MAC layer implements the chosen HSDPA scheduler and other functionalities like HARQ. The underlying physical layer is modeled in detail, as described in [45], and this model is used to compute a Channel Quality Indicator (CQI) which is fed back from UEs to the base station[3].

In order to be able to simulate both the four chosen schedulers and a DiffServ-aware streaming architecture, we implemented in EURANE a DiffServ queuing behavior as described in Section 3.2, as well as the PF and RG schedulers. Remark that each IP queue in the RNC (running RIO) holds packets from *a single flow*.

5.2 Methodology

The evaluation study was performed as follows. A raw (YUV) video sequence is coded by a H.264 software encoder, whose output (a bitstream) is fed to a parser and packetizer module. In order to be able to perform DiffServ-aware streaming, this parser first detects syntax elements (NALUs) in the bitstream, then sets the appropriate DSCP values in the header of the resulting IP packets, depending on the type of NALU carried by the packet. It is this pre-defined mapping function which implements the particular QoS marking strategy chosen. The output of the parser is a trace file in a format readable by the simulator.

Next, the video packet trace file is fed to the ns-2 simulator (compiled with the EURANE extensions). This trace file serves as a traffic generator during the simulation. A simulation script allows to define the particular scenario under consideration (network topology, simulation parameters, and so on). When the simulation is run, an output trace file is produced which contains delay- and loss-related information for every video packet sent by the (simulated) video server.

Finally, from the simulation output trace, relevant parameters are computed then fed to the PSQA tool which gives a quality score.

[3]For the benefit of the reader, a short summary of the physical layer model used in EURANE is given in Appendix A.

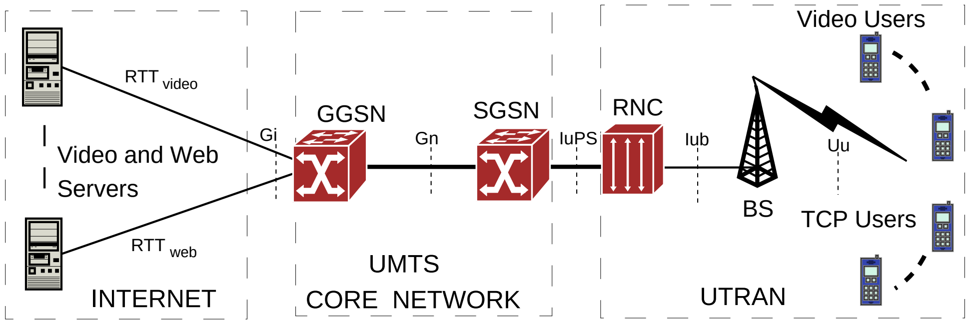


Figure 4: Simulation topology.

5.3 Simulation Scenarios

Table 1 summarizes the default values of simulation parameters; these parameters remain unchanged in all scenarios unless specified otherwise.

All scenarios under consideration correspond to the network topology shown in Fig. 4. There is a fixed number of video users (four) and several background TCP flows, representing users running data applications. Two types of background flows were considered: long-lived flows (corresponding to users downloading large data files) and web-like, on-off traffic (modeling WWW-browsing users).

The video is an H.264-coded sequence ("mother and daughter") of duration 64 seconds and bit rate $\approx$ 384 kbps, streamed in unicast mode using UDP.

For DiffServ-aware streaming, a simple marking policy taken from [40] was adopted: I-type slices are put in packets marked as "green", whereas the mark applied to P-type slices ("yellow" or "red") is chosen at random for each packet, according to a Bernoulli distribution with parameter $p = 0.5$. No B-type slices were produced by the encoder.

For a given scenario with a specific set of studied parameters, independent runs are performed for at least 45 times or till the 95%-confidence intervals for the PSQA metric are less than a given threshold θ. We take $\theta = 0.05$ for all the schedulers except CI, for which $\theta = 0.25$. This is because, as we will see later, the CI scheduler provides a quality that is highly unstable.

Finally, we will consider that a PSQA score lower than 3.0 means that the video quality is "bad", and users experiencing this quality will not be satisfied.

6 Results

We will now present and discuss results from several scenarios, obtained following the methodology and tools described in Section 5. Unless stated otherwise, all results correspond to the DiffServ-aware streaming case.

Table 1: Default simulation parameters.

Parameter	Default Value
Carrier frequency	1950 MHz
TTI	2 ms
HARQ cycle period	6
Time delay (TTI) in CQI	3
Site to site distance	1.0 km
Multipath fading environment	Pedestrian A
User Speed	3.0 km/h
Distance Loss model	Okamura-Hata
L_{init} (distance loss at 1 km)	137.4
Path loss exponent	3.52
Shadow fading	correlation model [46]
Correlation in shadow fading	40 m
Std. dev. in shadow fading	8 dB
BS Transmission power	38 dBm
Base station antenna gain	17 dBi
I_{intra} (intra cell interference)	30 dBm
I_{inter} (inter cell interference)	-70 dBm
$BLER_{target}$	10%
Maximum Data Rate	3.6 Mbps
Credit-allocation interval	15 timeslots
α in Equation (7)	1.25 [26]
β in Equation (7)	0.0313 kbps^{-1} [26]
λ_{min} in Equation (7)	400 kbps
τ in Equation (3)	1000
RLC mode	UM
Number of Video Users	4
WWW file size model	Pareto-distributed
Pareto shape parameter	1.2
Mean file size	10 KB
WWW thinking-times model	Exp-distributed
Mean thinking time	5 seconds
RNC queue limit Q	128 IP packets
RIO parameters (Fig. 2)	
$minth_{red}$, $maxth_{red}$	$0.1Q$, $0.3Q$
$minth_{yellow}$, $maxth_{yellow}$	$0.3Q$, $0.5Q$
$minth_{green}$, $maxth_{green}$	$0.5Q$, $0.7Q$
$maxP_{red}$	0.2
$maxP_{yellow}$	0.1
$maxP_{green}$	0.02
Averaging weight w_q	0.9
Round Trip Times	
RTT_{video}	100 ms
RTT_{web}	40 ms to 200 ms
$RTT_{Gn}, RTT_{IuPS}, RTT_{Iub}$	20 ms, 1 ms, 30 ms
Simulated time (single simulation run)	72 seconds

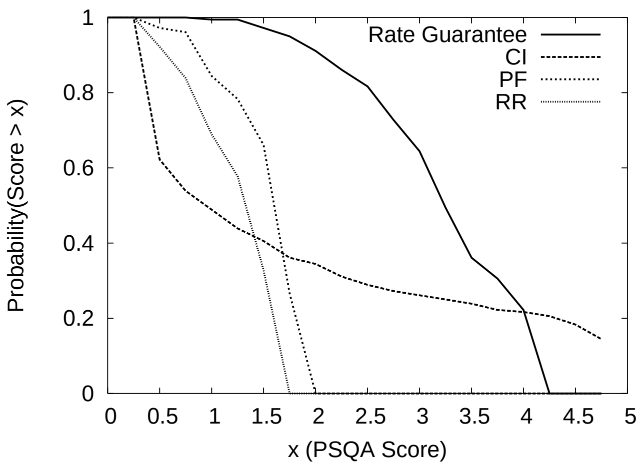


Figure 5: Inverse CDF of PSQA scores with 10 TCP long flows as background traffic. The bin size for the CDF is 0.25.

6.1 Long-Lived Flows as Background Traffic

In this scenario, $N > 1$ TCP flows were present as background along with the video flows. It was assumed that the TCP users had enough data to transmit for the duration of simulation.

6.1.1 Case I

Figure 5 shows the result when four video users were present with $N = 10$ TCP long-lived flows. The distance of the users was randomly varied with a uniform distribution, and a maximum distance from the base station equal to 500 m. This scenario is equivalent to one where a simple connection-admission control (CAC) is used that restricts the number of users queued at a single time.

It can be seen that the Rate-Guarantee (RG) scheduler performs best and is able to provide a PSQA score of 3.0 or more to about 65% of users, as compared to only $\approx$30% for the CI scheduler. A good CAC should be able to block users (or reduce their guaranteed bit rate value) that will eventually get a low PSQA score (35% of users, in this case) due to their bad channel conditions. It can be noticed that the PF and RR schedulers never achieve the required PSQA score as they divide the bandwidth in a fair way, without taking special care of video users. Thus, 10 TCP long flows turns out to be a high background traffic load for the PF and RR schedulers.

If we focus on the RG scheduler, then it should be noted that in this scenario the bandwidth given to 35% of the users was *"wasted"*, in the sense that the PSQA scores for these users are < 3.0 and so they would not be satisfied with the video quality. More bandwidth was wasted in the case of the other schedulers. These users should not have

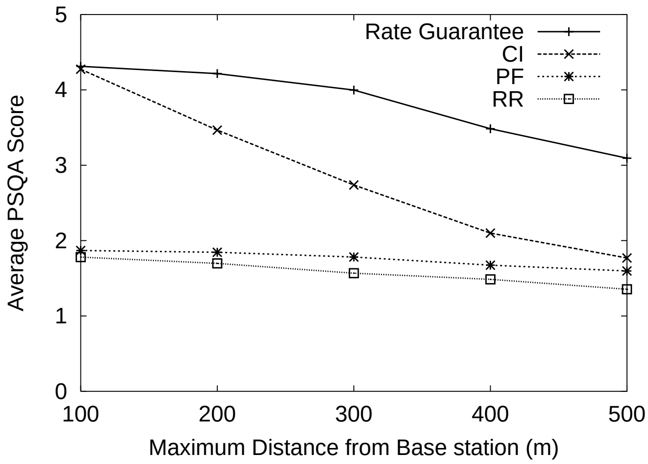


Figure 6: Average PSQA scores.

been admitted for the video service, as it is desirable to either give a satisfactory quality to the video users or none at all to save the bandwidth that will eventually be wasted.

This can be achieved by a CAC that monitors the channel conditions for a user demanding a particular video service. CAC will use a threshold to decide if the user be given access to the demanded video service or not. If the channel condition is lower than the threshold then CAC can either drop the demand or can re-negotiate a lower bandwidth (in the case of the RG scheduler). Moreover, if the channel conditions of a user who has already been given access fall considerably for a "prolonged time", then the CAC could drop the user to avoid the degradation of the quality experienced by the *other* video users. This call dropping should be done in extreme cases and only when the channel conditions deteriorate for a prolonged period, as fluctuations in the channel conditions over a short-time scale are inherent in the wireless link.

6.1.2 Case II

The aforementioned CAC may use a minimum threshold corresponding to the channel conditions, to decide whether a user should be given access or not. Remark that the received SNR (Signal to Noise Ratio) decreases in proportion to the distance of the UE from the base station. Thus, if the CAC uses such a threshold, that will in turn result in a given cell coverage radius. A higher threshold will mean that only the users near to the BS will be given access to the video service, whereas a lower threshold will mean that the users farther from the BS will be able to access the video service.

In this second scenario we employed a CAC to give access to video users whose distance d from the base station is $\leq d_{max}$. The value of d_{max} was increased in steps. This scenario can be used to estimate the cell coverage, since the scheduler that will provide user satisfaction for the highest value of d_{max} will provide better cell coverage. As in the

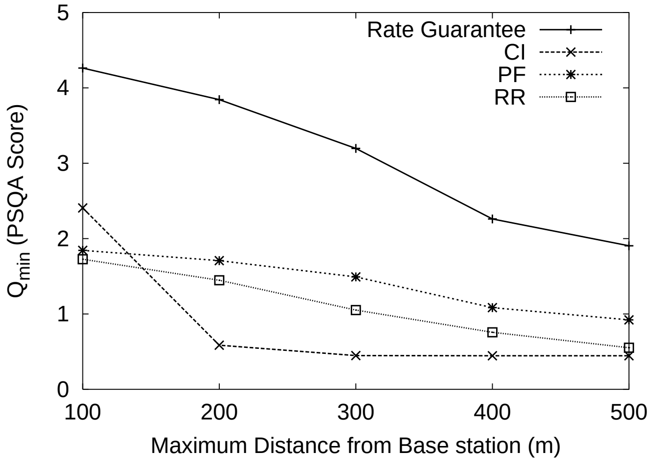


Figure 7: Minimum PSQA score obtained after removing lower 5 percentile of users.

previous case, the number of background TCP users is 10.

Figure 6 shows the average PSQA obtained with different values of d_{max}. It can be seen that, on the average, the RG scheduler performs much better than the other schedulers. It should be noted that the background load generated by the TCP users remains high even when d_{max} is as low as 100 m. Moreover, on the average the CI scheduler gives better quality scores than the PF or RR ones.

Let us denote by Q_{min} the value such that the 95% of the video user population, *across all simulation runs*, gets a PSQA score $Q \geq Q_{min}$. Q_{min} is a good design parameter, as in any system it is desired that a minimum quality should be experienced by as many QoS users as possible. Figure 7 shows the value of Q_{min} with varying d_{max}. In our scenario, the RG scheduler can satisfy users up to 300 m from the BS if a PSQA threshold of 3.0 is considered.

It is worth noting that, with respect to Q_{min}, the CI scheduler performs worst except for the users that are within 100 m from the BS. This is contrary to what can be observed when average PSQA scores are considered (Fig. 6). This means that the CI scheduler gives very good quality to few users, which increases the average quality score, but many users get a video quality that is very poor. Thus, it will not be possible to provide satisfaction to as many as 95% of the users. This particular case also emphasizes the usefulness of a metric such as Q_{min} for the performance evaluation.

It is interesting to look at the quality experienced by the *background* flows, measured here in terms of throughput. Figure 8 shows the CDF of per-user throughput obtained by the TCP flows. As expected the CI scheduler is the most unfair and gives very good throughput to $\approx$ 50% of the users and a very bad throughput to the rest. This figure also illustrates the trade-off provided by the RG scheduler: it provides satisfaction to most video users, but it results in lower per-user throughput for TCP flows. One good point

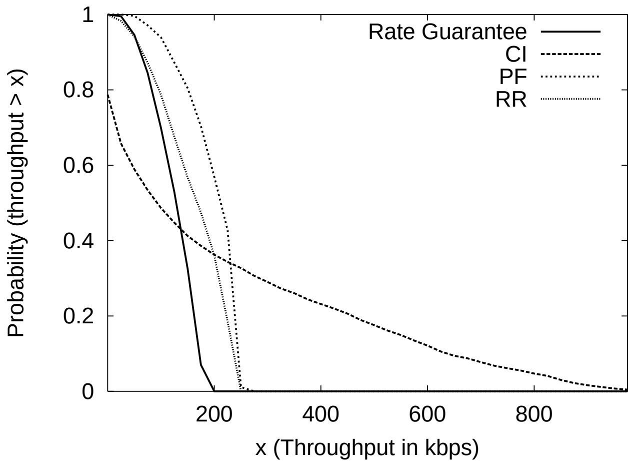


Figure 8: Inverse CDF of average per-user throughput obtained by background TCP flows. The bin size for CDF is 25 kbps.

about this scheduler is that it divides the remaining capacity to best-effort (TCP) users in a fair way. This can be inferred from the shape of the inverse CDF graph of the RG scheduler, which is similar to that of the PF and RR schedulers as seen in Fig. 8.

6.1.3 Case III

When the number of users increases, the quality will decrease as HSDPA is a shared channel. Thus, the CAC will have to decide on the maximum number of users that can be admitted in the system. We study the performance of the schedulers with different numbers of background TCP users. A benchmark similar to the one used in [23, 27] is employed to evaluate the performance of the scheduler, as follows. The number N of background flows is increased. When N cannot be increased while keeping up to 95% of the users satisfied, its value is called the *maximum load.*

Figure 9 shows the Q_{min} value for different values of N. It can be seen for instance that, if 95% of the users should get a PSQA score ≥ 3.0, then the maximum load with a PF scheduler is 4; that is, at most 4 TCP long flows can be accepted in the system. With the RG scheduler, up to 14 TCP long-lived flows can be served while satisfying the video quality constraint.

6.2 Mix of TCP Long-Lived and WWW Flows as Background Traffic

Since a mix of TCP long-lived flows and WWW flows is a more realistic background traffic pattern, we also ran simulations with 4 video flows, 2 TCP long-lived flows and N_{www} WWW flows. The value of N_{www} was increased in order to increase the load. WWW traffic was simulated as follows. A simple on-off session-level model of WWW

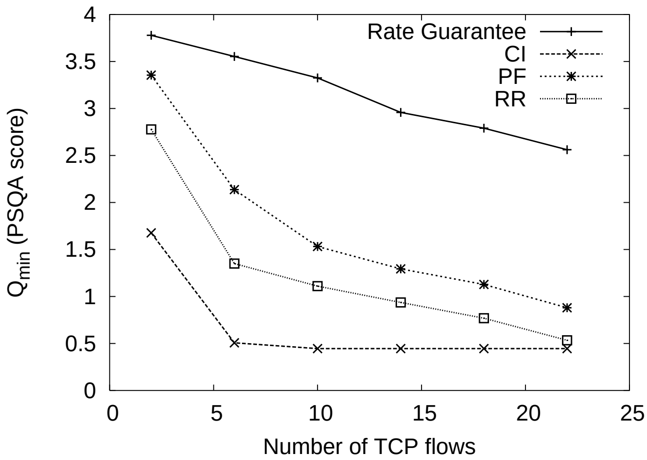


Figure 9: Minimum PSQA score obtained after removing lower 5 percentile of users. The number N of background TCP flows was varied.

traffic is used, in which downloading of a "file" (web page) is followed by a "thinking time", after which the user will initiate a new download. File sizes follow a truncated Pareto distribution with an average of 10 KB and with shape parameter equal to 1.2; sizes are truncated at a minimum value of 100 bytes and a maximum value of 2 MB. Thinking times were modeled using an exponential distribution with an average of 5 seconds.

A point worth noting is that, when the PF scheduler was used, Equation (3) was not updated if the RNC queue for the WWW user was empty. This is a practical approach because if (3) is updated for a user even when there is no data in the corresponding RNC queue, it will make the $\lambda_i(t)$ close to zero if no data arrives in the queue while the user is still "thinking". This will lead to a very high priority for the WWW flow when the next data burst arrives to the RNC queue. Thus, this approach avoids the decrease in the estimated value of $\lambda_i(t)$ when in fact it is not decreasing.

Figure 10 shows the value of Q_{min} for video users when the number of WWW flows is varied and the number of TCP long flows is kept equal to 2. In addition to 2 long TCP flows, it can be seen that if we assume a threshold of 3.0 for Q_{min}, then with PF at most 22 WWW users can be supported in the system. On the other hand, with RG up to 40 WWW users can be supported and still the satisfaction to the 95% of video users can be provided.

Figure 11 shows the CDF of the download times per single web page experienced by users when 22 WWW users are allowed in the system. It can be seen that the CDF for the RG scheduler is almost identical to that of the PF and RR ones. Thus, the RG scheduler is providing almost the same download times but with increased Q_{min} and hence better quality to the video users.

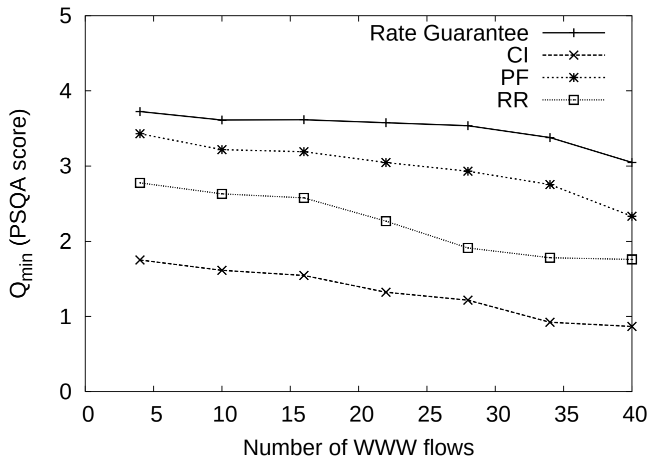


Figure 10: Minimum PSQA score obtained after removing lower 5 percentile of users. The number N_{www} of WWW flows was varied and the number of long-lived flows was kept equal to 2.

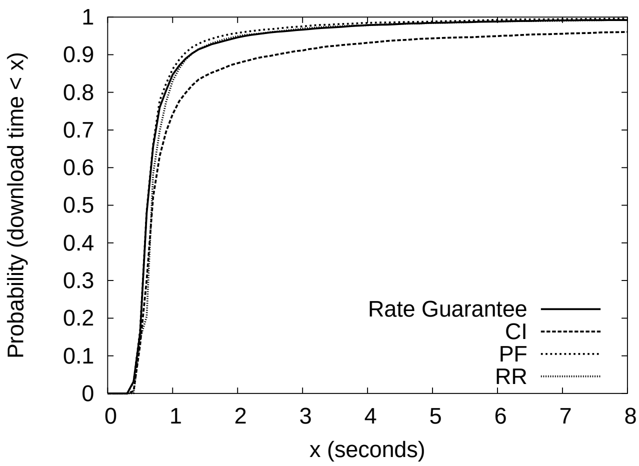


Figure 11: CDF of the download times experienced by users. The number N_{www} of WWW flows is 20 and the number of long-lived TCP flows is 2.

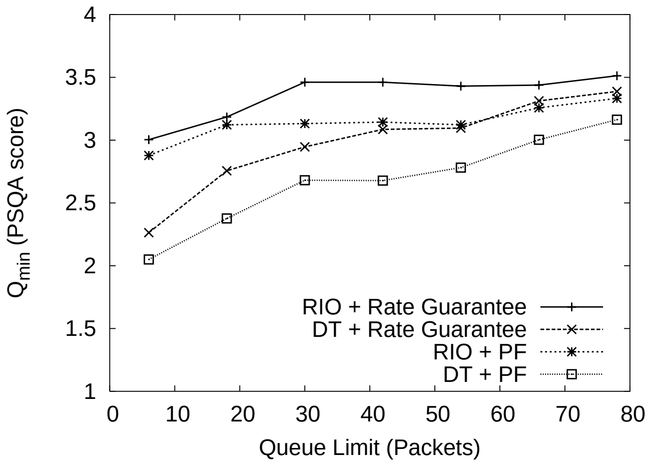


Figure 12: Comparison of DT and RIO. The number N_{www} of WWW flows is 18 and the number of long-lived TCP flows is 2.

6.3 Drop-Tail vs. RIO Queue Management

We implemented RIO queuing management in the RNC as it protects the video packets considered important by the decoding process. Figure 12 shows the comparison between drop-tail (DT) and RIO. In order to compare the two queue management schemes, we fixed the DT queue limit equal to the value of the $maxth_{red}$ threshold of RIO. This is done because in RIO all "red" packets are dropped if this threshold is reached and a significant loss of "red" packets will bring down the PSQA score to less than 3.0. Only two schedulers are compared and it can be seen that for both schedulers, RIO provides gains in terms of PSQA scores as it protects the important packets when packets have to be dropped.

7 Conclusions and Future Work

In this paper we focused on streaming H.264-coded video over HSDPA and were interested in evaluating the impact of different HSDPA schedulers on the video quality. For evaluating the performance we considered mixed traffic scenarios. QoS users streamed video on their equipment while Best-Effort (BE) users were represented by long-lived TCP flows and WWW flows. WWW users were modeled using demands in terms of download sizes and thinking times with statistical properties that have been established in the literature before. Though we do not claim of using very realistic traffic scenarios, we think that a reasonable mix of traffic was considered.

We also used a QoS-aware queuing mechanism called RIO in the radio buffers, to protect the video packets considered as important by the decoding process. In comparison to RIO, results showed that indeed DropTail (DT) queuing was degrading the quality by dropping the important packets when the situation of ultimately having to discard a packet arrived.

Different HSDPA schedulers were compared in terms of their impact on the video quality. Quality experienced by users was evaluated using a tool that yields good estimates of the user-perceived quality. Our focus was on achieving a video quality higher than a minimum threshold to at least 95% of the video users. While testing the 384-kbps video streaming service it was clear that, for our cell configuration, the full cell could not be covered and bandwidth would be wasted by users getting an unacceptable video quality. Thus, some approaches to limit access to the service were considered. First, the cell coverage limit of video users (in terms of maximum distance between the UE and the BS) was studied. After that the maximum video user distance was fixed and the limit in terms of number of users that can be admitted in the system was studied.

Further, it should be noted that no handovers are considered as a single cell[4] is analyzed in our study; taking handovers into account is left for future work. Moreover, an admission control algorithm that limits the access to the channel, based on the user's distance from BS, remains theoretical. A real implementation of CAC would have to monitor the channel quality for some time before taking the decision regarding the admission of the user. Another desired, but not implemented, property would be the *automatic* re-negotiation of e.g. the video bit rate, if the channel conditions become bad during the connection.

In terms of the maximum number of users that can be served at one time and the cell coverage in terms of maximum distance for the video users we found that QoS aware scheduler performed better. For the low load conditions all the fair schedulers like RR and PF gave good performance but as the load was increased it was only QoS scheduler that was able to guarantee a minimum quality to the 95% of the video users. The tradeoff was the lower per-user throughput that the BE users were getting in comparison. Nevertheless, the QoS-aware scheduler was still dividing the remaining capacity among the BE users in a fair manner and was significantly better than the Max C/I scheduler. Moreover, though the use of QoS-aware resulted in lower per-user throughput for long TCP flows it did not degrade the download times of the WWW users.

References

[1] H. Holma and A. Toskala, Eds., *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, 3rd ed. John Wiley & Sons Ltd., 2004.

[2] ——, *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications.* John Wiley & Sons Ltd., 2006.

[3] "3GPP TS 25.848, Physical layer aspects of UTRA High Speed Downlink Packet Access: Iub/Iur protocol aspects, release 4," 2001.

[4]Nevertheless, inter-cell interference is accounted for in the physical layer model of the simulator.

[4] "3GPP TS 25.308, High Speed Downlink Packet Access: Overall description, release 5," Mar. 2002.

[5] S. Parkvall, E. Englund, M. Lundvell, and J. Torsner, "Evolving 3G Mobile Systems: Broadband and Broadcast Services in WCDMA," *IEEE Communications Magazine*, pp. 68–74, Feb. 2006.

[6] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users," *IEEE Communications Magazine*, pp. 70–77, July 2000.

[7] R. Wu, T. Ferguson, and B. Qiu, "Digital video quality using quantitative quality metrics," in *4th International Conference on Signal Processing*, Beijing, China, 1998, pp. 1013–1016.

[8] S. Mohamed and G. Rubino, "A Study of Real–time Packet Video Quality Using Random Neural Networks," *IEEE Transactions On Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1071 –1083, dec 2002.

[9] G. Rubino, M. Varela, and S. Mohamed, "Performance Evaluation of Real-time Speech through a Packet Network: a Random Neural Networks-based Approach," *Performance Evaluation*, vol. 57, no. 2, pp. 141–162, may 2004.

[10] G. Rubino, P. Tirilly, and M. Varela, "Evaluating Users' Satisfaction in Packet Networks Using Random Neural Networks." in *Proceedings of ICANN'06*, Athens, Greece, September 2006.

[11] S. McCanne and S. Floyd, "`ns` Network Simulator," http://www.isi.edu/nsnam/ns/.

[12] T. E. Kolding, "Link and system performance aspects of proportional fair scheduling in WCDMA/HSDPA," in *Proceedings of IEEE VTC 2003-Fall*, vol. 3, Oct. 2003, pp. 1717–1722.

[13] M. C. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation," in *Proceedings of MobiCom '02*. New York, NY, USA: ACM Press, September 2002, pp. 71–82.

[14] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proceedings of IEEE VTC 2000 Spring*, vol. 3, 2000, pp. 1663–1667.

[15] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 636–647, 2005.

[16] T. Bonald and A. Proutière, "Wireless Downlink Data Channels: User Performance and Cell Dimensioning," in *Proceedings of Mobicom'03*, Sept. 2003.

[17] S. Landstrom, L.-A. Larzon, and U. Bodin, "Congestion control in a high speed radio environment," in *Proceedings of the International Conference on Wireless Networks*, Las Vegas, Nevada, USA, June 2004, pp. 617–623.

[18] M. Sagfors, R. Ludwig, M. Meyer, and J. Peisa, "Queue Management for TCP Traffic over 3G links," in *IEEE WCNC*, Mar. 2003, pp. 1663–1668.

[19] A. Lo, G. Heijenk, and I. Niemegeers, "Evaluation of MPEG-4 Video Streaming over UMTS/WCDMA Dedicated Channels," in *Proceedings of WICON'05*, July 2005.

[20] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150–154, Feb. 2001.

[21] G. Barriac and J. Holtzman, "Introducing delay sensitivity into the proportional fair algorithm for CDMA downlink scheduling," in *IEEE Seventh International Symposium on Spread Spectrum Techniques and Applications, 2002*, Sept. 2002.

[22] P. Hosein, "Qos Control for WCDMA High Speed Packet Data," in *IEEE Proc International Workshop on Mobile and Wireless Network Communications*, Sept. 2002, pp. 169–173.

[23] T. E. Kolding, "QoS-Aware Proportional Fair Packet Scheduling with Required Activity Detection," in *Proceedings of IEEE VTC 2006-Fall*, Sept. 2006.

[24] P. Ameigeiras, "Packet Scheduling and Quality of Service in HSDPA," Ph.D. dissertation, Institute of Electronic Systems, Aalborg University, Denmark, Oct. 2003.

[25] G. Liebl, H. Jenkac, T. Stockhammer, C. Buchner, and A. Klien, "Radio Link Buffer Management and Scheduling for Video Streaming over Wireless Shared Channels," in *Proceedings of the 14th International Packet Video Workshop*, July 2004.

[26] M. Lundevall, B. Olin, J. Olsson, N. Wiberg, S. Wanstedt, J. Eriksson, and F. Eng, "Streaming Applications over HSDPA in Mixed Service Scebarios," in *Proceedings of IEEE VTC 2004-Fall*, vol. 2, Sept. 2004, pp. 841–845.

[27] K. I. Pedersen, P. E. Mogensen, and T. E. Kolding, "QoS Considerations for HSDPA and Performance Results for Different Services," in *Proceedings of IEEE VTC 2006-Fall*, Sept. 2006.

[28] "ISO/IEC 14496-10 and ITU-T Recommendation H.264, Advanced Video Coding," 2003.

[29] N. Kamaci and Y. Altunbasak, "Performance comparison of the emerging H.264 video coding standard with the existing standards," in *Proceedings of IEEE ICME'03*, July 2003.

[30] A. Joch, F. Kossentini, and P. Nasiopoulos, “A performance analysis of the ITU-T draft H.26L video coding standard,” in *Proceedings of the 12th International Packet Video Workshop*, Apr. 2002.

[31] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia.* Wiley, 2003.

[32] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003.

[33] S. Wenger, “H.264/AVC over IP,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, July 2003.

[34] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services,” Internet Standards Track RFC 2475, IETF, Dec. 1998.

[35] A. Ziviani, J. de Rezende, O. Duarte, and S. Fdida, “Im- proving the delivery quality of MPEG video streams by using differentiated services,” in *Procedings of ECUMN’2002*, 2002, pp. 107–115.

[36] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured Forwarding PHB Group,” Internet Standards Track RFC 2597, IETF, June 1999.

[37] D. D. Clark and W. Fang, “Explicit allocation of best-effort packet delivery service,” *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362–373, 1998.

[38] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[39] R. Makkar, I. Lambadaris, J. Salim, N. Seddigh, B. Nandy, and J. Babiarz, “Empirical study of buffer management scheme for DiffServ Assured Forwarding PHB,” in *Proceedings of IEEE ICCCN*, 2000.

[40] J. Orozco and D. Ros, “Diffserv-Aware Streaming of H.264 video,” in *Proceedings of the 14th International Packet Video Workshop*, December 2004.

[41] F. Kelly, “Charging and rate control for elastic traffic,” *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[42] ITU-R Recommendation BT.500-10, “Methodology for the Subjective Assessment of the Quality of Television Pictures,” International Telecommunications Union (ITU), March 2000. [Online]. Available: http://www.itu.int/

[43] “EURANE,” http://www.ti-wmc.nl/eurane/.

[44] "3GPP TS 25.877, Technical Specification Group Radio Access Network: High Speed Downlink Packet Access: Iub/Iur protocol aspects, release 5," June 2002.

[45] N. Whillans, I. de Bruin, S. Oosthoek, F. Brouwer, A. Jehangir, and G. Heijenk, "Seacorn project, deliverable D3.2v2, End-to-end network model for Enhanced UMTS," Oct. 2003.

[46] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, no. 23, pp. 2145–2146, 1991.

A Physical layer model in EURANE

The transmitted signal T_x has a constant power to which the effects of the physical layer are added. First the intra-cell interference I_{intra}, assumed to be a constant value, is added. After that, the signal strength is reduced by a variable amount due to channel propagation loss that is modeled as follows. The propagation loss L consists of three elements: $L = P + S + M$, where P is the path loss, S is the shadow fading and M is the short-term fading. The path loss is described by the Okamura-Hata model for suburban areas: $P(x) = L_{init} + 10 \cdot n \cdot \log_{10}(x)$, where x is the distance between UE and BS in km, L_{init} is the loss at $1km$ and n is the path loss exponent.

The shadow fading is caused by the obstacles in the propagation path between UE and BS. EURANE uses the correlation model for shadow fading from [46]. The loss due to shadow fading is constructed as follows [45]: $S(x+\delta x) = aS(x) + b\sigma N$, where δx is the distance between two subsequent time samples and N is a random variable that satisfies the standard normal distribution. The parameter b is taken such that the vector containing all realisations equals σ. The parameter a is determined such that the autocorrelation function of S satisfies: $E[S(x)S(x+\delta x)] = a\sigma^2$.

The fast fading M model that is caused by multi-path is taken from the standard 3GPP models such as Pedestrian A, Vehicular A, etc. At the end, the inter-cell interference I_{inter} is also added to the signal and is also assumed to be a constant value. A value of *SNR* is estimated from the resulting signal that is equal to: $SNR = T_x - 10 \cdot \log_{10}(10^{\frac{I_{intra}}{10}} + 10^{\frac{I_{inter}+L}{10}})$, where T_x is the transmitted code power in dBm, L in dB and interference in dBm. Proper delays in the reception are also accounted.

The resulting signal is also used by the receiver to do the HARQ Chase Combining. Chase Combining is a type of soft combining that is used to combine the successive retransmissions of the frame in case of erroneous frames. Moreover, depending on the resulting signal R_x, an ACK or NACK is generated corresponding to whether the frame was received correctly or not. In order to determine whether a frame was received correctly or not, a random number from a uniform distribution is drawn and is mapped, using a BLER to *SNR* table, to a minimum value of *SNR* required for correct reception. This table corresponds to a AWGN channel. This minimum value of the *SNR* and the estimated value of the *SNR* from T_x is used for all HARQ transmissions to determine if the block was received correctly or not and an ACK or NACK is generated. After a maximum number of HARQ re-transmissions, the block is discarded and, depending on the RLC mode (UM or AM), is left to the RLC to handle it.

The resulting SNR is also used by the receiver to estimate the channel quality. The CQI value is generated, using estimated SNR value, for feedback. This CQI value is generated for a target Block Error Rate, $BLER_{target}$.

Imprimé à l'Ecole nationale supérieure des Télécommunications de Bretagne – Brest
Dépôt légal : Avril 2007
ISSN : 1255-2275