Multicast Scheduling for Scalable Video Streaming in Wireless Networks

Vladimir Vukadinović[†‡]
vvuk@ee.kth.se

György Dán[†‡]
gyuri@ee.kth.se

[†]Wireless@KTH
KTH, Royal Institute of Technology
16440 Kista, Sweden

[‡]ACCESS Linnaeus Centre
KTH, Royal Institute of Technology
10044 Stockholm, Sweden

ABSTRACT

We consider how relatively simple extensions of popular channel-aware schedulers can be used to multicast scalable video streams in high speed radio access networks. To support the evaluation, we first describe a model of the channel distortion of scalable video coding and validate it using eight commonly used test sequences. We use the distortion model in a detailed simulation setup to compare the performance of six schedulers, among them the Max-Sum and Max-Prod schedulers, which aim to maximize the sum and the product of streaming utilities, respectively. We investigate how the traffic load, user mobility, layering structure, and users' aversion of fluctuating distortion influence the streaming performance. Our results show that the Max-Sum scheduler performs better than other considered schemes in almost all scenarios. With the Max-Sum scheduler, the gain of scalable video coding compared to non-scalable coding is substantial, even when users do not tolerate frequent changes in video quality.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *wireless communication*.

General Terms

Algorithms, Performance.

Keywords

Video streaming, multicast scheduling, scalable video, distortion model, wireless networks.

1. INTRODUCTION

With the proliferation of mobile TV services, video multicast is expected to increase its share in the traffic load of cellular networks [1, 2]. In future systems, a significant part of this load might be carried on shared channels [3] (e.g., on the Downlink Shared Channel of LTE) for several reasons: First, video traffic produced by H.264/AVC compliant coders remains bursty even after being smoothed over several consecutive frames. Therefore, a dedicated multicast transport channel might need to be provisioned at a level significantly higher than the average bit-rate [4]. Multiplexing of multicast streams on a shared channel minimizes the over-provisioning of radio resources. Second, scheduling of the multiplexed streams on a time-slot basis—which is only possible on shared channels—may exploit the diversity in the varying radio conditions of different users to maximize the spectral efficiency of the system. This, however, requires new and efficient *multicast scheduling* algorithms.

The design and analysis of channel-aware schedulers for unicast have received significant interest within the research community. Channel-aware scheduling exploits the variations in the channel conditions (i.e., in the achievable throughputs) of the users to optimize the assignment of time-slots. If the satisfaction of each user is given by his utility curve, which defines the benefit of obtaining a time-slot to receive certain throughput, then the optimization problem is to maximize the overall utility under some fairness constraints. Maximum Carrier to Interference Ratio (Max-C/I) and Proportional Fair (PF) are examples of popular channel-aware scheduling schemes for unicast [5]. Both schemes are gradient-based: at each time slot, the schedulers aim to maximize the weighted sum of users' achievable rates, where the weights are given by the gradients of users' utility curves (Max-C/I) or by the gradients normalized by the users' average achieved throughputs (PF). For elastic applications it is often assumed that the utilities are given by the average achieved throughput and therefore the weights become equal to one (Max-C/I) or reciprocals of the average achieved throughputs (PF). Numerous unicast channel-aware schedulers targeting *non-scalable* video streaming have been proposed [6]-[11]. Typically, the users' utilities in these schemes are designed as functions of the video distortion and the weights are modified to take into account the playout deadlines of video packets. Packets within a stream are prioritized based on their importance (a packet is considered more important if its loss would cause larger increase in the video distortion). Therefore, in addition to being channel aware, these schemes are often labeled as deadline-aware and content-aware.

Unicast scheduling of *scalable* video streams has been recently addressed in [12, 13]. With scalable video coding, the video stream is encoded into a base layer and several enhancement layers. When a user has a bad radio channel and his achievable throughput is therefore low, he will receive the base layer only. When his radio conditions are favorable, he might be able to receive some additional enhancement layers. The main contribution of these works is in the proposed algorithms to

MMSys'10, February 22–23, 2010, Phoenix, Arizona, USA.
Copyright 2010 ACM 978-1-60558-914-5/10/02...$10.00.

prioritize the video packets within a stream. Each packet is assigned a distortion gradient that measures its importance. Scalable video coders employ complex motion-compensated prediction algorithms, which makes it very difficult to calculate the expected distortion increase due to the loss of a particular packet. Besides, the importance of a packet depends on the history of losses among previously transmitted packets, which is not available at the base station. This makes the scheduling schemes based on explicit packet prioritization too complex to be practical.

Scalable video coding is particularly suitable for multicast because it facilitates the delivery of streaming media to a set of receivers with heterogeneous channel capacities. When a non-scalable video stream needs to be delivered to all users in a multicast group, it has to be streamed at the rate of the weakest user in the group. This significantly limits the utility of users with favorable radio conditions. With the appropriate scheduling algorithms, scalable coding would ensure that the users with good channels receive additional layers and achieve better playback quality. Therefore, it is important to extend the schedulers for scalable unicast streaming to handle multicast scenarios.

In this paper, we address the problem of *multicast* scheduling of *scalable* video streams. To the best of our knowledge, this problem is addressed here for the first time. In our earlier wok [14], we studied the performance limits of multicast scheduling for a mix of non-scalable video streams and elastic flows. Multicast scheduling imposes a number of new challenges compared to unicast scheduling, especially in terms of fairness among different multicast groups and among users that belong to the same multicast group. On the one hand, the users' utility curves depend on the video stream and its rate-distortion characteristics. Therefore, users in different multicast groups have different utility curves. On the other hand, users in the same group may have the same utility curve, but they have diverse radio conditions and therefore different achievable throughputs. We formulate the users' utilities as functions of their average achieved throughputs for different layers. Hence, in the schedulers that we consider, the gradients of the utility are associated with layers, not with individual packets. Since the base station does not have to be aware of the utilities of the individual packets, the complexity of the schedulers is lower than that of content-aware schedulers. The tasks of the schedulers are to decide at each time-slot i) which multicast group should be served, ii) which layer of the stream should be transmitted, and iii) what the transmission rate should be. The choice of the transmission rate determines which subset of the users in the selected multicast group is able to receive the data. We compare the performance of six schedulers, and evaluate the effect of the layering structure on the scheduling performance. We explore the performance gains of scalable coding (compared to non-scalable coding) that can be achieved with the proposed scheduling schemes. We also consider how the gains are affected if end users are distortion variation averse.

The rest of the paper is organized as follows. In Section 2 we describe the test sequences, the distortion model for scalable video coding, and users' utilities. In Section 3 we describe different scheduling algorithms. We describe our evaluation methodology in Section 4, and present performance results in Section 5. Section 6 concludes the paper.

2. SYSTEM SETUP

We consider a single base station, where M video streams need to be delivered to M multicast groups with S_{m} (m=1,…,M) subscribers in each group. The videos are encoded in L layers (one base layer and L-1 enhancement layers) using SNR scalability of the SVC coder. The encoding rate for layer i is denoted by ρ_i and it is controlled by the quantization parameter for that layer. The encoding process introduces *source distortion* which comes mainly from quantization artifacts. The source distortion is measured in terms of the mean square error (pixel by pixel) between the original raw sequence and the subsequently encoded and decoded version of the same sequence. The source distortion is a function of the encoding rates $\rho_1,...,\rho_L$ of the layers, denoted as $D_L(\rho_1,...,\rho_L)$. In addition to the source distortion, a video stream can be distorted by packet losses in the transmission system, this is called *channel distortion*. The channel distortion is a function of the loss probabilities in the layers, denoted as $D_C(p_1,...,p_L)$. The total distortion at the receiver is the sum of the source distortion and the channel distortion. By D_i we denote the total distortion when only layers up to and including layer i are received and decoded. This distortion is the sum of the source distortion D_L and the channel distortion $D_i - D_L$ (Figure 1). The SNR scalability ensures that the total distortion at the output of the decoder decreases with each successfully received and decoded enhancement layer.

Our focus is on scheduling algorithms that can be implemented with low overhead. Therefore we consider schedulers that operate based on the distortion functions $D_L(\rho_1,...,\rho_L)$ and $D_C(p_1,...,p_L)$. This mode of operation requires significantly less side information than the content aware schedulers considered in the literature [13] because the scheduler only needs to care about which layer the individual packets belong to. In the following we describe the test video sequences and the distortion model of scalable video coding used in the paper, following which we describe the utility functions of the individual users.

2.1 Test Sequences

We encoded eight commonly used test sequences into three layers using the SVC reference software [15]. Hierarchical B-frames with GOP size 8 and intra-period 32 are used at each layer. The quantization parameter for the AVC-compatible base layer is set to 38, while the corresponding parameters for the first and the second enhancement layer are 32 and 26. Adaptive inter-layer prediction is used in the enhancement layers (i.e., the prediction is handled by the rate-distortion optimization framework implemented in the reference codec). The obtained data rates ρ_i and the distortions D_i for the layers are summarized in Table 1. The parameters α and β in the last two columns of the table will be explained in the next sub-section.

2.2 Distortion Model

The model we describe in the following was originally proposed in [16]. We made slight modifications to the original model and introduced new constraints to ensure the model's validity for the range of encoding parameters and video sequences used in this study.

We introduce the following notation: Let p_i denote the packet loss probability in layer i. Let π_i denote the probability that a packet from layer i has not been decoded, either because it has been lost

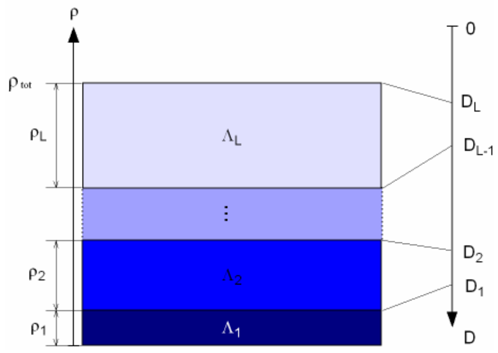

Figure 1. Encoding rates and distortions for scalable video.

Table 1. Parameters of the distortion model for the test sequences.

qp=(38,32,26)	ρ_1	ρ_2	ρ_3	ρ_{tot}	D_1	D_2	D_3	α	β
Crew	130.3	211.1	460.0	801.4	50.8	26.2	11.5	56.5	442.0
Foreman	88.5	167.3	339.0	594.9	48.0	23.4	10.0	83.1	614.0
Hall	45.4	104.7	236.6	386.7	34.2	18.0	8.8	21.5	150.9
Harbour	134.1	302.0	637.6	1073.7	95.9	45.1	19.8	58.0	644.1
Mobile	200.4	402.3	736.7	1339.4	124.7	52.2	21.3	146.8	1742.4
Mother	29.1	71.9	165.8	266.9	24.1	13.0	5.1	16.3	98.9
News	60.2	120.1	238.1	418.4	36.3	17.0	6.3	33.6	232.8
Soccer	127.0	198.3	454.4	779.8	52.0	27.9	11.1	122.4	923.8

or because of missing dependencies from lower layers. E.g., for a video stream encoded in three layers, the π_i's are given by

$$\begin{aligned} \pi_1 &= p_1 \\ \pi_2 &= \pi_1 + p_2 - \pi_1 p_2 = p_1 + p_2 - p_1 p_2 \\ \pi_3 &= \pi_2 + p_3 - \pi_2 p_3 = p_1 + p_2 + p_3 - p_1 p_2 - p_1 p_3 - p_2 p_3 + p_1 p_2 p_3 \end{aligned} \qquad (1)$$

The model of the channel distortion distinguishes between packet losses in the base layer, which are seen as more severe, and packet losses in the enhancement layers. It is commonly assumed that the distortion in the base layer increases linearly with the loss probability p_1 [15]. Results shown in Figure 2 (left) indicate that this assumption is valid only for low loss probabilities ($p_1 < 0.1$). The loss-distortion curves were obtained by inflicting random uniform packet losses on the test sequences (losses in a coded channel can be highly uncorrelated even when the fading process exhibits strong correlation). Lost frames were concealed by duplicating the previous frame in the sequence ("frame-copy"). In Figure 2 (right), we show that the slope of the increase in distortion depends on the quantization parameter used for the base layer. Therefore, the slope can be expressed as a function of D_1. We propose the following model for the channel distortion in the base layer

$$D_{C,1} = \alpha \sqrt{D_1}\, p_1 \,, \qquad (2)$$

where α is a parameter that is constant for a particular sequence and D_1 is the source distortion that depends on the quantization parameter. Figure 2 (right) shows a good agreement between the model and the results for the Foreman sequence. Our experiments with the other sequences (not shown in the paper) confirm the validity of (2).

The model assumes fine-grain scalability in the enhancement layers. Consequently, the channel distortion in an enhancement layer is proportional to the portion of the layer that has not been decoded. If all packets from layer *i* are received and decoded, the channel distortion is decreased by $D_{i-1} - D_i$ (Figure 1). If π_i is the share of packets from layer *i* that cannot be decoded, the channel distortion is decreased by $(D_{i-1} - D_i)(1-\pi_i)$ only. Hence, the channel distortion due to the undecoded packets is $(D_{i-1} - D_i)\pi_i$. Now, the total channel distortion can be written as

$$D_C = \alpha\sqrt{D_1}\,\pi_1 + (D_1 - D_2)\pi_2 + (D_2 - D_3)\pi_3 + \cdots + (D_{L-1} - D_L)\pi_L \,, \qquad (3)$$

where $0 \le \pi_1 \le 0.1$ and $0 \le \pi_{i\ne 1} \le 1$. Hence, the model is valid for loss rates in the base layer lower than 10%; this threshold is selected based on measurement results shown in Figure 2 (left). Based on (1), the model can also be written as

$$\begin{aligned} &D_C(p_1,\ldots,p_L) = \\ &= \beta p_1 + (D_1 - D_L)p_2(1-p_1) + \cdots + (D_{L-1} - D_L)p_L \prod_{i=1}^{L-1}(1-p_i) = \\ &= \beta p_1 + \sum_{k=2}^{L}\left((D_{k-1} - D_L)p_k \prod_{i=1}^{k-1}(1-p_i) \right), \end{aligned} \qquad (4)$$

where $\beta = \alpha\sqrt{D_1} + D_1 - D_L$, $0 \le p_1 \le 0.1$, and $0 \le p_{i\ne 1} \le 1$. The difference compared to the model in [16] is that we define the

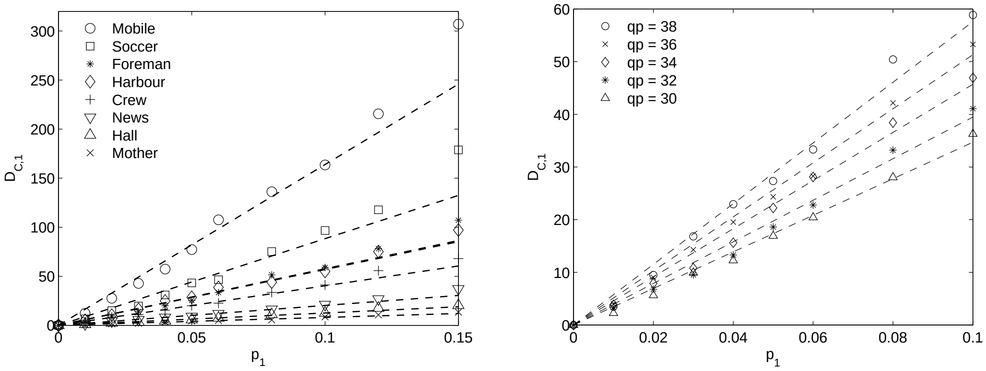

Figure 2. Channel distortion in the base layer encoded with qp = 38 for test sequences (left) and for the base layer of the *Foreman* sequence encoded with various quantization parameters (right). Dashed lines are obtained by fitting the parameter α in (2).

parameter β as a function of D_1 rather than as a constant. This allows us to vary the thickness of the base layer in Section 5.2. The parameters α and β of the test sequences are shown in the last two columns of Table 1. The data in the table define the shapes of the utility curves for the test sequences, based on which the evaluation of different scheduling schemes is performed in Section 5.

2.3 User Utility Model

Our premise is that a scheduler should be concerned with minimizing the channel distortion, and should not explicitly consider the total distortion (source and channel distortion) of a sequence. Minimizing the total distortion would imply that the scheduler would give preference to sequences with low source distortion. Such behavior might be unwanted: different streaming applications have different quality requirements and different users have different perceptions of video quality. Furthermore, a scheduling scheme that is jointly optimized with the source and channel encoders in order to provide the best perceptual quality is likely too complex to be practical. Our approach to minimize the channel distortion does not imply that the scheduler is oblivious to the source distortion and the layering structure of the encoded streams. The channel distortion is in fact heavily influenced by these factors, and hence the problem is still the joint optimization of the end-to-end distortion, but in two nested loops. In the inner loop the scheduler minimizes the channel distortion. In the outer loop the encoder allocates the source distortion as a function of the channel distortion. Our focus in this paper is on the inner loop, i.e., the problem of minimizing the channel distortion—we briefly discuss its impact on the outer loop, i.e., the encoder optimization. Consequently, the objective of our schedulers is to minimize the channel distortion $D_C(p_1,...,p_L)$, which is achieved when the following utility function is maximized

$$u(r_1,r_2,...,r_L)=\begin{cases}1-\dfrac{D_C(p_1,...,p_L)}{D_{C,MAX}}, & r_1\geq 0.9\rho_1 \;\; (p_1\leq 0.1)\\ 0, & r_1<0.9\rho_1 \;\; (p_1>0.1)\end{cases}, \quad (5)$$

where $r_k=(1-p_k)\rho_k$ is the average throughput for layer i. The channel distortion is normalized by the maximum channel distortion that can be tolerated ($D_{C,MAX}$) to obtain the utility function that takes values from the interval [0,1] for every sequence. Since we assume that a video stream is useless if the loss rate in the base layer is higher than 10%

$$D_{C,MAX}=D_C(0.1,1,...,1)=\alpha\sqrt{D_1}\cdot 0.1+D_1-D_L\,. \quad (6)$$

The gradients of the utility curve $\partial u/\partial r_i$ for $i=1,...,L$ determine the gains in the utility when different layers are chosen at the scheduler. Note that defining the utility as a function of the normalized channel distortion is common in the literature. For example, in [13] the utility gradient of a video packet is defined as the channel distortion caused by the loss of that packet normalized by the packet's size.

3. SCHEDULING ALGORITHMS

We consider the following setup: Let $R^*_{k,n}(m)$ denote the transmission rate selected by the base station for layer k in group m at time-slot n, $\Lambda^*_n(m)$ denote a video layer that is transmitted if group m is scheduled, and J^*_n denote a group that is scheduled for transmission at time-slot n. The moving average throughput for layer k of each user (m,s) is updated at the base station at each time-slot n as

$$\begin{aligned}r_{k,n}(m,s)&=(1-\omega)\cdot r_{k,n-1}(m,s)+\\&+\omega\cdot R^*_{k,n}(m)\cdot 1_{\{J^*_n=m\}}\cdot 1_{\{\Lambda^*_n(m)=k\}}\cdot 1_{\{R^*_{k,n}(m)\leq R_n(m,s)\}}\end{aligned} \quad (7)$$

where ω is an averaging window, $R^*_{k,n}(m)\in\mathbf{R}$ is the transmission rate selected for layer k in group m at time n from a set of possible transmission rates $\mathbf{R}$, and $1_{\{\}}$ is an indicator that equals one if the condition in the curly brackets holds and zero otherwise. Hence, user (m,s) will receive the throughput $R^*_{k,n}(m)$ for layer k only if the following three conditions are fulfilled:

i) group m is scheduled for transmission ($J^*_n=m$),

ii) layer k is selected among the available layers ($\Lambda^*_n(m)$),

iii) the achievable rate of user (m, s) is larger than the selected transmission rate ($R^*_{k,n}(m)\leq R_n(m,s)$).

We assume that the base station knows the achievable rates $R_n(m,s)$ of all users based on their channel quality feedbacks, which are readily available in modern systems (details are explained in Section 4.3). A user will be able to receive data with zero (or close to zero) loss probability if the selected transmission rate is lower than $R_n(m,s)$. Otherwise, the user will not be able to receive the data, which will be taken into account by the base station when calculating his average throughput $r_{k,n}(m,s)$ in (7). His average loss probability in layer k is given by $1-r_{k,n}(m,s)/\rho_k(m)$, where $\rho_k(m)$ is the encoding rate of layer k in stream m. We also assume that the parameters that determine the shapes of utility curves (Table 1) are known at the base station. They may change between scenes and need to be updated accordingly. In the case of pre-coded video, the parameters can be estimated off-line. In our setup, they remain constant since each of the test sequences contains a single scene.

In the following we consider six scheduling algorithms. The first two are opportunistic schemes, which we refer to as Max-Prod and Max-Sum. The objectives of these schemes are to maximize the product and the sum of users' utilities, respectively. They can be seen as extensions of the well-known Proportional Fair and Max C/I schedulers. The second two schemes differ from the first two in that they select groups at round-robin. The last two schemes differ from the first two in that they prioritize the base layers of the streams when they select groups.

3.1 Max-Prod Algorithm

The Max-Prod scheduler is an extension of the Proportional Fair scheduler [5], which is proposed for HSDPA and LTE. The Max-Prod scheduler is obtained when the product of utilities $\prod_{m=1}^{M}\prod_{s=1}^{S_m}u(m,s)$ is maximized, where M is the number of multicast groups (video streams), S_m is the number of subscribers in group m, and $u(m,s)$ is the utility of user s in group m, which depends on his achieved throughputs $r_k(m, s)$ for layers k=1,…, L of stream m. This is equivalent to maximizing the objective function:

$$\Phi=\sum_{m=1}^{M}\sum_{s=1}^{S_m}\log(u(m,s))\,. \quad (8)$$

We assume that only users whose utilities are larger than zero ($p_1 \leq 0.1$) are considered in the objective function (8). Otherwise, we say that the user is in outage and it is excluded from the objective. The outage probability is one of the performance measures that we use to compare different scheduling schemes in Section 5.

The implementation of the Max-Prod scheduler that we consider here is gradient-based: at each time-slot n, the scheduler aims to maximize $\Phi(n)-\Phi(n-1)$. Hence, we are interested in finding what the next best step is, regardless of the scheduling decisions made in the past. The presented algorithm is a heuristic—its asymptotic optimality for $\omega \to 0$ can be studied, but this issue is outside of the scope of this paper. In [17, 18], the author shows that similar gradient-based schemes for unicast may be asymptotically optimal even if the utility function is not strictly concave, which is the case in (5).

Let $u_n^{(k)}(m,s)$ denote the derivative (slope) of the utility function with respect to the received throughput for layer k : $u_n^{(k)}(m,s) = \partial u(m,s) / \partial r_k \big|_{r_k = r_{k,n}(m,s)}$. The gain in the objective function can be written as

$$\Phi(n)-\Phi(n-1) \approx \sum_{m=1}^{M} \sum_{s=1}^{S_m} \sum_{k=1}^{L} \frac{u_n^{(k)}(m,s)}{u_n(m,s)} \cdot \Delta r_{k,n}(m,s)$$

where

$$\Delta r_{k,n}(m,s) = (R_{k,n}^{*}(m) \cdot 1_{\{J_n^{*}=m\}} \cdot 1_{\{\Lambda_n^{*}(m)=k\}} \cdot 1_{\{R_{k,n}^{*}(m) \leq R_n(m,s)\}} - r_{k,n}(m,s))\omega \, .$$

Following the same reasoning as in [14], the gain is maximized in the following three steps.

In the first step, the optimal transmission rates for each layer of each group are selected so as to maximize the sum of the relative gains in the utilities of the users. The *transmission rate* for each group *m* and each layer *k* at time-slot *n* is selected from the set of available rates $\mathbf{R}$ as

$$R_{k,n}^{*}(m) = \arg\max_{R \in \mathbf{R}} \left\{ \sum_{s=1}^{S_m} \frac{u_n^{(k)}(s,m)}{u_n(s,m)} \cdot R \cdot 1_{\{R \leq R_{k,n}(m,s)\}} \right\} . \tag{9}$$

In the second step, the layers that maximize the gain in each group are determined based on the selected transmission rates. The video *layer* for each group *m* at time-slot *n* is selected as

$$\Lambda_n^{*}(m) = \arg\max_{1 \leq k \leq L} \left\{ \sum_{s=1}^{S_m} \frac{u_n^{(k)}(m,s)}{u_n(m,s)} \cdot R_{k,n}^{*}(m) \cdot 1_{\{R_{k,n}^{*}(m) \leq R_{k,n}(m,s)\}} \right\} . \tag{10}$$

Finally, in the third step, a group that contributes the largest gain is chosen for transmission. The *group* is selected for transmission at time-slot *n* as

$$J_n^{*} = \arg\max_{1 \leq m \leq M} \left\{ \sum_{s=1}^{S_m} \frac{u_n^{(\Lambda_n^{*}(m))}(m,s)}{u_n(m,s)} \, R_{\Lambda_n^{*}(m),n}^{*}(m) \cdot 1_{\{R_{\Lambda_n^{*}(m),n}^{*}(m) \leq R_{k,n}(m,s)\}} \right\} . \tag{11}$$

The algorithm performs iterative searches in (9), (10), and (11) to determine, respectively, the transmission rate, layer, and group to be scheduled. The search space and, therefore, computational complexity of the algorithm increases linearly with the number of possible transmission rates, multicast groups, and layers. In practical scenarios, however, these numbers remain modest.

3.2 Max-Sum Algorithm

The objective function of the Max-Sum scheduler is given by

$$\Phi = \max \sum_{m=1}^{M} \sum_{s=1}^{S_m} u(m,s) \, . \tag{12}$$

A gradient-based Max-Sum algorithm is obtained when the slope of the utility function in (9)-(11) is not normalized by the utility $u_n(m,s)$.

3.3 Max-Prod Round-Robin (Max-Prod-RR) Algorithm

The Max-Prod-RR is based on the following criteria:

- A transmission rate for the base layer is selected based on the achievable rate of the weakest user in the multicast group. This ensures that each user will be able to receive the base layer whenever it is scheduled. Transmission rates for the enhancement layers are selected so as to maximize the gain in objective (8), as in the case of the Max-Prod scheduler.
- A video layer is selected based on the received throughput in the base layer: The base layer is chosen until the received throughput of all not-in-outage users in the multicast group becomes equal to the encoded rate of the base layer (then we say that the base layer is "saturated"). When the base layers of all users are saturated, one of the enhancement layers is selected so as to maximize the gain in objective (8), as in the case of the Max-Prod scheduler.
- A multicast group is selected for transmission based on the round-robin principle.

Table 2. Summary of the multicast scheduling algorithms considered in this work.

	Max-Prod	Max-Prod-RR	Max-Prod-BLP	Max-Sum	Max-Sum-RR	Max-Sum-BLP
What rate?	Max-Prod	BL: Weakest user ELs: Max-Prod	Max-Prod	Max-Sum	BL: Weakest user ELs: Max-Sum	Max-Sum
Which layer?	Max-Prod	BL until all non-outage users are saturated, then Max-Prod		Max-Sum	BL until all non-outage users are saturated, then Max-Sum	
Which group?	Max-Prod	Round-Robin	Groups with unsaturated BLs served round-robin; if no such, Max-Prod	Max-Sum	Round-Robin	Groups with unsaturated BLs served round-robin; if no such, Max-Sum

3.4 Max-Sum Round-Robin (Max-Sum-RR) Algorithm

The Max-Sum-RR differs from the Max-Prod-RR in that the transmission rates for the enhancement layers are selected based on the Max-Sum instead of the Max-Prod criterion. The same holds for the selection of the enhancement layer when the base layers of all users are saturated.

3.5 Max-Prod with Base Layer Priority (Max-Prod-BLP) Algorithm

Max-Prod-BLP is based on the following criteria:

- The transmission rates for the enhancement layers are selected so as to maximize the gain in objective (8), as in the case of the Max-Prod scheduler.
- A video layer is selected in the same way as for Max-Prod-RR.
- Multicast groups with users whose base layer is not yet saturated are served based on the round-robin principle. When the base layer of all users is saturated, a multicast group is selected so as to maximize the gain in objective (8), as in the case of the Max-Prod scheduler.

3.6 Max-Sum with Base Layer Priority (Max-Sum-BLP) Algorithm

The Max-Sum-BLP differs from the Max-Prod-BLP scheduler in that transmission rates for video layers are selected based on the Max-Sum instead of the Max-Prod criterion. The same holds for the selection of the enhancement layer and the multicast group when the base layer of all users is saturated.

4. EVALUATION METHODOLOGY

We used extensive simulations to evaluate the performance of the schedulers under diverse conditions. In the following we describe the simulation setup, the mobility and the channel models used in the simulations, and our performance metrics.

4.1 Simulation Setup

The eight video sequences described in Section 2.1 are delivered to a number of multicast groups in a single cell. Multiple groups may be subscribed to the same video sequence, yet they are considered to be distinct multicast groups, which receive separate streams of the same sequence. This enables us to consider more than eight multicast groups. Each sequence is streamed to equally many groups, hence the total number of multicast groups is always a multiple of eight. The number of users in each group is randomly chosen to be between one and ten. The traffic load in the cell is given by the sum of the total rates (ρ_{tot} in Table 1) of all streams. The load is changed by varying the number of multicast groups; it ranges from 5.6 Mb/s for 8 groups up to 34 Mb/s for 48 groups.

The achieved throughput for the base layer of each user is initialized to $r_1 = \rho_1$, where ρ_1 is the bit rate of the base layer for the stream received by the user. Hence, the base layer for each user is assumed to be saturated initially. The video streams are scheduled according to one of the schemes described in Section 3. We calculate the achieved throughputs (r_1, r_2, and r_3) for each user in every time-slot as described in (7). Based on the average throughput we calculate the distortion (4) and the utility (5). If the utility of a user drops to zero (user's channel distortion increases to $D_{C,MAX}$) due to the low achieved throughputs, the user is considered to be in outage. The outage might be temporary: if the user, for instance, moves closer to the base station, it might be able to achieve higher throughputs. Therefore, the scheduler tries to "re-connect" to the user every 30 seconds by resetting his base layer throughput r_1 to the initial value. If a user was in outage during more than 50% of the simulated time, we say that this user was in outage during the simulation run. The results presented are obtained by averaging over 30 simulation runs. The simulated time in each run (after a warm-up period of 1 minute) was 10 minutes (300,000 time-slots).

4.2 Mobility Scenarios

The performance of channel-aware schedulers depends on the rate and the range of channel fluctuations, which is, to a large extent, determined by the nodes' mobility: their speeds and distances from the base station. Therefore, capturing the mobility of users is an important part of our study. We focus on urban scenarios and assume that nodes move on a topology that represents 1000×1000 m^2 of Chicago's downtown area. We consider three different mobility scenarios.

Manhattan: In this scenario nodes are restricted to travel along a grid of streets and intersections at a constant speed of 1 m/s. At street intersections, nodes proceed straight ahead (if possible) with probability 0.8, or turn to one of the adjoining streets with equal probabilities.

Static: In this scenario nodes are assumed to be static. Their channel conditions and achievable rates are constant over time and are drawn from the steady state distribution obtained for the Manhattan mobility scenario. This scenario is used as a base line for comparison.

UDel: In this scenario we distinguish between pedestrian (65%) and vehicular (35%) nodes. The nodes move on a map that represents a part of Chicago's downtown area. Pedestrians move according to the UDel mobility model [19, 20], which was developed based on empirical data on human mobility. They initially appear at random residential locations, and move according to their activity models. The activity model can be that of a working or non-working person (e.g., a tourist). The working pedestrians commute between their homes and offices, and possibly between offices and other locations during lunch breaks. Their activity model is based on empirical data from the US department of labor statistics. A non-working person visits one or more random locations before returning home. In our simulations 80% of the pedestrian nodes are working and 20% are non-working persons. More details on the UDel mobility models are provided in [19, 20]. The walking speeds of pedestrian nodes are drawn from a uniform distribution in [0.7, 1.8] m/s. However, for most of the time pedestrian nodes are stationary (sitting in the office or at home). The vehicular nodes represent persons that are using their mobile devices while they are riding public transportation, for example. These nodes are restricted to move on a grid of streets, as described in the first mobility scenario. Their desired speeds are uniformly distributed in the interval [5, 20] m/s. They might have to slow down due to platooning (car-following model) and to stop at traffic lights.

4.3 Channel Model

The channel model includes fast fading, shadowing, and propagation loss models. The fast fading model is 3GPP Typical Urban where the maximum Doppler shift is calculated based on the node's speed obtained from the mobility model. The shadowing model assumes the correlation function described in [21], where the correlation between fading samples depends on a node's speed as $a = e^{-V \cdot TTI / d_{cor}}$, where $d_{cor} = 40\text{m}$ is the de-correlation distance. The standard deviation of shadowing is set to 6 dB, which is within the range of typical values for medium-sized cells. Propagation loss is described by the Okamura-Hata reference model with the distance loss exponent 3.52.

We assume that the signal-to-noise ratio (SNR) is estimated for each node and reported back to the base station. The estimation is perfect and the reporting delay is negligible. Therefore, the base station has the perfect channel state information based on which it may determine the achievable data rate of each node. This is a reasonable assumption at low speeds. The achievable rate is the rate at which the expected block error probability P_b is lower than certain value (10^{-6} in our setup). The set of possible data rates—which is determined by the set of possible modulation and channel coding schemes—is given in the fourth column of Table 3 (values are taken from the HSDPA specification). Threshold SNRs required to receive data at the respective data rates with $P_b \leq 10^{-6}$ are provided in the fifth column. They are calculated based on an empirical model described in [13]:

$$SNR \;=\; 0.5(\sqrt{3} - \log(CQI)) \cdot \log(P_b^{-0.7} - 1) + 1.03 \cdot CQI - 17.3\,,$$

where CQI is the Channel Quality Index from Table 3. Nodes whose SNRs are above the threshold required for the selected transmission rate will receive the transmitted data with probability $1 - P_b \approx 1$. Nodes whose SNRs are below the threshold will not be able to receive the transmitted data—their achievable rates are lower than the selected transmission rate. The transmission power at the base station is constant over time (no power control). It is selected so that the probability that the SNR of a node is lower than the lowest of the SNR thresholds is below 1%. The cumulative distribution functions of the SNRs and the achievable rates of 600 nodes aggregated over a one-hour period are shown in Figure 3.

4.4 Performance Metrics

We use seven metrics to measure the different aspects of the performance of the streaming system:

- *The outage probability*: measures the proportion of users whose streaming performance is considered to be unacceptable due to the high channel distortion.

The following three utility-based metrics are used to measure the efficiency of the scheduling algorithms:

- *The average utility of users that were not in outage*: measures the channel distortion experienced by users whose streaming performance is considered to be acceptable.
- *The average utility of all users*: measures the trade-off between the outage probability and the utility of the users that are not in outage. A scheduler may, for instance, maximize the utilities of a small number of users and starve the rest, which will be reflected in the overall average.
- *The coefficient of variation (CoV) of utilities of users that were not in outage*: measures the variability of the channel distortion within a stream.

The following three PSNR-based metrics are defined in a similar way to measure the end-user performance, e.g., when comparing different encoding schemes:

- *The average PSNR of users that were not in outage*
- *The average PSNR of all users*
- *The coefficient of variation (CoV) of PSNRs of users that were not in outage*

5. PERFORMANCE EVALUATION

In this section we show simulation results for the six scheduling algorithms under various traffic loads and mobility models.

5.1 Scheduler Performance

We start the evaluation with the comparison of the six scheduling algorithms in terms of achieved utilities. The results for the first four performance measures are shown in the four rows of Figure 4. The first column of the figure shows the results for the static scenario. The Max-Prod scheduler provides very good performance at low traffic loads: the outage probability is negligible and the average utility tops the rest of the schemes together with the Max-Sum. However, as the load increases, the performance of the Max-Prod scheme in terms of the average utility and its variance worsens abruptly. Max-Prod gives preference to the users/groups with low utilities, which is obvious from (9)-(11). As the load increases, this approach leads to low

Table 3. Transmission rates in HSDPA.

CQI	*k/n*	*Mod.*	*R* (kb/s)	*SNR* (dB)
1	0.17	QPSK	1028	-12.63
2	0.21		1298	-12.23
...	...		...	...
14	0.68		4843	-1.65
15	0.70		4979	-0.68
16	0.37	16-QAM	5348	0.29
17	0.44		6284	1.26
...	...		...	...
29	0.84		12111	13.14
30	0.89		12779	14.13

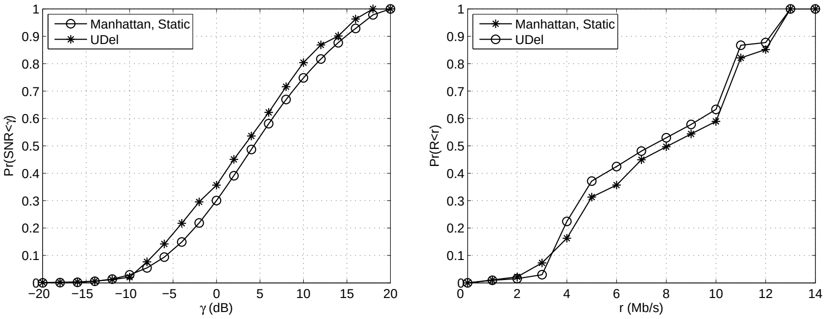

Figure 3. Aggregated cumulative distribution functions of SNR's (left) and achievable rates (right). The static scenario assumes the same distributions as obtained for the Manhattan mobility.

average utilities for everyone, although the outage probability is relatively well controlled. The high coefficient of variation, which reflects the high variability in the channel distortion, is due to the variation in base-layer throughputs, as the scheduler struggles to keep the utilities of as many as possible users above zero.

The Max-Sum scheduler, on the other hand, provides consistently good performance (in terms of all metrics) as the load increases. It is interesting to notice that Max-Sum also provides the lowest outage probability. Max-Sum-types of algorithms have been often criticized for being extremely unfair when the utility is a linear function of the throughput. In that case, the sum of the utilities can be maximized by serving only a small number of users with favorable conditions and starving the rest. In the case of video streaming, the utility is a concave function of the received throughput and it is limited from above due to the finite encoding rate, which alleviates the fairness problem of the Max-Sum scheduler to a large extent.

The two round-robin schemes (Max-Prod-RR and Max-Sum-RR) aim to share the time-slots equally among the groups that have at least one user that is currently not in outage. This leads to a very high outage probability as the load (number of groups) increases because the "fair share" of time-slots becomes insufficient for

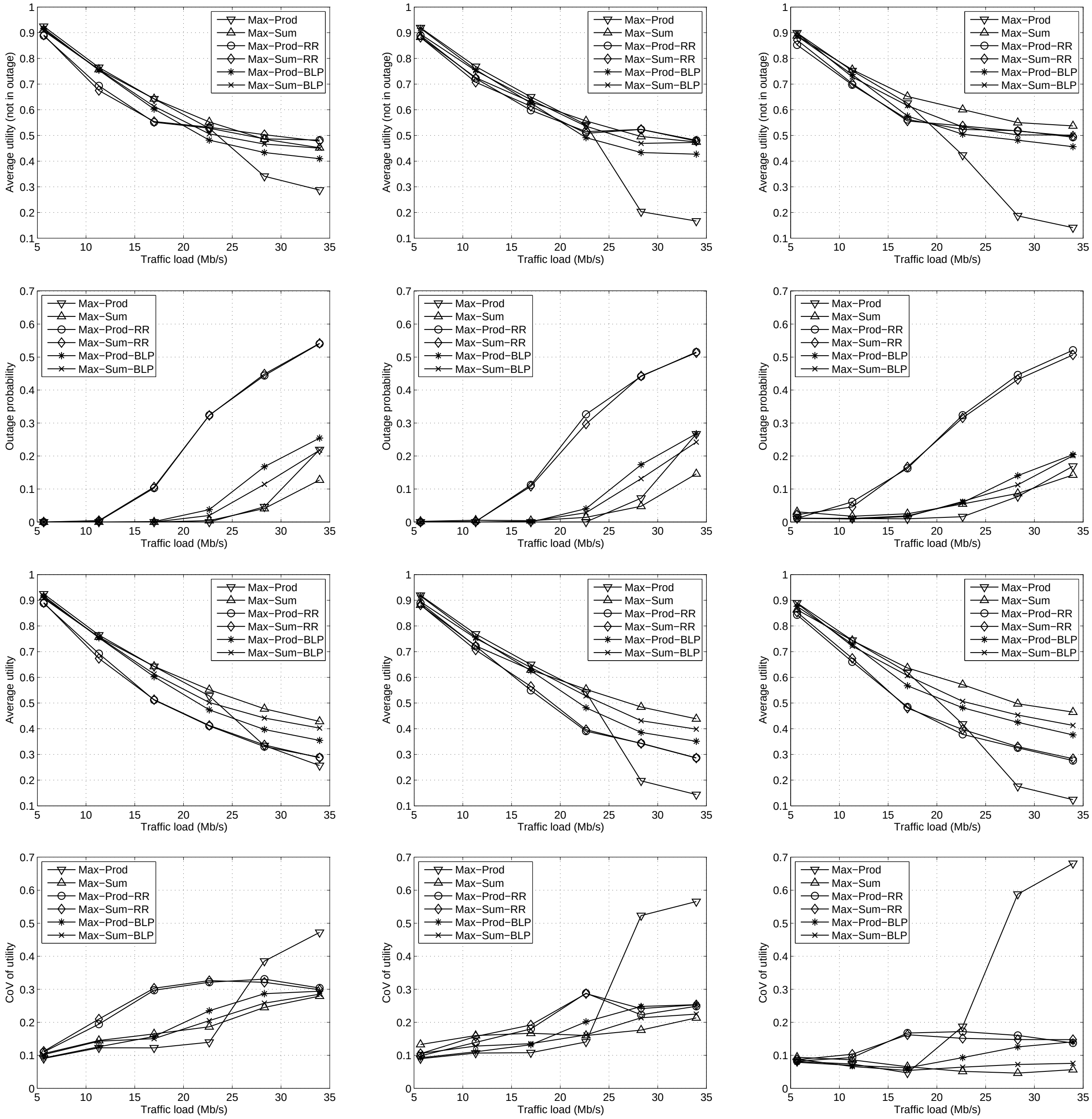

Figure 4. Scheduling performance for different mobility scenarios: static (left), UDel (middle), and Manhattan (right).

many. Although they are fair in terms of resources, the round-robin schemes are unfair in terms of channel distortion. For example, the "fair share" of time-slots could be sufficient to receive all three layers of a low bit-rate sequence, such as *Mother*, but not sufficient to receive the base layer of a high bit-rate sequence, such as *Mobile*. Giving priority to the base layer alleviates this problem: the two base layer priority (BLP) schedulers performed quite well, yet slightly worse than the Max-Sum.

The results for the UDel and the Manhattan mobility scenarios, shown in the middle and the right column of Figure 4, do not reveal any qualitative differences in the performances compared to the static scenario. All considered schemes, to a larger or lesser extent, attempt to exploit the variations in channel quality. Therefore, we would expect that the performance of the schemes improves with the level of user mobility (the static, followed by the UDel, and finally the Manhattan scenario). Our results confirm this assumption with one exception, the Max-Prod scheduler, which performs best in the static scenario, when the channel qualities are constant over time. This makes it easier for the Max-Prod scheduler to "identify" the users that should not be served when the load increases (to increase the product of utilities for the remaining users). Therefore, the gain in the utility is achieved at the expense of slightly higher outage probability.

5.2 Effects of the Layering Structure

In this section, we study how different layering structures fare in terms of distortion for the six scheduling algorithms. The purpose of this study is twofold: to show how the performance of the algorithms changes when the distribution of bit rates between the base and the enhancement layers varies and to provide an intuition on how the layering structure should be chosen to minimize the total distortion.

5.2.1 Extension of the Distortion Model

We consider the case where the total bit rate budget of the encoded video ρ_{tot} is fixed, but the "thickness" of the base layer $\eta = \rho_1 / \rho_{tot}$ varies ($\eta = 1$ represents a non-scalable single-layer video) (Figure 5). In the following we describe how the distortions D_i in the channel distortion model (4) change with η. A commonly accepted model for the source distortion of a non-scalable video is a decaying exponential function of the encoding rate [23]. This model also applies to the AVC-compatible base layer of a scalable video. Based on it, D_1 can be written as

$$D_1(\rho_1) = \chi \rho_1^{\varepsilon}, \tag{13}$$

where χ and ε are sequence dependent parameters.

It is well known that single-layer video coding results in lower source distortion than scalable video coding for the same total bit-rate. Therefore, it is reasonable to assume that the source distortion $D_L(\rho_1,...,\rho_L)$, decreases as the thickness of the base layer increases. $D_L(\rho_1,...,\rho_L)$ is minimized when $\rho_1 = \rho_{tot}$ (i.e. $\eta = 1$), hence

$$D_{L,MIN}(\rho_{tot}) = D_L(\rho_{tot}, 0, ..., 0). \tag{14}$$

We assume that D_L increases linearly with slope δ as the thickness of the base layer decreases

$$\begin{aligned} D_L(\rho_1,...,\rho_L) &= D_{L,MIN}(\rho_{tot}) + \delta \cdot (\rho_{tot} - \rho_1) = \\ &= D_{L,MIN}(\rho_{tot}) + \delta \cdot \rho_{tot} \cdot (1-\eta). \end{aligned} \tag{15}$$

The distortion penalty for not being able to decode any of the enhancement layers is $D_L - D_1$, while the penalty for the enhancement layer i is $D_{i-1} - D_i$. If we keep the distribution of $\rho_{tot} - \rho_1$ among the enhancement layers constant when ρ_1 varies, (i.e., $\rho_i / (\rho_{tot} - \rho_1) = \text{const.}$) it is reasonable to assume that the following holds

$$\frac{D_{i-1} - D_i}{D_L - D_1} = \text{const.}, \quad 2 \le i \le L. \tag{16}$$

Hence, we assume that the relative penalties for the enhancement layers remain constant when η changes.

We encoded the test sequences using different bit rates for the base layer to determine the parameters (χ, ε, and δ) of the model (13)-(16). The total bit rate ρ_{tot} and the distribution of the bit-rates among the enhancement layers are assumed to be the same as shown in Table 1. Results are summarized in Table 4. We do not consider extremely low values of η because the base layer has to contain essential information such as headers, motion vectors, and low-frequency DCT coefficients. Therefore, we restrict the validity of the model to $\eta \ge 0.1$. The model allows us to calculate the distortions D_i in (4) when the thickness of the base layer η varies.

5.2.2 Performance Results

We performed simulations to evaluate how users' utilities change with the thickness of the base layer η. The traffic load in the cell

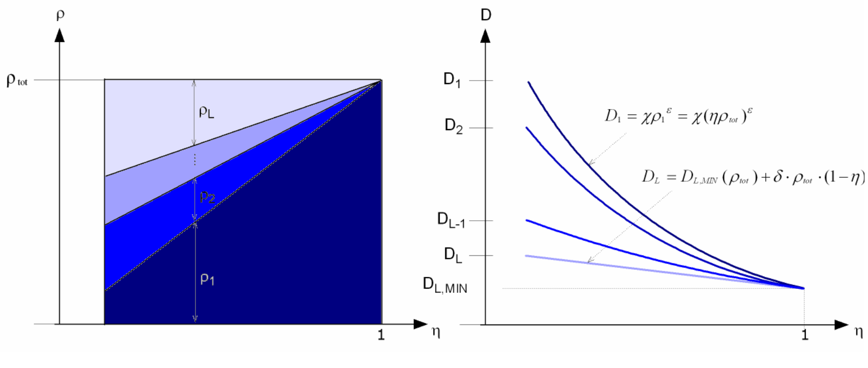
ρ
ρ tot
ρL
ρ2
ρ1
1
η
D
D1
D2
DL-1
DL
DL,MIN
$D_1 = \chi\rho_1^{\varepsilon} = \chi(\eta\rho_{tot})^{\varepsilon}$
$D_L = D_{L,MIN}(\rho_{tot}) + \delta\cdot\rho_{tot}\cdot(1-\eta)$
1
η

Figure 5. Illustration of how the data rates ρ_i and the distortions D_i change with the thickness of the base layer η.

Table 4. Parameters of the distortion model (13)-(16) for the test sequences.

	ρ_{tot}	$D_{3,\text{MIN}}$	δ	χ	ε
Crew	801.4	8.70	0.0042	5784.3	-0.972
Foreman	594.9	6.33	0.0073	5644.2	-1.063
Hall	386.7	6.73	0.0060	619.2	-0.759
Harbour	1073.7	15.76	0.0043	6722.4	-0.868
Mobile	1339.4	14.65	0.0058	49052.5	-1.127
Mother	266.9	2.75	0.0097	655.7	-0.980
News	418.4	2.90	0.0094	7568.9	-1.303
Soccer	779.8	7.34	0.0057	9699.6	-1.079

is 16.8 Mb/s (24 multicast groups in total, 3 groups per sequence) and it does not change with η. Results for the UDel mobility scenario are shown in Figure 6. Note that the average utility measures the channel distortion only and, therefore, any increase in coding efficiency is not reflected in it. The average utility for users that were not in outage, shown in Figure 6 (left), increases with the thickness of the base layer, except for very small values of η. However, the average calculated over all users, shown in Figure 6 (right) decreases due to the increasing number of users that are in outage. To explain this behavior we plotted the utility curves for $\eta = 0.1$, $\eta = 0.2$, and $\eta = 0.5$ for some of the test sequences (Figure 8). Not all sequences are shown to make the figures as legible as possible. The utility curves, as defined in (5), are functions of the achieved throughputs in the base layer (r_1) and in the enhancement layers (r_2 and r_3). To be able to plot them, we assume that the layers are served sequentially (the base layer, the first enhancement, and finally the second enhancement layer). Then we can plot the utilities as functions of the total achieved throughput $r = r_1 + r_2 + r_3$. The utilities are piecewise linear functions with three slopes, which correspond to the three layers. The tips of the arrows in Figure 7 indicate the average utility achieved by the user receiving a particular sequence. As shown in Figure 7 (left), for $\eta = 0.1$ users are able to receive (on average) the base layer and some parts of the enhancement layers. Exceptions are the users subscribed to the high bit-rate *Mobile* video, which receive the base layer only. When the base layer thickness increases to $\eta = 0.2$, the enhancement layers have to be dropped to accommodate for bit rates in the base layers, as shown in Figure 7 (middle). Also, some of the users, mostly those subscribed to the high bit-rate videos *Mobile* and *Harbour*, will not be able to receive the streams: the outage probability increases slightly. Although the utility of receiving the base layer increases with its thickness, this is not enough to compensate for the loss of the enhancement layers, and therefore we see a dip in the average utility of served users for $\eta = 0.2$ (Figure 6, left). Once all the enhancement layers are dropped, any further increase in the rate of the base layers has to be accommodated by dropping users (to be able to transmit at higher rates). Therefore, we see a sharp increase in outage probability for $\eta > 0.2$. The utilities of the users that are able to receive the thicker base layers increases, as indicated in Figures 6 (left) and 7 (right). However, the overall utility decreases due to the high outage probability (Figure 6, right). By far the most graceful decrease in the utility is achieved by the Max-Sum scheduler. The Max-Prod scheduler performed the worst in this respect, which is consistent with our previous observations. The results presented in this section show that from the common-good perspective it is preferred that all content providers encode the streams with "lean" base layers.

5.3 Scalable vs. Non-Scalable Coding

In this section we evaluate the end-to-end performance for scalable and non-scalable video streams. We measure the performance in terms of the PSNR, which reflects both the source distortion (the coding efficiency) and the channel distortion (the performance of the scheduler). We consider three scenarios: In the first scenario, the scalable (multi-layer) streams described in Section 2.1 are transmitted to the users. In the second scenario, non-scalable (single-layer) streams encoded at the same total rate as their scalable counterparts (ρ_{tot} in Table 1) are transmitted. In the third scenario, only the base-layers of the scalable streams are transmitted (the enhancement layers are dropped at the base station). The third scenario helps us to evaluate how much of the PSNR gain can be attributed to the enhancement layers. Results for the Max-Sum scheduler are shown in Figure 8.

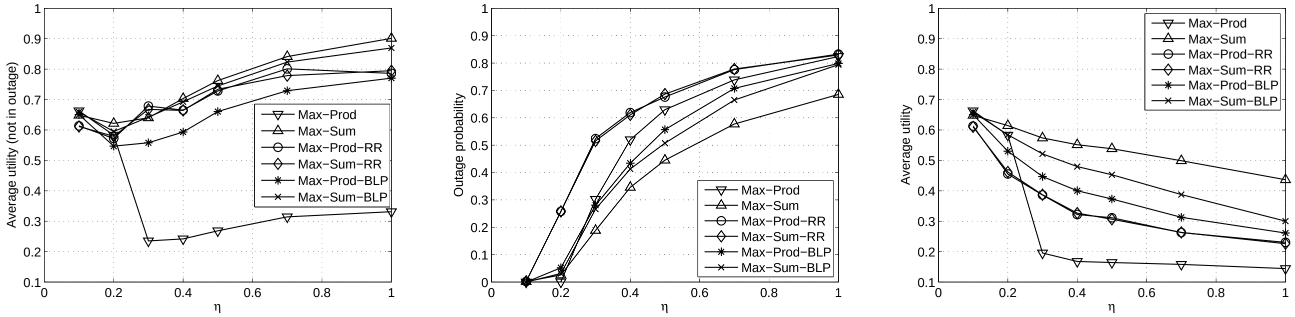

Figure 6. Performances of the scheduling algorithms vs. the thickness of the base layer (UDel mobility).

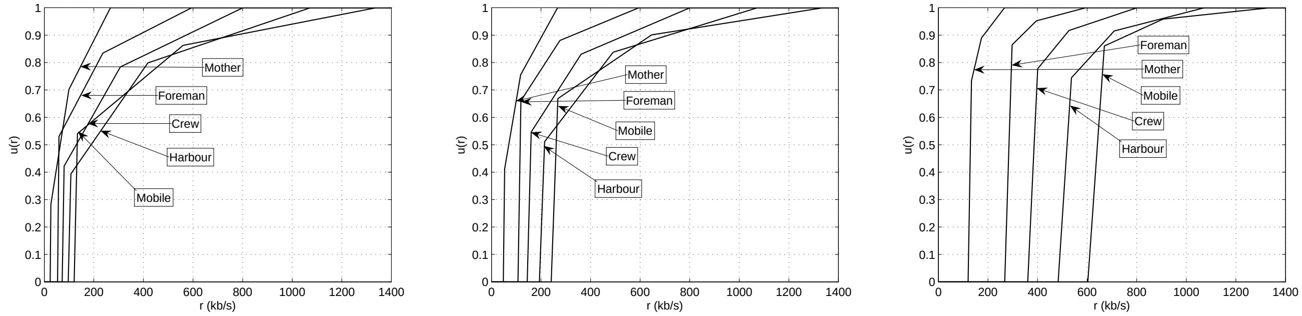

Figure 7. Utilities for different video sequences and η=0.1 (left), η=0.2 (middle), and η=0.5 (right). The tips of the arrows indicate the average utility achieved by users subscribed to a particular sequence under the Max-Sum scheduling policy.

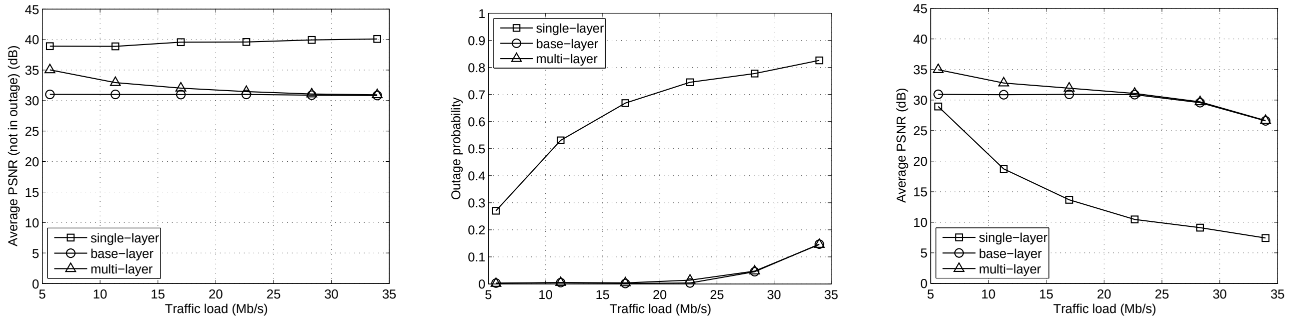

Figure 8. PSNRs and outage probabilities of different encoding schemes for Max-Sum scheduling and UDel mobility.

The results indicate that the PSNR of the scalable streams is on average 4 dB (low load) and up to 8 dB (high load) lower than the PSNR of the non-scalable streams. However, the better quality of the non-scalable streams is achieved at the price of very high outage probabilities. The average PSNR calculated over all users is shown in Figure 8 (right). The figure shows a significant gain for the scalable streams, which increases with the traffic load. The gain is mainly due to the high outage probability among the non-scalable streams. It starts to decrease at very high loads when scalable streams start to be dropped as well. The results for the base-layer indicate that, at low loads, up to 4 dB of the gain can be attributed to the enhancement layers. At high loads, the enhancement layers have to be dropped, so only the base layer of the scalable streams is received anyway.

5.4 Fluctuations in Video Quality

We are not aware of a simple and good model of the impact of the variations of the distortion on the user perceived quality, but it is generally accepted that the rapid fluctuation of the distortion is annoying. Hence, one of the concerns with streaming scalable video is the fluctuations in video quality it introduces if the number of decoded enhancement layers changes often. The fluctuations can be reduced by dropping some of the enhancement packets at the decoder, which increases the distortion. In this section our goal is to explore the trade-off between the variance of the PSNR and average PSNR due to scalable coding.

The enhancement layers typically have several “extraction points” that decoders may use to determine which frames should be decoded and which should be dropped. For example, the enhancement layers of the sequences considered in this paper contain four temporal layers (T0 contains I and P frames, while B frames are hierarchically organized in T1–T3) and each temporal layer is a potential extraction point. The problem of minimizing the fluctuations in quality while preserving the most of the enhancement gain is the problem of choosing the optimal extraction point. We consider a simple trimming scheme to decrease the fluctuations at the decoder. Clients monitor the packet loss rates in the enhancement layers and every 60 seconds they make decisions which enhancement layers to decode: If the loss rate in an enhancement layer was above 10% during the previous 60-seconds interval, all packets from that layer are discarded in the subsequent 60-seconds interval. Hence, no extraction points within the enhancement layers are used. In a worst case scenario the enhancement layers will be added and dropped every 60 seconds, which we assume to be acceptable annoyance for a typical viewer. This simple scheme provides a worst-case estimate of the loss in the average PSNR.

We show in Table 5 the average of the PSNRs and of the coefficient of variation (CoV) of the PSNRs calculated over 60-seconds intervals. The table contains the results for three scenarios: SVC without trimming, SVC with trimming and the case when only the base layer is decoded at the clients. The Max-Sum scheduler is employed at the base station. We considered different traffic loads and mobility scenarios. The results for the original streams indicate that the fluctuations in the video quality can be non-negligible: For the Manhattan mobility and low traffic load the CoV was 0.02 with the average PSNR of 35.66 dB, which correspond to the standard deviation of 0.7 dB over 60 second intervals. The CoV might be significantly higher at vehicular speeds. The simple trimming scheme reduces the CoV close to that of decoding the base layer only, while it keeps most of the gains of the enhancement layers in terms of mean PSNR. Based on this result we expect that a more sophisticated scheme would be able to efficiently control the fluctuations with a minimal loss in mean PSNR. Note that, at high traffic loads most of the enhancement packets are dropped at the scheduler and the fluctuations in PSNR are mostly due to the losses in the base layer. The fluctuations in the base layer are not addressed here and can not be eliminated by the trimming scheme.

Table 5. PSNR and the CoV of PSNR for the Max-Sum scheduler and the “trimming” scheme.

Mobility / Traffic load		Static		UDel		Manhattan	
		CoV (×10^3)	PSNR	CoV (×10^3)	PSNR	CoV (×10^3)	PSNR
Low (5.6 Mb/s)	original	7.25	36.07	16.86	35.94	20.67	35.66
	trimmed	0.17	34.71	3.94	34.20	4.62	33.67
	base layer	0.10	31.05	1.67	31.03	2.01	31.04
High (28 Mb/s)	original	1.12	31.24	7.44	31.18	12.11	31.19
	trimmed	0.47	31.19	6.42	31.13	10.01	31.06
	base layer	0.44	31.16	6.36	31.09	9.93	31.04

6. CONCLUSIONS

This work shows how low-complexity multicast scheduling can be combined with scalable video coding to improve the streaming performance in high speed mobile wireless networks. Our results indicate that the Max-Sum scheduler may provide good performance both in terms of the channel distortion and the fairness measured by the outage probability. We showed that thin base layers fare best in terms of system performance and outage probability. We quantified the benefits of scalable streaming compared to non-scalable streaming, and showed that the gains can be up to several dB under light load conditions. The gains are significant even if users employ a simple trimming scheme to decrease the quality fluctuations due to layering. We believe that the performance gains of scalable video multicast justify the additional scheduling complexity.

7. ACKNOWLEDGMENTS

This work was financed by Wireless@KTH.

8. REFERENCES

[1] E. Kaasinen, M. Kulju, T. Kivinen, and V. Oksman, “User acceptance of mobile TV services,” *Proc. Int. Conf. Human-Computer Interaction with Mobile Devices and Services* (*MobileHCI*), Bonn, Germany, Sept. 2009.

[2] S. Buchinger, S. Kriglstein, and H. Hlavacs, “A comprehensive view on user studies: survey and open issues for mobile TV,” *Proc. European Conf. Interactive TV and Video* (*EuroITV*), Leuven, Belgium, June 2009.

[3] A. Alexiou, C. Bouras, and V. Kokkinos, “Efficient assignment of multiple E-MBMS sessions towards LTE,” *IFIP Wireless and Mobile Networking Conference* (*WMNC*), Gdańsk, Poland, Sept. 2009.

[4] V. Vukadinovic and J. Huschke, “Statistical multiplexing gains of H.264/AVC video in E-MBMS,” *Proc. IEEE Int. Symp. Wireless Pervasive Computing* (*ISWPC*), Santorini, Greece, May 2008.

[5] 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 Commun. Magazine*, vol. 38, pp. 70–78, July 2000.

[6] S. Shakkottai and R. Srikant, “Scheduling real-time traffic with deadlines over a wireless channel,” *Wireless Networks*, vol. 8, no. 1, pp. 13–26, 2002.

[7] R. S. Tupelly, J. Zhang, and E. K. P. Chong, “Opportunistic scheduling for streaming video in wireless networks,” *Proc. 37th Annual Conference. Info. Sciences and Systems* (*CISS*), Baltimore, MD, March 2003.

[8] P. Ameiqeiras, J. Wigard and P. Mogensen, “Performance of the M-LWDF scheduling algorithm for streaming services in HSDPA,” *Proc. IEEE Vehicular Technology Conference* (*VTC*), Los Angeles, CA, Sept. 2004.

[9] K. Elsayed and A. Khattab, “Channel-aware earliest deadline due fair scheduling for wireless multimedia networks,” *Springer/Kluwer Journal on Wireless Personal Communications*, vol. 38, pp 233–252, July 2006.

[10] J. Tang, L. Zhang, and D. Siew, “An Opportunistic Video Scheduling Algorithm over Shared Wireless Downlink”, *Elsevier Computer Comm.*, vol. 29, no. 11, July 2006, pp. 1917–1926.

[11] P.V. Pahalawatta, R. Berry, T.N. Pappas, and A.K. Katsaggelos, “Content-Aware Resource Allocation and Packet Scheduling for Video Transmission Over Wireless Networks,” *IEEE Journal Sel. Areas Comm.*, vol. 25, no. 4, May 2007, pp. 749–759.

[12] X. Ji, J. Huang, M. Chiang, G. Lafruit, and F. Catthoor, “Scheduling and resource allocation for SVC streaming over OFDM downlink systems,” to appear in *IEEE Trans. Circuits and Systems for Video Tech.*, 2009.

[13] E. Maani, P.V. Pahalawatta, R. Berry, T.N. Pappas, and A.K. Katsaggelos, “Scalable video coding and packet scheduling for multiuser video transmission over wireless networks,” *Proc. SPIE Optics+Photonics*, San Diego, CA, July 2009.

[14] V. Vukadinovic and G. Karlsson, “Multicast Scheduling with Resource Fairness Constraints,” *ACM/Springer Wireless Networks*, vol. 15, no. 5, 2009, pp. 571–583.

[15] SVC Reference Software. [Online]. Available: http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm. [Accessed Sept. 1, 2009].

[16] D. Jurca, P. Frossard, and A. Jovanovic, “Forward error correction for multipath media streaming,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 19, no. 9, 2009, pp. 1315–1326.

[17] A. Stolyar, “On the Asymptotic Optimality of the Gradient Scheduling Algorithm for Multi-User Throughput Allocation,” *Operations Research*, vol. 53, no. 1, pp. 12–25, 2005.

[18] R. Agrawal and V. Subramanian, “Optimality of Certain Channel-Aware Scheduling Policies,” *Proc. 40th Annual Allerton Conf. Commun., Control, and Comp.*, pp. 1532–1541, Monticello, IL, Oct. 2002.

[19] UDel Models. [Online]. Available: http://udelmodels.eecis.udel.edu. [Accessed Jan. 20, 2009].

[20] J. Kim, V. Sridhara, and S. Bohacek, “Realistic mobility simulation of urban mesh networks,” *Elsevir Ad Hoc Networks*, vol. 7, no. 2, pp. 411–430, March 2009.

[21] M. Gudmundson, “Correlation model for shadowing fading in mobile radio systems,” *IEEE Electronic Letters*, vol. 27, pp. 2145–2146, Nov. 1991.

[22] F. Brouwer, I. de Bruin, J. Carlos Silva, N. Souto, F. Cercas, and A. Correia, “Usage of link-level performance indicators for HSDPA network-level simulations in E-UMTS,” *Proc. Int. Symp. Spread Spectrum Techniques and Applications*, Sydney, Australia, Sept. 2004.

[23] O. Verscheure, P. Frossard, and M. Hamdi, “User-oriented QoS analysis in MPEG-2 video delivery,” *Journal of Real-Time Imaging*, vol. 5, no. 5, pp. 305–314, Oct. 1999.