THE VIRTUE OF PATIENCE WHEN SCHEDULING MEDIA IN PRESENCE OF FEEDBACK

Christophe De Vleeschouwer and Pascal Frossard
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute - LTS4, CH-1015 Lausanne

ABSTRACT

We consider streaming of pre-encoded and packetized media over best-effort networks in presence of acknowledgment feedback. Given an estimation of future transmission resources and knowing about past transmissions and received acknowledgments, a scheduling algorithm is defined as a mechanism that selects the data to send over the network at any given time, so as to minimize the end-to-end distortion. Our work first reveals the sub-optimality of popular greedy schedulers, which might be strongly penalized by anticipated retransmissions. It then proposes an original scheduling algorithm that avoids premature retransmissions, while preserving the simplicity of the greedy paradigm. The proposed patient greedy (PG) scheduler appears to save up to 50% of rate in comparison with the conventional greedy approach.

1. INTRODUCTION

The proliferation of high-bandwidth and wireless Internet connections has increased the demand for a low-cost and flexible access to media content. Yet, to become a reality, widespread media dissemination has still to face the lack of guarantee offered by the network in terms of bandwidth, delay and error rates. Our work addresses the problem of streaming packetized media over a best-effort packet network. Sender-driven (re)transmission using acknowledgement (ACK) feedback is considered. For arbitrary packetization of encoded media content, our paper targets the definition of appropriate *scheduling* methods to decide which packet should be forwarded to the client at any given time. Most previous works about scheduling end-up in recommending the implementation of a greedy mechanism to match the instantaneous rate of a connection, while approximating some rate-distortion optimal, generally computationally intractable, solution. Our paper demonstrates that the greedy approach is sometimes far from optimal, and defines a novel and original patient greedy (PG) scheduler. PG outperforms the conventional greedy approach, while preserving its low computational complexity. The paper is organized as follows. Section 2 formalizes the streaming system, summarizes earlier contributions, and demonstrates greedy scheduling sub-optimality. Section 3 and 4 respectively define and validate our proposed PG scheduler. Section 5 concludes.

2. RATE-DISTORTION OPTIMIZED STREAMING

2.1. Media and channel models

To formalize the streaming framework, we follow [1]. The media source is assumed to have been encoded and packetized into a finite set of data units, stored on a media server. The interdependency between the data units is expressed by a direct acyclic graph, which induces a partial order relation $\prec$ among the data units. We write $l' \prec l$ when data unit l can only be decoded if data unit l' has been decoded. We say that data unit l' (l) is an ancestor (descendant) of data unit l (l'). The l^{th} data unit is characterized by its size S_l in bytes, its importance ΔD_l in units of distortion, and its delivery deadline $t_{D,l}$. The gain ΔD_l in distortion is the amount by which the distortion is decreased if data unit l is decoded, compared to the distortion if only the ancestors of l are decoded. When the streaming server selects a data unit for transmission, the data unit is encapsulated into a packet and sent over the network. A data unit can be encapsulated in more than one packet i.e., retransmissions are possible, but we assume that a packet contains one and only one data unit. As in [1], the network forwarding path is modeled as an independent time-invariant packet erasure channel with random delays. That means that a packet sent at time t can be either lost with probability ε_F, independent of t, or received at time t', where the delay $\tau_F = t' - t$ is randomly drawn with probability density function p_F. Similarly, when an acknowledgment packet is sent from the client to the server through the backward channel, it is either lost with probability ε_B, or received after a delay τ_B, drawn with probability density function p_B. Each forward or backward packet is lost or delayed independently of other packets. For convenience, to combine the packet loss probability and the packet delay density into a single probability measure, we define a forward (backward) trip time random variable, denoted FTT (BTT), that is assigned to ∞ when the packet is lost, and is set to τ_F (τ_B) when the packet is not lost. The round trip time RTT is a random variable defined as the sum of FTT and BTT.

2.2. Ode to greedy scheduling

Numerous authors have addressed the problem of scheduling media content over unreliable networks. Most of them have proposed to control streaming systems based on rate-distortion optimization techniques. Essentially, the authors in [1, 2, 3] formalize the scheduling decision as a partially observable Markov decision process. However, in final, to reduce computational complexity and to match the instantaneous rate imposed by the network, all these works recommend the use of heuristic greedy scheduling mechanisms that transmit, at any given time, the data unit that maximizes the decrease in distortion expected per unit of rate. As an example, to control the instantaneous rate of the streaming system, [1] proposes to adjust the scheduling parameters so that exactly one data unit is selected at each transmission opportunity offered by the network. In that case, the scheduler derived based on the finite horizon Markov process degenerates to a greedy approach. Other examples of studies that recommend the use of greedy mechanisms are [4] and [5]. Formally, the greedy scheduling mechanism advertised by all these authors is defined as follows. Let t and τ respectively denote the current time and the maximal pre-fetching delay. The set of data units whose delivery deadline lies between t and $t+\tau$ is denoted Γ_τ^t. At time t, when a packet has to be sent over the network, the greedy approach selects the data unit in Γ_τ^t that maximizes the expected decrease in distortion per unit of rate. Let $\varkappa_l^t$ denote the transmission history for the l^{th} data unit at time t. Specifically, $\varkappa_l^t = \{t_l^1, t_l^2, ..., t_l^{N_l}\}$ defines the N_l time instants at which the l^{th} data unit has been transmitted in the past. We now estimate the probability $p_c^t(l \mid \varkappa_l^t)$ for the l^{th} data unit to be received in-time at the client, knowing about its transmission history $\varkappa_l^t$. When an acknowledgment has been received for data unit l,

$p_c^t(l \mid \varkappa_l^t)$ is obviously equal to 1. In absence of acknowledgment for l, we note that data unit l only fails to reach the client in-time when all its transmission attempts fail. Because we assume independent packet transmissions, in absence of ACK for l, we have thus

$$p_c^t(l \mid \varkappa_l^t) = 1 - \prod_{i \leq N_l} P\{FTT_i^l > t_{D,l} - t_l^i \mid RTT_i^l > t - t_l^i\} \quad (1)$$

where FTT_i^l and RTT_i^l respectively denote the forward and round trip time random variables associated to the i^{th} (re)transmission of the l^{th} data unit. These random variables have the same distribution as the RTT and FTT variables defined in Section 1. We now estimate the decrease in distortion β_l^t to expect at time t from an additional transmission of the l^{th} data unit. Taking the dependency among data units into account, we have

$$\beta_l^t = \left[p_c^t(l \mid \varkappa_l^t \cup \{t\}) - p_c^t(l \mid \varkappa_l^t)\right] \times \sum_{l' \succeq l} \left(\Delta D_{l'} \prod_{l'' \preceq l', l'' \neq l} p_c^t(l'' \mid \varkappa_{l''}^t) \right) \quad (2)$$

The sum in Equation (2) reflects the fact that the reception of the l^{th} data unit is beneficial for l but also for all its descendants. The product in Equation (2) expresses the fact that correct decoding of data unit l' is subject to in-time reception of all its ancestors. At current time t, the greedy approach (re)transmit the data unit, denoted $l^G(t)$, that maximizes the gain in distortion to expect per unit of rate. We have thus

$$l^G(t) = \underset{l \in \Gamma_\tau^t}{\operatorname{argmin}} \frac{\beta_l^t}{S_l} \quad (3)$$

2.3. Greedy sub-optimality

This section analyzes the limitations of greedy scheduling in a toy example. For that purpose, we restrict our study to a specific content and propose a scheduling algorithm that is dedicated to that particular kind of content, but that is expected to be optimal in the rate-distortion (RD) sense. A comparative analysis of the greedy and proposed RD optimal approaches lays the ground for the definition of the patient greedy algorithm proposed in Section 3.

Due to the lack of space, and because it is just an intermediate step towards Section 3, we only give the flavour of the studied RD optimal scheduling system. We first define the format of the content handled by the RD system. In short, the chosen content is a sequence of identical, independent, and temporally equidistant frames. Each frame is composed of data units organized in a hierarchy of layers. All data units have the same size, and the decrease in distortion associated to a data unit only depends on its layer index. We now briefly summarize the algorithm proposed to stream that kind of content in a rate-distortion optimal way. For that purpose, similar to [1], we associate the notion of transmission opportunities and transmission policy to a data unit. The transmission opportunities refer to the set of time instants at which a data unit may be put into a packet and transmitted. A policy is then defined to assign actions to observations at future transmission opportunities. Concretely, it tells whether the data unit should be retransmitted or not given the acknowledgement feedback(s) received about itself and possibly some other dependent data units. Based on these definitions, the guiding principle of the RD system states that the sender should commit to follow a pre-defined transmission policy for each data unit sent over the network. As a consequence of the commitment principle, the scheduler prioritizes the retransmission of committed data, and only sends new data over the network if none of the committed data needs to be retransmitted. New data are selected in increasing order of deadline within a layer. To define the layer index and the transmission policy of the new data, we make the assumption that all future data of a given layer commit to the same policy (which only makes sense when the media is composed of identical frames in size and distortion). We then select the data and the policy of each layer so as to converge to an optimal equilibrium that is characterized by:

- the amount of committed data units allocated to each layer. A large number of committed data for one layer means that a large period of time is available before expiration of the delivery deadline for the data sent in that layer. As a consequence, everything being equal, it allows for more efficient retransmission mechanisms, which in turns improves streaming performance.
- the policy associated to each layer. The retransmissions scheduled by these policies affect the quality in two opposite ways. First, more retransmissions increase the probability that the data arrives in-time at the client, which improves the rendered quality. Second, too many retransmissions w.r.t. the available rate progressively drain the buffer of committed data (because prioritized retransmissions prevent the transmission of new data). It makes future retransmissions less efficient, and degrades the quality.

A complete and formal description of the method used to converge to a rate-distortion optimal equilibrium is planned for an upcoming report. Here, we just compare the behaviors of the RD optimal and greedy (G) schedulers to derive appropriate and of-general-use heuristics to improve the greedy mechanism. Table 1 compares the statistics of both G and RD. The streamed frames are displayed every 50 ms, and are composed of 5 layers, defined such that $S_{l+1} = S_l$ and $\Delta D_{l+1} = \Delta D_l/2 \; \forall l \leq 5$ (see $R21$ template in Section 4). The channel delay pdf is modeled as shifted exponential with mean μ_F and μ_B. Table 1 presents (i) the percentage of data units that have reached the client in-time, (ii) the average number of transmissions per data unit, and (iii) the average number of unnecessary retransmissions, for which an acknowledgment triggered by a previous transmission was on the way to reach the sender before data delivery deadline. For both G and RD, these statistics are presented as a function of the layer index. We observe that the greedy algorithm fails to transmit the fifth layer because it retransmits too much other layers. Based on the last column in Table 1, we conclude that lots of these retransmissions would be avoided if the sender was more patient in triggering retransmissions, so as to give previous ACKs the opportunity to reach the sender. This observation is fundamental, and motivates the definition of the patient greedy algorithm in Section 3.

Layer	% in-time		Average # of trans.		Average # of trans. while ACK to come	
	RD	G	RD	G	RD	G
1	100	100	1.42	2.56	0.09	1.27
2	100	100	1.39	2.19	0.08	0.91
3	100	100	1.37	1.36	0.07	0.15
4	100	38	1.26	1.03	0.02	0.02
5	96	0	1.08	0	0.00	-

Table 1. Statistical comparison of RD optimal and greedy (G) scheduling mechanisms. $\mu_{F,B} = 180ms$, $\varepsilon_F = 0.2$, and $\varepsilon_B = 0$.

3. OUR PROPOSAL: PATIENT GREEDY SCHEDULING

We have shown in Section 2.3 that greedy solutions might result in significantly suboptimal rate-distortion (RD) trade-offs. Nevertheless, the RD system studied in Section 2.3 to highlight the lim-

itations of greedy approaches is of little practical interest because it relies on specific assumptions about the media content. So, the primary goal of this section is to derive a scheduling approach that can be used for any media content while integrating the lessons drawn based on the complex and dedicated system described in Section 2.3. In short, Table 1 suggests that the greedy scheduler should wait longer between successive retransmissions, so that the ACKs triggered by previous transmissions of the same data unit do have the opportunity to reach the sender. This learning is supposed to stay valid in general, for any kind of content. For this reason, we evaluate the advantage to get from a postponed retransmission, and propose to constrain the conventional greedy algorithm to forbid the retransmission of data units for which a delayed retransmission is likely to bring a benefit in the rate-distortion sense. The proposed solution, named Patient Greedy (PG) algorithm, preserves the simplicity offered by the conventional greedy scheduler while significantly improving its performances.

3.1. Consequences of a delayed transmission

We now explicitly consider the possibility to wait before the retransmission of a data unit. Let $\beta_l^{t,t'}$ denote the expected decrease in distortion estimated at current time t for the (re)transmission of the l^{th} data unit at time $t' \geq t$. Similar to (2), we have

$$\beta_l^{t,t'} = \left[p_c^t(l \mid \varkappa_l^t \cup \{t'\}) - p_c^t(l \mid \varkappa_l^t)\right] \times \sum_{l' \succeq l} \left(\Delta D_{l'} \prod_{l'' \preceq l', l'' \neq l} p_c^t(l'' \mid \varkappa_{l''}^t) \right) \quad (4)$$

In the right hand side of (4), only $p_c^t(l \mid \varkappa_l^t \cup \{t'\})$ depends on t'. Based on (1), in absence of ACK for data unit l in t, we have

$$p_c^t(l \mid \varkappa_l^t \cup \{t'\}) = 1 - P\{FTT_{N_l+1}^l > t_{D,l} - t'\} \times \prod_{i \leq N_l} P\{FTT_i^l > t_{D,l} - t_l^i \mid RTT_i^l > t - t_l^i\} \quad (5)$$

which shows that $p_c^t(l \mid \varkappa_l^t \cup \{t'\})$, and consequently the benefit in distortion $\beta_l^{t,t'}$, decrease as t' increases. Furthermore, because an ACK might be received between t and t', the cost in rate associated to a postponed transmission also decreases as t' increases. Formally, we introduce the expected cost $\zeta_l^{t,t'}$ estimated in t and associated to the transmission of the l^{th} data unit at time $t' \geq t$. Given the transmission history $\{t_l^i\}_{i \leq N_l}$ of data unit l, we have

$$\zeta_l^{t,t'} = S_l \prod_{i \leq N_l} P\{RTT_i^l > t' - t_l^i \mid RTT_i^l > t - t_l^i\} \quad (6)$$

3.2. Patient greedy algorithm

Based on Section 3.1, we know that postponing the retransmission of the l^{th} data unit has a positive impact on the rate consumption i.e., $\zeta_l^{t,t'}$ decreases as t' increases, but a negative impact on the media quality i.e., $\beta_l^{t,t'}$ decreases when t' increases. To estimate whether the gain in rate is worth the loss in quality, we introduce the Lagrangian factor $\lambda(t)$, which balances the expected gain in rate versus distortion. In concrete words, $\lambda(t)$ defines the decrease in distortion that can be expected per additional unit of rate at time t. Given the Lagrangian factor $\lambda(t)$, we can figure out whether postponing the retransmission from t to t' is likely to bring a global benefit in the rate-distortion sense. For the l^{th} data unit, delaying transmission is beneficial when $\lambda(t)[\zeta_l^{t,t} - \zeta_l^{t,t'}] > \beta_l^{t,t} - \beta_l^{t,t'}$ or, equivalently, when

$$-\beta_l^{t,t} + \lambda(t)\zeta_l^{t,t} > -\beta_l^{t,t'} + \lambda(t)\zeta_l^{t,t'} \quad (7)$$

Based on (7), we say that a data unit is *eligible* for transmission at current time t if there is no global RD benefit to expect from a postponed transmission. Formally, the l^{th} data unit is eligible at time t if

$$t = \underset{t' \in [t, t_{D,l}]}{\operatorname{argmin}} \left(-\beta_l^{t,t'} + \lambda(t)\zeta_l^{t,t'} \right) \quad (8)$$

We can now define our proposed Patient Greedy (PG) scheduling mechanism as a greedy scheduling that is constrained to select the data to transmit among the set of eligible data units. Formally, let Ψ_τ^t denote the set of eligible data units contained in Γ_τ^t, and let $l^{PG}(t)$ denote the index of the data unit selected at time t by the PG algorithm. By definition $\beta_l^{t,t} = \beta_l^t$ and $\zeta_l^{t,t} = S_l$. So, similar to (3), we have

$$l^{PG}(t) \underset{l \in \Psi_\tau^t}{\operatorname{argmin}} \frac{\beta_l^t}{S_l} \quad (9)$$

3.3. Practical implementation considerations

This section explains (i) how to estimate $\lambda(t)$, and (ii) how the eligibility condition defined in (8) is checked in practice.

Regarding the Lagrangian factor $\lambda(t)$, we observe that, everything being equal, the rate spared in postponing the retransmission of a data unit is used to transmit one or several additional data unit(s). Following the greedy approach principle, these additional data units are selected as the ones that are expected to bring the largest benefit per unit of rate among the data units that have not been transmitted yet. We can thus estimate the factor $\lambda(t)$ based on the history of the streaming session. Specifically, $\lambda(t)$ is estimated as the smallest expected benefit per unit of rate observed among the data units that have been sent over the network in a recent past. In practice, $\lambda(t)$ is defined to be a piecewise constant function i.e., it is updated at regular time intervals. Let $\{\upsilon_k\}_{k \geq 0}$ denote the sequence of time instants at which $\lambda(t)$ is updated, and let υ_k^- and υ_k^+ denote the instants immediately preceding and following υ_k. We also define λ_k to be the smallest expected benefit per unit of rate encountered among the data units sent during the $[\upsilon_{k-1}, \upsilon_k]$ time interval. The piecewise function $\lambda(t)$ is then derived based on the sequence $\{\lambda_k\}_{k>0}$. Formally, starting with an initial value of $\lambda(\upsilon_0)$ equal to zero, we update the Lagrangian factor based on a weighted exponential average. We have

$$\lambda(\upsilon_k^+) = \alpha\lambda_k + (1-\alpha)\lambda(\upsilon_k^-) \quad \forall k > 0 \quad (10)$$

To complete (10), we still have to define the parameter α and the sequence of time instants $\{\upsilon_k\}_{k \geq 0}$ at which $\lambda(t)$ is updated. For that purpose, we introduce the notion of self-contained group of interdependent data units. A self-contained group is defined so that it does not have any ancestor or descendant among data units that are outside the group. Typically, it corresponds to a group of pictures in the MPEG terminology, or to a frame in the J2K terminology. We propose to update $\lambda(t)$ each time a self-contained group becomes obsolete i.e., when all data units contained in the group have passed their delivery deadlines. We have chosen to synchronize the $\{\upsilon_k\}_{k \geq 0}$ sequence with the delivery deadlines of self-contained groups because they occur at regular time intervals, and because we migh expect some consistency between the smallest expected benefit per unit of rate observed in such intervals. In our simulations, the parameter α has been chosen equal to 0.4, but the function $\lambda(t)$ appears to be quite insensitive to the α parameter because successive values of λ_k are indeed close to each other.

To verify the eligibility condition defined by (8), only a finite number of t' values are considered. The computational complexity associated to each additional t' is small as $\beta_l^{t,t'}$ and $\zeta_l^{t,t'}$ can be computed without referring to ancestors and descendants of l. In practice, the possible values of t' involved in the verification

of (8) are distributed regularly between the current time t and the data unit delivery deadline $t_{D,l}$. We have chosen to use the average time elapsed between successive packet transmissions in the recent past as the interval between two successive investigated t' values. Doing so, we roughly investigate all realistic transmission alternatives. It is worth noting that the scheduling system is not sensitive to the sampling period of t'. Indeed, the postponed transmission alternatives are investigated to check whether waiting before retransmission is worthwhile or not. The purpose is thus not to find the exact time t' that minimizes $-\beta_l^{t,t'} + \lambda(t)\zeta_l^{t,t'}$.

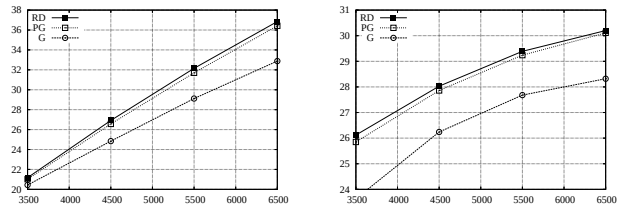
4. SIMULATION RESULTS

This section demonstrates that the proposed patient greedy algorithm outperforms conventional greedy solutions. Forward (F) and backward (B) paths are modeled as independent time-invariant packet erasure channels with random delays (see Section 2.1) and constant bandwidth. The probability density functions p_F and p_B are modeled as a shifted exponential with mean μ_F (μ_B) and shift $\kappa_F = \mu_F/2$ ($\kappa_B = \mu_B/2$)[1, 5]. Furthermore, our preliminary simulations consider the streaming of data units corresponding to a sequence of identical and temporally equidistant frames that are decoded independently of each others. Streaming a strictly formatted content provides two major advantages. First, it makes the results easy to reproduce and compare with other contributions. It also facilitates the interpretation and understanding of the scheduling mechanisms as they are not affected by fluctuation of the media features along the time. Second, it allows for a comparison with the RD optimal sheduling mechanism described in Section 2.3. We are thus able to estimate how far the greedy and patient greedy schedulings are from an optimal. Whilst being helpful to understand the scheduling behavior, formatted media content is not fully sufficient to apprehend all the components of real-life streaming system. For this reason, experiments that are based on real video content will be reported in an upcoming publication.

We now present some of our initial results. In these simulations, the frame rate is 20 fps, and all (patient) greedy approaches use a maximal pre-fetching delay τ equal to 1 sec. Each frame is composed of N = 5 data units organized in a hierarchy of layers. All data units have the same size, set to 1000 bits. The increase in quality (or equivalently the decrease in distortion) associated to a data unit only depends on its layer index, and obeys a predefined distortion template, characterized by a constant ratio between the decrease in distortion provided by consecutive layers. Let ΔD_l denote the decrease in distortion for the l^{th} layer. We denote $R11$ the template for which $\Delta D_1 = 8$ and $\Delta D_{l+1} = \Delta D_l$. We denote $R21$ ($R12$) the template for which $\Delta D_1 = 32$ ($\Delta D_1 = 1$) and $\Delta D_{l+1} = \Delta D_{l+1}/2$ ($\Delta D_{l+1} = 2\Delta D_{l+1}$). For all templates the quality achieved in absence of any data unit is set to 0. Note that the $R11$ and $R21$ templates are the most realistic, as media coders generally encode the most important information in the first layers.

Figure 1 presents the quality as a function of the forward channel bitrate for the greedy (G), the patient greedy (PG), and the rate-distortion optimal (RD) algorithms. Figure 1 (a) and (b) respectively consider the $R11$ and $R21$ distortion templates. We observe that PG significantly outperforms G, and achieves performances that are close to the RD ones. Extended simulations with a large range of channel parameters have confirmed that observation.

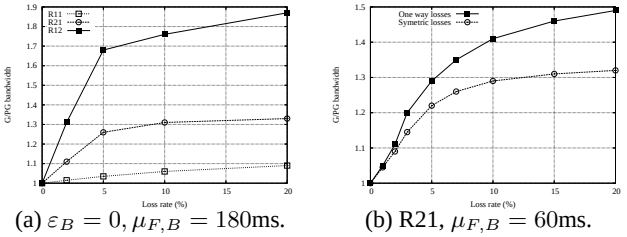
To evaluate the gain of PG in terms of rate consumption, Figure 2 plots the ratio between the rates consumed by G and PG to achieve the same average quality as a function of ε_F. The targeted



(a) R11. (b) R21.

Fig. 1. Quality (in units of quality) versus channel bitrate (bits/sec). $\mu_F = \mu_B = 180ms$, $\varepsilon_F = 0.2$, and $\varepsilon_B = 0$.

quality is the one obtained by PG at 4.5 kbits/sec. Figure 2 (a) considers a lossless backward channel, and compares the G and PG for different layer distortion templates. We observe that the amount of transmission rate saved by the PG approach is highly dependent on the way quality is allocated among layers. Some distortion templates end-up in relatively good behavior of the greedy algorithm, while other cause a lot of penalizing anticipated retransmissions. Figure 2 (b) analyzes the impact of feedback reliability on the gain provided by the PG algorithm. Losses are either symmetric (ε_B = ε_F) or one-way (ε_B = 0). We conclude that PG is even more beneficial with reliable feedback, which makes sense as the main PG achievement is a better usage of received ACKs.



(a) $\varepsilon_B = 0, \mu_{F,B} = 180$ms. (b) R21, $\mu_{F,B} = 60$ms.

Fig. 2. Ratio between the G and PG transmission rates needed to achieve a constant quality, as a function of ε_F.

5. CONCLUSIONS

The paper reveals the sub-optimality of popular greedy schedulers, which are penalized by premature retransmissions. The proposed patient greedy (PG) solution solves the problem, while preserving the simplicity of greedy approaches. PG appears to be close to optimal in the rate-distortion sense. At constant quality, PG saves up to 50% of rate in comparison with conventional greedy schedulers.

6. REFERENCES

[1] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Microsoft reasearch technical report, MSR-TR-2001-35.*

[2] Danjue Li, Gene Cheung, Chen-Nee Chuah, and S.J. Ben Yoo, "Joint server/peer receiver-driven rate-distortion optimized video streaming using asynchronous clocks," in *ICIP*, Singapore, October 2004.

[3] D. Tian, X. Li, G. Al-Regib, Y. Altunbasak, and J.R. Jakson, "Optimal packet scheduling for wireless video streaming with error-prone feedback," in *IEEE WCNC*, Atlanta, March 2004.

[4] Z. Miao and A. Ortega, "Optimal scheduling for the streaming of scalable media," in *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 2000.

[5] C.-L. Chang, S. Han, and B. Girod, "Rate-distortion optimized streaming for 3-D wavelet video," in *ICIP*, Singapore, October 2004.