



**HAL**  
open science

## Adaptive Scalable Layer Filtering Process For video scheduling over wireless networks based on MAC buffer management

Nesrine Changuel, Nicholas Mastronarde, Mihaela van Der Schaar, Bessem Sayadi, Michel Kieffer

### ► To cite this version:

Nesrine Changuel, Nicholas Mastronarde, Mihaela van Der Schaar, Bessem Sayadi, Michel Kieffer. Adaptive Scalable Layer Filtering Process For video scheduling over wireless networks based on MAC buffer management. IEEE International Conference on Acoustics, Speech, and Signal Processing, May 2011, Prague, Czech Republic. pp.2352-2355, 10.1109/ICASSP.2011.5946955 . inria-00614567

**HAL Id: inria-00614567**

**<https://inria.hal.science/inria-00614567>**

Submitted on 12 Aug 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# ADAPTIVE SCALABLE LAYER FILTERING PROCESS FOR VIDEO SCHEDULING OVER WIRELESS NETWORKS BASED ON MAC BUFFER MANAGEMENT

Nesrine Changuel\* Nicolas Mastronarde<sup>†</sup> Mihaela Van der Shaar<sup>†</sup> Bessem Sayadi\* Michel Kieffer<sup>+,o</sup>

\* Alcatel-Lucent - Bell-Labs, France

<sup>†</sup> UCLA, 420 Westwood Plaza, Los Angeles, USA

<sup>+</sup> L2S, CNRS - SUPELEC - Univ Paris-Sud, F-91192 Gif-sur-Yvette

<sup>o</sup> LTCI, CNRS-Télécom ParisTech, F-75013 Paris

## ABSTRACT

In this paper, the problem of scalable video delivery over a time-varying wireless channel is considered. Packet scheduling and buffer management in both Application and Medium Access Control (MAC) layers are jointly considered. Various levels of knowledge of the state of the channel are considered. The control is performed via scalable layer filtering (some scalability layers may be dropped). In all cases, the problem is cast in the context of Markov Decision Processes which allows the design of foresighted policies maximizing some long-term reward. Without channel state observation, the control has to rely on the observation of the level of the MAC buffer only. Experimental results show that even with a lack of knowledge of the channel state, the foresighted control policy provides only a moderate loss in received video quality.

## 1. INTRODUCTION

The increase of available bandwidth in wireless networks allows a larger diversity of services provided to users. Mobile television, video on demand (payed or via YouTube), telephony via data channel are applications that are currently emerging. Providing good-quality compressed data to mobile receivers with widespread large-display devices, such as smart-phones or tablets is becoming more and more difficult. Apart from the bandwidth scarcity, due to an increased number of users, multimedia delivery to wireless receivers is particularly challenging to the time-varying characteristics of wireless channels (error-rate and bandwidth) and to delivery delay constraints of some applications such as mobile television or telephony.

Scalable coders, such as H.264/SVC [1] for video, have been developed to partially address these issues. Packets with various priority levels are generated. Receiving only the highest-priority packets allows to get an acceptable quality, which increases with the number of lower-quality packets correctly received. One of the main drawback of this approach is that any low-priority packet is useless unless all associated higher-priority packets have been received. Therefore, an efficient packet scheduling scheme has to be designed either in unicast or in multicast scenarios.

This paper focuses on Quality of Service (QoS) optimization in the context of scalable video streaming applications to mobile users. When considering unicast applications, various types of feedback from the receiver may be obtained, for example at the application layer via RTCP feedback [2] to get information about the level of buffers at application layer or at MAC layer via HARQ ACK/NACK [3] to get information about the channel conditions.

Using information fed back to the controller, [4] have designed scheduling schemes for time-varying channel conditions. In [4], a

cross-layer scheduling architecture for video transmission is presented; the time-varying characteristics of the wireless channel are modeled by a discrete-time Markov model, which state is fed back to the controller. Scalable video coders have been considered in [5, 6], focusing on the application layer. A Markov Decision Process (MDP) framework [7] is considered to design a long-term control policy accounting for the level of the playback buffer. In [8, 9, 10] and [11], focus is more on the MAC layer, since these papers address buffer management problems within the Radio Access Network (RAN). Application packets may be dropped depending on their priority and on the level of the buffers at the MAC layer.

The main difficulty when designing such control schemes is that feedback comes with delay. This delay may be of the order of tens to hundreds of milliseconds for HARQ ACK/NACK messages to one or several seconds for RTCP packets, which may cause stability problems. Our aim in this paper is to design a cross-layer control mechanisms in an MDP framework [12, 13] to control the level of buffers at MAC and at Application layers at the transmitter side *only* of a communication chain. Feedback at MAC layer is implicitly used, but no feedback at Application layer is considered, avoiding the use of delayed measurements. The considered scalable video transmission scheme is presented in Section 2. The control is performed by a filtering of the scalability layers of the encoded video. A Markov model of the channel is considered and three hypotheses concerning the knowledge of the state of the channel are considered, namely, known state, known state with delay, and unknown state. Even in the last two cases, the design of a control law may be cast in a classical MDP framework, see Section 3. Experimental results show that when a foresighted policy is considered, the absence of knowledge of the channel state results in a moderate degradation of the behavior of the system, thanks to the availability of the MAC buffer state, which provides a reasonable estimate of the state of the channel, see Section 4.

## 2. SYSTEM

The considered single user unicast streaming system is sketched in Figure 1. The core network consists of a streaming server, hosting a scalable video coder, a *proxy*, and a base station. Packets are transmitted through a wireless channel and received by a mobile client. Among the components of the base station (Node B), this paper considers mainly the MAC buffer. The MAC scheduler of Node B, as well as its physical layer, its radio front-end, the wireless channel, the physical layer of the receiver, and the part of the MAC layer at receiver side managing ACK/NACK procedures are denoted in the remainder of the paper as the *channel*. No RTCP feedback is considered in what follows. Only feedback provided by the ACK/NACK

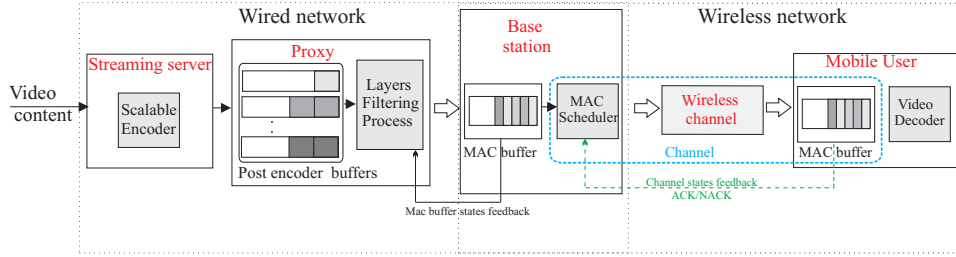


Fig. 1. End-to-end video streaming scheme

may be exploited to derive the state of the channel.

### 2.1. System description

In the streaming server, the video sequence is segmented into frames encoded into a base layer and a set of  $L - 1$  enhancement layers. Access Units (AU) represent basic processing units, macroblock(s), slices, or frame(s), consisting of the base layer and its corresponding enhancement layers. Here an AU corresponds to a single frame identified by its time index  $t$ . AU are generated with a constant period of time  $\Delta t$ . An H.264/SVC Medium-Grain quality Scalable (MGS) coding scheme [14] is considered in this paper. Its encoding parameters (quantization steps, frame rate, etc.) are controlled by the streaming server (not by the controller designed in this paper), independently of the remainder of the transmission chain. Moreover, a *base layer only* control scheme [1] for the encoder is adopted to minimize the drift when refinement layers are lost. Each SNR layer of each encoded frame is packetized into RTP/UDP/IP packets, which are delivered via an over-provisioned core network (assumed lossless) to  $L$  post-encoder buffers (one per layer) situated in the proxy.

The controller performs *layer filtering* within the proxy: for each layer, packets may be sent, kept, or dropped. Sent packets are fed to the MAC buffer of the base station after being segmented into Packet Data Units (PDUs) of constant size. PDUs are then transmitted to the mobile receiver, which stores correctly received PDUs in its own MAC buffer. Packet de-encapsulation and buffering in one of the  $L$  buffers at application layer is done as soon as all corresponding PDUs have been received. Complete or incomplete AU are then processed by the video decoder. Outdated packets are dropped, without being decoded. When one of the post-encoder, MAC, or receiver buffers is full, packets in the queue are dropped in a head-of-line order.

### 2.2. System requirements and constraints

Ideally, the layer filtering process should be designed to maximize the QoS at receiver side. Some necessary conditions have to be satisfied to perform this task. First, and most important, is avoiding overflow of the MAC buffer to prevent PDUs from being dropped. Underflow should also be prevented to use the channel in an optimal way. Second, overflow of the post-encoder buffers should be avoided to limit the delay introduced by the system. Underflow is not satisfying too, especially at the base layer, since this indicates that too much importance has been given to the base layer compared to the other layers.

## 3. MODEL OF THE SYSTEM

The design of an efficient layer filtering process is translated in the framework of discrete-time MDP [7]. An MDP is a 4-tuple  $(\mathcal{S}, \mathcal{A}, P, R)$ , where  $\mathcal{S}$  is the set of states of the system,  $\mathcal{A}$  is the set of actions,  $P(s, s', a)$  is the transition probability from  $s \in \mathcal{S}$  at time  $t$  to  $s' \in \mathcal{S}$  at time  $t + 1$ , when the action  $a \in \mathcal{A}$  is applied

to the system. Finally  $R(s, s', a)$  indicates the immediate reward received after a transition from  $s$  to  $s'$  obtained by using action  $a$ . Some *policy*  $\pi(s) \in \mathcal{A}$ ,  $s \in \mathcal{S}$  maximizing the immediate reward (*myopic policy*) or a discounted sum of future rewards (*foresighted policy*) has then to be found.

### 3.1. States

The state of the controlled system consists of  $\mathbf{s}^e$  gathering the levels  $s_\ell^e$ ,  $\ell=1, \dots, L$  of the post-encoder buffers,  $s^m$  corresponding to the level of the MAC buffer in the base station, and  $h$  representing the channel state. The state of the system is thus  $\mathbf{s} = (\mathbf{s}^e, s^m, h)$ . Each components of the state vector is more clearly defined in what follows.

**Buffers.** The state of the  $\ell$ -th post-encoder buffer is  $s_\ell^e \in \mathcal{S}_\ell^e$  and the state of the MAC buffer is  $s^m \in \mathcal{S}^m$ . Various granularity levels may be considered to represent the content of a buffer: number of bits, number of packets, or even coarser description such as underflow/normal/overflow, see [13, 12] for more details. In this paper, the state of the post-encoder buffer will describe the amount of packets corresponding to parts of stored AU. This help to track the delay introduced within the system. The state of the MAC buffer correspond to the number of PDUs or of bits (since PDUs have all the same length).

**Channel model.** The channel state  $h_t$  describes the channel conditions (rate, probability of error, capacity, etc.) assumed constant between  $t - 1$  and  $t$ . The channel is modeled by a first-order Markov process with  $n$  states, with known transition probability  $p(h_{t+1}|h_t)$  and stationary probability  $p(h_t)$ . In this paper,  $h_t \in \mathcal{R}_c$  represents the rate available during the considered time slot, with  $\mathcal{R}_c = \{R_1, \dots, R_n\}$  bits/s. Three hypotheses concerning the knowledge of the state of the channel are considered. *Hyp. 1: instantaneous channel state*, where  $h_t$  is assumed available when choosing the action to apply between time  $t$  and  $t + 1$ ; this is realistic only when feedback with very short delay is possible. *Hyp. 2: delayed channel state*, where only  $h_{t-\delta}$  ( $\delta$  is a strictly positive integer) is assumed available when choosing the action to apply between  $t$  and  $t + 1$ ; represents a more realistic situation, where the channel state is only known with some delay  $\delta$ . *Hyp. 3: unknown channel state* which is a scenario where no channel state feedback is considered.

### 3.2. Actions

For each layer, the proxy may send, hold, or drop packets. The action  $a_{\ell,t} \in \mathcal{A}$  taken for the  $\ell$ -th layer between time  $t$  and  $t + 1$  represents the number of transmitted packets from the post-encoder buffer to the MAC buffer (when  $a_{\ell,t} > 0$ ), or the number of dropped packets (when  $a_{\ell,t} < 0$ ). If  $a_{\ell,t} = 0$ , packets are kept in the post-encoder buffer. The vector gathering all actions is  $\mathbf{a} = (a_1, \dots, a_L) \in \mathcal{A}^L$ .

### 3.3. Transition matrices and reward function

Once all states and actions have been identified, one has to determine the transition probability matrix  $P(s, s', a)$  has to be evaluated. For

that purpose, models of the lengths of packets delivered for each layer by the streaming server have to be considered first.

**Packet model.** The length of each encoded packet at each layer depends on the type of frame that has been encoded (I, P, or B), on the activity of the frame, and of the chosen encoding parameters. A Markov model has been proposed in [15] to capture the dependency between the length of consecutive packets. This model could be used in an MDP framework by supplementing the state with a state vector representing the length of packets stored in the buffers. Nevertheless, this would significantly increase the complexity of the problem. To limit the dimension of the problem, such state is not introduced and the lengths of packets stored in the  $\ell$ -th post-encoder buffer is described by the probability distribution  $q_\ell(\lambda)$ ,  $\ell = 1, \dots, L$  assumed to be known (it may be provided by the streaming server).

**Transition matrices.** Now, the transition probability matrix may be evaluated in the three hypotheses concerning the knowledge of the state of the channel described in Section 3.1.

*Hyp. 1:* In this case,  $h_t$  is available at time  $t$  when applying  $\mathbf{a}_t$  to the system. The state transition matrix then

$$P_1(\mathbf{s}_t, \mathbf{s}_{t+1}, \mathbf{a}_t) = \Pr(\mathbf{s}_{t+1}^e, s_{t+1}^m, h_{t+1} | \mathbf{s}_t^e, s_t^m, h_t, \mathbf{a}_t), \quad (1)$$

which may be easily evaluated using the fact that  $p(h_{t+1}|h_t)$  and  $q_\ell(\lambda)$ ,  $\ell = 1, \dots, L$  are known.

*Hyp. 2:* This case is a special class of partially delayed MDP. The current state of the channel is available without noise with a unit delay. In [16], the case  $\delta = 1$  was considered and the delayed MDP model is transformed into a standard fully observable MDP. Here, the situation is simpler, and one may obtain a new state transition matrix as follows

$$\begin{aligned} P_2(\mathbf{s}_t, \mathbf{s}_{t+1}, \mathbf{a}_t) &= \Pr(\mathbf{s}_{t+1}^e, s_{t+1}^m, h_t | \mathbf{s}_t^e, s_t^m, h_{t-1}, \mathbf{a}_t) \\ &= \sum_{h_{t+1}} \Pr(\mathbf{s}_{t+1}^e, s_{t+1}^m, h_{t+1} | \mathbf{s}_t^e, s_t^m, h_t, \mathbf{a}_t) \Pr(h_t | \mathbf{s}_t^e, s_t^m, h_{t-1}, \mathbf{a}_t) \\ &= \sum_{h_{t+1}} P_1(\mathbf{s}_t, \mathbf{s}_{t+1}, \mathbf{a}_t) p(h_t | h_{t-1}) \end{aligned} \quad (2)$$

The last line is obtained thanks to the fact that  $h_t$  is known,  $\mathbf{s}_t^e, s_t^m$ , or  $\mathbf{a}_t$  do not provide additional information on  $h_t$ . Thus  $P_2$  may be easily derived from  $P_1$ .

*Hyp. 3:* In this case, no channel state is available to the controller. The state transition matrix may then be written as

$$\begin{aligned} P_3(\mathbf{s}_t, \mathbf{s}_{t+1}, \mathbf{a}_t) &= \Pr(\mathbf{s}_{t+1}^e, s_{t+1}^m | \mathbf{s}_t^e, s_t^m, \mathbf{a}_t) \\ &= \sum_{h_{t+1}} \sum_{h_t} \Pr(\mathbf{s}_{t+1}^e, s_{t+1}^m, h_{t+1} | \mathbf{s}_t^e, s_t^m, h_t, \mathbf{a}_t) \Pr(h_t | \mathbf{s}_t^e, s_t^m, \mathbf{a}_t) \\ &= \sum_{h_{t+1}} \sum_{h_t} P_1(\mathbf{s}_t, \mathbf{s}_{t+1}, \mathbf{a}_t) p(h_t) \end{aligned} \quad (3)$$

since  $\mathbf{s}_t^e, s_t^m$ , or  $\mathbf{a}_t$  do not provide additional information on  $h_t$ .

In order to limit the size of these transition matrices, it is possible to quantize the values which may be taken by the states in a more or less coarse way to get a compromise between complexity and description accuracy.

#### Reward function.

Concerning the reward function, a time  $t$ , the layer filtering process chooses an action that maximizes the QoS at receiver side. One of the QoS to satisfy is the video quality. To maximize the received video quality, one may maximize the average Peak Signal to Noise Ratio (PSNR) of the decoded frames. However building a reward function using this quality metric is not easy due to the variability of

the delay between the time at which an AU is filtered and the time at which it is displayed. This time varying delay is linked to the time-varying channel conditions.

A alternative reward function is then built that penalize dropped packets, as well as buffer overflow and underflow, according to the constraints described in Section 2.2. The expected value of the reward is

$$\begin{aligned} R_t(\mathbf{s}_{t+1}, \mathbf{s}_t, \mathbf{a}_t) &= E \left[ \underbrace{\sum_{\ell=1}^L \gamma_\ell a_{\ell,t}}_{\text{transmission}} + \underbrace{\sum_{\ell=1}^L \lambda_\ell \rho_\ell(s_{\ell,t+1}^e, a_{\ell,t})}_{\text{post encoder buffer}} \right. \\ &\quad \left. + \underbrace{\beta \rho_2(s_{t+1}^m, a_{\ell,t})}_{\text{MAC buffer}} \right]. \end{aligned} \quad (4)$$

The positive parameters  $\gamma_\ell, \lambda_\ell, \beta$ , with  $\ell = 1 \dots L$ , trade off the importance of the various constraints. The reward function (4) involves several parts, the first linked to the number of transmitted SNR layers, the others to the post encoder and the MAC buffers constraints.

Assuming that increasing the amount of transmitted packets increases the received quality, the transmission reward should help to maximize the amount of transmitted packets. The parameters  $\gamma_\ell$  allow to give a higher priority to packets belonging to the base layer compared to those of the enhancement layers. For post encoder and MAC buffer constraints,  $\rho_1(\cdot)$  and  $\rho_2(\cdot)$  provide positive rewards for satisfying buffer states and negative rewards for states that should be avoided.

The proposed reward function depends on the actions and on the states of the post encoder and the MAC buffers. Assuming that these states do not depend on the state of channel, the three MDP problems would have the same reward function and different transition matrix.

#### 3.4. Myopic and foresighted policies

The policy  $\pi$  as a mapping from joint states to joint actions. In the considered system, the policy  $\pi$  simply indicates the number of SNR layers to transmit knowing the state of the post encoder buffer, of the MAC buffer, and possibly of (delayed) state of the channel. The optimal *foresighted* policy consists in finding the optimal stationary Markov policy  $\pi^*$  corresponding to the optimal state-value function defined as

$$V^*(\mathbf{s}_t) = \max_{\mathbf{a} \in \mathcal{A}} \sum_{k=0}^{\infty} \alpha^k R_{t+k+1} | \mathbf{s}_t, \quad (5)$$

where  $0 < \alpha < 1$  is the discount factor, which defines the relative importance of present and future rewards. The optimal foresighted policy may be obtained by value or policy iteration algorithms, as detailed, *e.g.*, in [7]. When  $\alpha = 0$ , one gets a *myopic* policy, maximizing only the immediate reward.

## 4. EXPERIMENTAL RESULTS

The performance of the proposed layer filtering process has been evaluated on several video sequences. Here, due to lack of space, only the results for Foreman.qcif at  $f_r = 30$  fps are reported. The H.264/SVC encoder using  $L = 3$  MGS scalability layers per frame has been used. The cumulated average rates (and PSNR for luminance) are 74.7 kbits/s (32.3 dB) for Layer 1, 165.0 kbits/s (34.7 dB) for Layer 1 and 2, and 327.0 kbits/s (36.82 dB) for all layers. Known average source and encoder characteristics have been considered, leading to known average packet lengths in each layers.

The wireless channel is modeled as a 2-state Markov model ( $n = 2$ ): Good state  $h_t = 1$  and bad state  $h_t = 0$ . The channel

rates are  $R_c = \{0, 240\}$  kbit/s. The channel state transition probabilities are  $p_{11} = 0.9$  and  $p_{00} = 0.8$ , resulting in an average channel rate of 190 kbits/s. Four possible actions per layer are considered at each time instant  $\mathcal{A} = \{-1, 0, 1, 2\}$ . To minimize complexity and facilitate the satisfaction of the constraints described in Section 2.2, the levels of all buffers are quantized into three possible values: 1 representing underflow, 2 for a satisfying level, and 3 for overflow. The post-encoder buffers are assumed to have a maximum size (in term of number of packets)  $S^e = 55$ ; the over and underflow levels are  $S_{max}^e = 50$  and  $S_{min}^e = 10$ . For the MAC buffer, a maximum size of  $S^m = 220$  equal size PDU of 200 bits each corresponding to a maximum size of 44 kbits, and the levels at which it is considered in underflow and overflow are  $S_{min}^m = 10$  kbits and  $S_{max}^m = 25$  kbits. The values of the parameters in the reward function (4), have been set to reflect the importance of the various constraints:  $\gamma_{1,2,3} = \{150, 60, 15\}$ ,  $\lambda_{1,2,3} = \{100, 40, 10\}$  and  $\beta = 300$ . They may be much better adjusted using Lagrange optimization.

In what follows, the channel state is assumed either known, or totally unknown. In the two considered cases, the evolution of the PSNR of the decoded sequence obtained with a myopic policy ( $\alpha = 0$ ) and that with a foresighted policy ( $\alpha = 0.9$ ) are represented in Figure 2.

When the channel state is available, an average gain of about 1.2 dB is obtained with the foresighted policy compared to the myopic one. This gain is mainly due to more packets of the first enhancement layer reaching the receiver. When the channel state is not available, with the myopic policy, about 46 % of the time, the MAC buffer is in the overflow state exceeding some time the maximum buffer size. This situation results in the loss of some PDUs which induces a notable decrease of the received video quality. With the foresighted policy, the MAC buffer is in overflow state for about 25 % of the time but never loses PDU packets.

Without channel information and using the foresighted policy results in a loss of 0.5 dB in PSNR compared to the case of an available channel state. The availability of the state of the MAC buffer provides thus a reasonable estimate of the state of the channel, allowing a satisfying regulation of the received video quality.

## 5. CONCLUSIONS AND PERSPECTIVES

This paper presents a scalable video streaming system over a time-varying wireless channel. This problem is addressed in the framework of MDP. Foresighted policies are developed for two models of the system characterized by the availability or unavailability of the state of the channel to the controller. The level of video packet buffers and that of the MAC buffer of the base station are controlled.

Experimental results show that without channel state information, the performance of the control system is only slightly degraded compared to a case where the channel state information is available, provided that a foresighted policy is used. The observation of the level of the MAC buffer provides a satisfying estimate of the channel state. With a myopic policy, the degradation is much larger.

This result may be very useful, since using channel state in the control system requires processing ACK/NACK or RTCP feedback messages, which usually only provide a delayed estimate of the channel state.

## 6. REFERENCES

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sept. 2007.

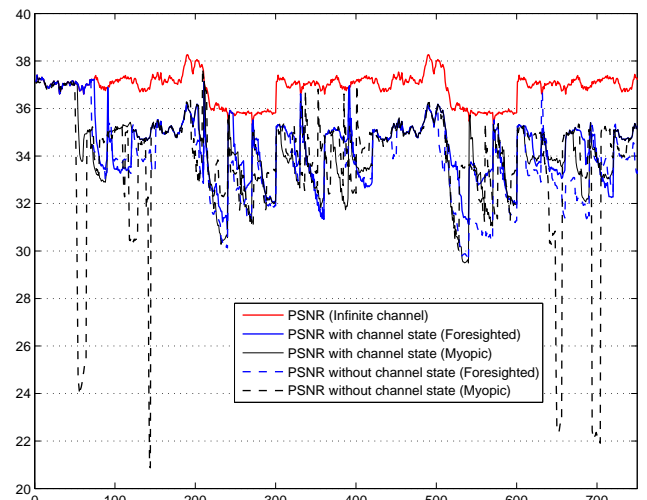


Fig. 2. PSNR of the decoded sequences, myopic and foresighted policies, for known and unknown channel state

[2] H. Schulzrinne and S. Casner, *RTP profile for audio and video conferences with minimal control*, The Internet Engineering Task Force, request for comments 3551 edition, July 2003.

[3] S. Sesia, I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice*, chapter 17, Feb. 2009.

[4] W. Kumwilaisak, Y.T. Hou, Qian Zhang, Wenwu Zhu, C.-C.J. Kuo, and Ya-Qin Zhang, "A cross-layer quality-of-service mapping architecture for video delivery in wireless networks," *IEEE JSAC*, vol. 21, no. 10, pp. 1685 – 1698, dec. 2003.

[5] P. de Cuetos and K. W. Ross, "Optimal streaming of layered video: joint scheduling and error concealment," in *ACM Multimedia*, Berkeley, Nov. 2003, pp. 2–8.

[6] N. Changuel, C. Mastronarde, M. Van dershaar, B. sayadi, and M. kieffer, "End-to-end stochastic scheduling of scalable video over time-varying channels," in *ACM Multimedia*, Oct. 2010.

[7] R.S Sutton and A.G Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[8] N. Baldo, U. Horn, M. Kampmann, and F. Hartung, "RTCP feedback based transmission rate control for 3g wireless multimedia streaming," in *PIMRC*, 2004.

[9] G. Liebl, H. Jenkac, T. Stockhammer, and C. Buchner, "Radio link buffer management and scheduling for wireless video streaming," in *Packet Video Workshop*, 2004, number 1-3.

[10] Z. Orlov and M. C. Necker, "Enhancement of video streaming QoS with active buffer management in wireless environments," in *European Wireless Conf.*, Apr. 2007.

[11] S. Yerima and K. Al-Begain, "Dynamic buffer management for multimedia qos in beyond 3g wireless networks," *International Journal of Computer Science*, vol. 36, Nov. 2009.

[12] F. Fu and M. van der Schaar, "A new systematic framework for autonomous cross-layer optimization," *IEEE Trans. on Vehicular Technology*, vol. 58, no. 4, pp. 1887 –1903, may 2009.

[13] N.H. Mastronarde and M. van der Schaar, "Designing autonomous layered video coders," *Signal processing. Image communication*, vol. 24, no. 6, pp. 417–436, July 2009.

[14] H. Mansour, V. Krishnamurthy, and P. Nasiopoulos, "Rate and distortion modeling of medium grain scalable video coding," in *ICIP*, oct. 2008, pp. 2564 –2567.

[15] F. Wan, L. Cai, and T.A. Gulliver, "A simple, two-level markovian traffic model for IPTV video sources," in *IEEE Globecom*, nov. 2008, pp. 1 –5.

[16] E. Altman and P. Nain, "Closed-loop control with delayed information," *Perf. Eval. Rev.*, vol. 20, pp. 193–204, 1992.