Link Adaptation and Cross-Layer Signaling for Wireless Video-Streaming in a Shared Medium

Ivaylo Haratcherev, Koen Langendoen, Reginald Lagendijk, Henk Sips
Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
{I.Haratcherev, K.G.Langendoen, R.L.Lagendijk, H.J.Sips}@ewi.tudelft.nl

***Abstract*— Changes in the quality of wireless links impose great demands on video codecs and underlying network layers when seamless video-streaming is to be achieved. Moreover, it is not enough that only the video codec or only the radio adapts to these changes; the efforts should be applied in both layers, and – if possible – synchronized. So, on the one hand, a responsive link adaptation method should be employed in the radio. And, on the other hand, the video codec should be able to follow the changes in the maximum throughput due to wireless link performance variations. In this paper we present the results of video-streaming over 802.11a link in the presence of background traffic, generated by other stations sharing the same medium. We show that great improvements in the quality of the video can be achieved by cross-layer signaling between the link layer and the video coder. However, we show that this is only realizable if a correct estimation can be made of the throughput decline due to the medium sharing.**

***Keywords:* video streaming, medium sharing, rate control, MAC layer, SNR, link adaptation, cross-layer interaction, 802.11**

I. INTRODUCTION

Having a wireless last hop in the Internet is something that becomes more and more popular today, and so is using the Internet for audio and video streaming. The combination makes that the demanding world of real-time multimedia (which does not tolerate drop-outs for example) meets the quite imperfect – and capricious – dark universe of radio links. A lot of effort is required to team these worlds together, so that the stringent packet delay, jitter, and loss requirements of multimedia applications can be met by the unstable and unreliable radio link.

Different approaches exist to keep the marriage of multimedia and wireless happy. First, the so called link adaptation is typically being applied at the link (MAC) level [1], [3], [6], [11]. Roughly speaking, link adaptation is the process of automatic selection of radio/MAC parameters, so that optimal quality of packet transmission is achieved. By optimal quality we mean minimizing packet loss and packet delay, while keeping the data throughput as high as possible.

Second, the video encoder can also adapt to the link quality, for example, by changing the compression degree, and thus modifying the (video) data rate [2], [7]. This adaptation requires that the video encoder is able to sense the link quality, for example, by getting feedback information from the decoder side. However, such a scheme is ineffective when the round-trip delays are long. Using this approach also introduces additional overhead. A better approach, described in [4], for the video encoder is to base the adaptation on information provided by the link adaptation entity at the radio level.

It is important to note here that control schemes in the lower layer (i.e. at the MAC level) have precedence in terms of achieved improvement in performance, to control schemes in the upper layer (the video codec). In other words, and to put it simply, there is no way that you can compensate for packets already lost at the link level.

Another significant observation is that the performance gain of the whole system can be nullified in situations where the information provided by the lower layer to the upper layer is incorrect. In our case failing to consider the sharing of the medium in the throughput estimation of the wireless channel can be fatal, as we show in this paper.

We present results of real experiments that show how much performance can be gained by using an ideal throughput estimator. Although we emulate the estimator (by knowing beforehand what would be the available throughput while sharing the medium), we present an idea of how such an estimator, having sufficient prediction accuracy, can be built in practice. Building and testing the complete estimator, however, constitutes future work.

The remainder of the paper is organized as follows. In the next section, a discussion is presented about the degree of control at codec and MAC side, and their inter-operation, depending on the harshness of the environment and the posed constraints. Section III gives a description of the adaptive video coder and the basics of link adaptation. The cross-layer communication and the medium sharing prediction are discussed in Section IV. The experimental results are discussed in Section V. The conclusions and future plans are presented in Section VI.

II. CONTROL COMPLEXITY AS FUNCTION OF CONSTRAINTS SEVERITY

When designing a communication system, it is important to know what the inherent problems in the communication link are, and their severity. Then, given the constraints that the application imposes, such as the performance quality variations that it could handle, an appropriate control mechanism is applied. On the one hand this mechanism should assure that the quality variation constraints are kept, and on the other, it is desirable to keep the complexity of this mechanism as low as possible. Keeping the complexity of the control low minimizes the design efforts, and, in most cases, also guarantees maximum robustness of the whole system, compared to the case with more-complex control.

In Figure 1 we show three cases of different control complexity, which are the basis of our experiments. Case 1 depicts the situation when there is a need for link adaptation by the radio/MAC layer, in which situation some kind of channel state prediction is employed. In case 2, the channel state information (CSI) is also supplied to the video codec via a simple radio throughput to real throughput function. This information is used by the codec to adapt the video rate accordingly, so that the video data does not choke the radio link. The most complex is case 3 where the real link throughput is calculated from the CSI by a medium sharing predictor (MSP) or estimator. As shown below, this is very important when there are other 802.11 users in the same Basic Service Set (BSS), for example. Cases 1 and 2 were explored in earlier work [5], [4], while 3 is investigated in this paper.

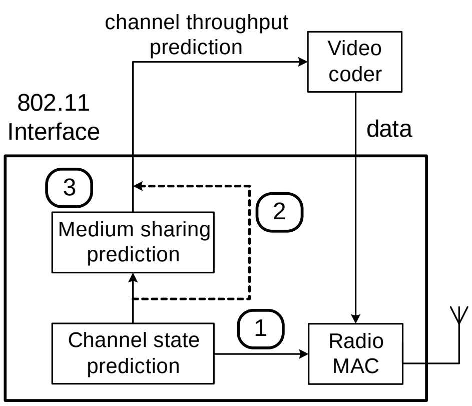

Fig. 1. Cases of prediction. 1) Channel state prediction. Used for Link Adaptation. 2) Channel throughput prediction without considering medium sharing. 3) Channel throughput prediction considering medium sharing.

In Table I we present the minimal degree of control complexity, depending on the video requirements and the environment conditions. Let us look at the upper part first (no medium sharing). In the left half we have the cases when the channel is static, i.e. there is no movement or any change in the link quality. Then we do not need channel state prediction, and we do not have to change any parameter in the radio. A typical example of such links are satellite links. However, in our case, the channel is dynamic, so we fall into the right half of the table. There we definitely need channel state prediction to be able to adapt to the channel quality variations. Failing to do so will have catastrophic consequences, no matter how smart the network layers above are, just because a broken radio link means that no packets arrive at all.

Moreover, in the case of high quality video requirements, we also need the codec to have some information about the available throughput. The rationale behind this is discussed in [4].

The picture changes completely when the wireless channel is shared. In Figure 2 we compare the throughput available

	Static channel		Dynamic channel (movement)	
Medium	Low quality video	High quality video	Low quality video	High quality video
No sharing	-	-	1	1+2
Sharing	(3)	3	1+3	1+3

TABLE I

MINIMAL SYSTEM CONFIGURATION (SEE FIGURE 1) FOR DIFFERENT REQUIREMENTS, NECESSARY FOR PROVIDING SEAMLESS VIDEO-STREAMING.

to a typical video-streaming application, with, and without background download (TCP) traffic present. For completeness we also present the predicted throughput value, given by the prediction model presented in Section IV. As it is shown in the Figure, the throughput can drop significantly, even with only one more user who is downloading. Therefore, we need medium sharing prediction to be employed in addition to channel state prediction, for all cases (static/dynamic channel, low/high video quality).

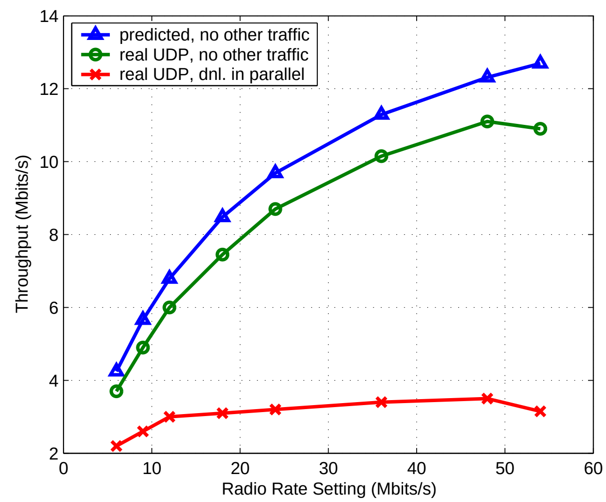

Fig. 2. Available throughput for a video-streaming application with, and without background download (TCP) traffic, compared with predicted value as well. The used UDP packet size is 341 bytes.

The data in Figure 2 (except the predicted values) is obtained by real-time experiments, where the video-streaming and the download traffics are emulated by using NetPerf. The UDP packet size is chosen to match the average packet size generated by our video encoder in the experiments in Section V.

III. MAC LINK ADAPTATION AND ADAPTIVE VIDEO CODING

In the IEEE 802.11 standard [8] several MAC-level parameters are defined, which are available for tuning at the side of the Wireless Network Interface Card (NIC). From those parameters, the most important one available for tuning is the transmit rate. In 802.11a [9], for example, it can be set to 6, 9, 12, 18, 24, 36, 48 and 54 Mbits/s. Each rate setting offers a different trade-off between data throughput and achievable distance between stations. Higher rate settings offer a larger throughput, but also provide a shorter operating range. Usually, one wants to extend the operating range as much as possible and, at the same time, to maximize the throughput. This can be done by proper (automatic) selection of the rate.

In order to decide which rate is optimal at each specific moment, a control algorithm needs information about the current link conditions, or so-called channel state information (CSI). Since it is difficult to get CSI directly, most MAC rate-control algorithms use some form of statistics-based feedback, for example, user-level throughput. The main disadvantage of this indirect feedback is that it is inherently slow, causing communication drop-outs when the link conditions degrade rapidly (e.g., when the user moves fast). Streaming applications are very sensitive to these drop-outs. Consequently, streaming applications perform poorly under standard automatic rate-control.

To achieve fast accommodation, we use the SNR-based hybrid rate controller that we described in previous work [5]. In this controller, the SNR information is used as a safeguard limiting the range of feasible rate-settings, from which the core (statistics-based) controller can choose. We also use the predictor that we developed in [4] to provide feedback information about the channel conditions to the video encoder. This feedback consists of information about the current channel status, along with short-term prediction (about 1 frame) that helps the video coder to promptly follow fast changes in the available channel throughput.

We use a H.263 video codec that is modified to support interaction with the link layer. Our version of the H.263 encoder supports a (video) rate-control algorithm (VRCA) that tries to achieve a certain rate, by adjusting the quantization step size. The quantization step size is the main parameter that controls the compression of the video. This VRCA was designed for constant bit rate (CBR) encoding, but it can also be used to dynamically change the bit rate that is produced by the encoder. Since the resulting bit rate depends on the selected quantization step size but also on the statistics of the picture itself, we cannot set the bit rate beforehand, and then expect that this bit rate will be exactly achieved by the encoder. The VRCA, is a simple feedback control loop that consists of setting an initial quantization step size, encoding part of the picture, measuring the resulting intermediate bitrate, changing the step size accordingly and then continuing with the next part of the picture. In total there are nine parts of a picture frame for which the quantization step size can be adjusted, which generally is enough to be able to achieve a certain preset rate. With this algorithm, we can change the target bit rate for each individual frame, meaning that we have a maximum delay of 40 ms to respond to changes in the channel. In our experiments, we adapt the target video encoding rate for the VRCA, based on the feedback information provided by the 802.11 link layer.

IV. Cross-Layer Signaling and Medium Sharing Prediction

As discussed in the previous section, the VCRA uses throughput prediction information, provided by the 802.11 link layer. This information is generated in the following way.

On the MAC side the predictor uses a history of SSIA (Signal Strength Indication of Acknowledgements) measurements to maintain a short-term (in order of few tens of milliseconds) SSIA average. From this average future SSIA are predicted. Our experiments show that the best results are obtained by a first or zero-th order predictor. This conclusion agrees with theory, stating that channel state change is a memoryless process.

The SSIA predictions are matched against the SSIA thresholds in the SSIA-rate lookup table in the radio rate-control algorithm [5], and so the rate predictions are determined.

Then, for the case with no sharing of the medium we use a simple model, to convert from radio rate setting to available user-space data throughput. Our model is derived from the IEEE 802.11 standards (see [8], [9] and [10]), and is in the form of:

$$T = \frac{8RL}{8L + bR + c}$$

where T is the throughput in Mbits/s, L is the length of a packet in bytes, R is the data rate setting in Mbits/s, and b and c are coefficients that depend on the 802.11 supplement. For 802.11a, $b = 161.5$ and $c = 156$. For 802.11b, $b = 754$ in the case of long preamble, or $b = 562$ in the case of short preamble, and $c = 112$.

When we share the medium with other users however, the throughput drops significantly, as was shown in Figure 2. We do not have a way to detect when other stations start using the radio channel. Therefore, to be able to predict the resulting throughput accurately, we implemented a throughput predictor emulator. This emulator is a lookup table with throughput values for each rate setting, that we measured beforehand in the presence of background traffic. The values are propagated as prediction values to the video codec, at the times we know the radio channel is used by other stations. In this way we can observe what the performance improvement will be in the ideal scenario of perfect throughput prediction.

As a practical implementation of a real throughput predictor, which takes the effects of medium sharing into account, we intend to investigate the following algorithm as future work. Trying to transmit over the air is usually a bursty process (streaming, download), and the time between switching to active state and then back to inactive state is in the magnitude of seconds. Then, we can use the statistics of the observed throughput for each radio rate setting, during previous transmissions, as prediction what the real throughput will be during the next transmissions. As the medium utilization changes slowly, our prediction should work fine for the period that we are interested in (couple of tens of milliseconds).

Another explanation of the algorithm used by the throughput predictor is the following (cf. Figure 2). The available throughput depends on both the radio rate setting (i.e. where we are on a curve), and the medium sharing (that is, which curve do we select). Although there is no practical way to find out how many stations are sharing the medium, the number of these stations changes relatively slowly. Therefore we can use throughput statistics to determine which curve we are currently on. Our position on the curve, which could change rapidly because of fast changes in link conditions, can be tracked by the help of the fast SNR feedback that we already employ for link adaptation.

V. Experiments

Our experimental setup is presented in Figure 3. The setup

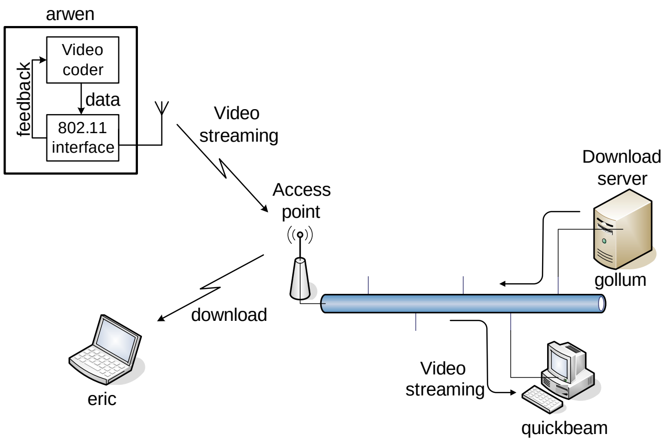

Fig. 3. Our experimental setup.

consists of two desktops (”golum” and ”quickbeam”) on an ethernet link, a 802.11a access point, and two laptops (”arwen” and ”eric”) – both running Linux. The laptops are equipped with 802.11a cards based on the Atheros AR5000 chipset, and the card driver uses the advanced hybrid rate control algorithm, as described in [5]. The experiments were carried out by streaming a video file between ”arwen” and ”quickbeam” while ”arwen” was moving, following a predetermined track. The track consists of three parts - a ”lead-in” of walking from the room where the access point lies, to a specific start position in the hallway. Then follows moving with the laptop up and down the hallway for about 20s (”action”). Finally we go back (”lead-out”) again into the room. While the video streaming takes place, ”eric” downloads a file from ”gollum” (during the ”action” part) for about 10 seconds.

To evaluate the performance of the coupled rate-control systems – that is, systems in which multimedia applications get feedback from the radio – we have examined two cases. For each case, we have repeated the experiments several times. We managed to quite precisely time the experiments and to obtain a reasonable repeatability of the results.

1) **Decoupled** : Decoupled rate control. The video codec has no indication of the actual throughput, and we set the target-rate for the VRCA to a fixed value, which is the *minimum* throughput that would be obtained in the case there is no medium sharing (about 4 Mbits/s - see Figure 2). In this way we cover both the cases when there is no cross-layer signaling (fixed rate) and when there is a throughput prediction feedback, but being incorrect due to the medium sharing.
2) **Coupled-MSP** : Coupled rate control, taking into account medium sharing. The rate control loops of the MAC-layer and the Video coder are coupled by letting the video coder obtain throughput prediction feedback information. In this case we are using the emulated throughput predictor that provides proper throughput estimation, when the medium is shared with another user.

In Figure 4 the quality (PSNR) is shown for the second (”action”) part of the experiment[1]. In Figure 5 the received-signal strength is shown for the same time span, so the change in link conditions due to the movement of ”arwen” can be observed.

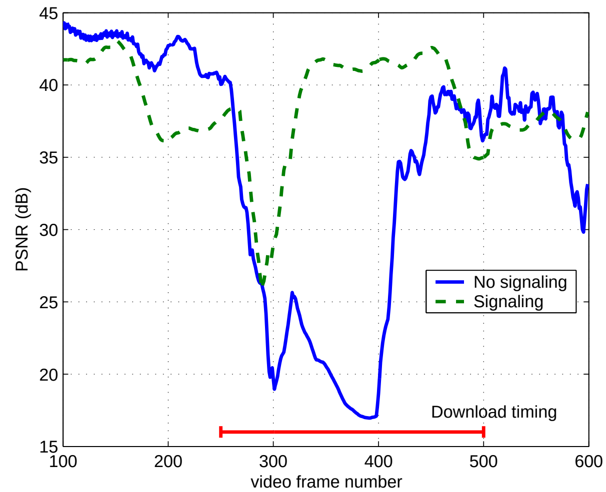

Fig. 4. PSNR of the received video: The solid line shows the results for a decoupled rate control using fixed rate (Decoupled). The dashed line shows the quality during the experiment for the Coupled-MSP.

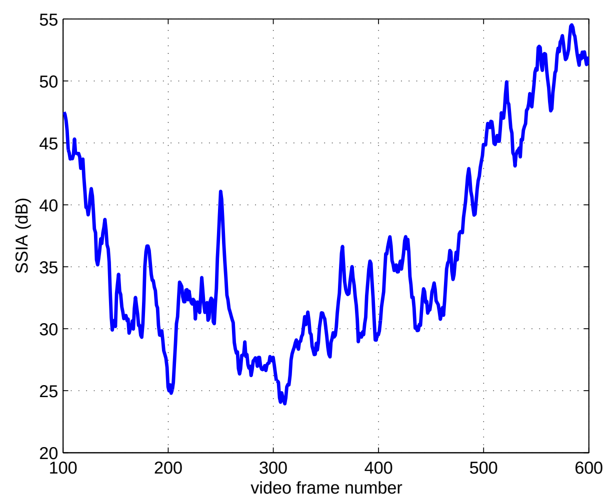

Fig. 5. Measured SSIA during experiments.

During the time when the download takes place, with the Coupled-MSP case we lose just a few frames, since the video encoder properly reduces the video-rate, following the throughput prediction feedback. With the Decoupled experiment, the higher video-rate selected by the video encoder

[1] We have applied an averaging sliding window to the data for better visibility.

causes the wireless interface to choke during the time the download takes place, and the video freezes as a result of the multiple frame loss. This ”freeze” can be localized by the regions with low PSNR. At about video frame 400, the video data manages to get through again, since the link conditions improve, as the latter can be observed in Figure 5 (the average SSIA increases).

In the periods where there is no background traffic (video frames from 100 till around 250, and from around 500 till 600), the PSNR of Coupled-MSP is slightly lower than the PSNR of Decoupled. This is because we use the throughput prediction for the case of background traffic, for the duration of the whole experiment. We do that to evaluate what the loss of quality will be, when there is pessimistic misprediction of the medium sharing (less stations to share the medium with than predicted).

The mean PSNR for the period of the whole Decoupled experiment is 38.1dB, and for Coupled-MSP is 39.1dB. The mean PSNR for the period with download, for Decoupled is 28.3dB, and for Coupled-MSP is 38.6dB. We see that a significant improvement (over 10dB) is achieved in the Coupled-MSP case, for the period the medium has been shared. The penalty for pessimistic throughput estimation in the case of wrongly predicting medium sharing when there is not one (no background traffic present), is quite low. This, however, does not justify abandoning throughput prediction and just designing the video codec to go for low throughput value. Doing the latter would be impractical, because with the increase of the number of stations in a BSS, the minimum available throughput will be very low, resulting in very low overall video quality.

It is important to note that the results shown in Figure 4 are not the most optimistic in terms of improvement that we have obtained during our tests. This is because in many of the experiments with the Decoupled case, the decoder on ”quickbeam” would just stall because too many frames were lost, and consequently, we were not able to get the PSNR trace.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the results of a real-time streaming video experiment over a wireless 802.11a connection, in the presence of background traffic. We show that great improvements in the quality of the video can be achieved by cross-layer signaling, only in the case of correct estimation of the throughput decline due to the medium sharing. Also, our results undoubtedly demonstrate that misprediction of medium sharing can be largely tolerated, if it is pessimistic. That is, when the real number of stations the medium is shared with, is lower than the predicted number. We introduce a promising idea for a real throughput predictor that takes into account the effects of medium sharing, as well. As future work, we plan to investigate practical implementation of this real throughput predictor.

REFERENCES

[1] K. Balachandran, S.R. Kadaba, and S. Nanda. Channel quality estimation and rate adaptation for cellular mobile radio. *IEEE Journal on Selected Areas in Communications*, 17(7):1244–1256, July 1999.

[2] J-C. Bolot and T. Turletti. Experience with control mechanisms for packet video in the internet. *Computer Communication Review*, 1998.

[3] Javier del Prado Pavon and Sunghyun Choi. Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *IEEE International Conference on Communications, 2003 (ICC ’03)*, volume 2, pages 1108–1113, Anchorage, Alaska, USA, May 2003.

[4] I. Haratcherev, J. Taal, K. Langendoen, R. Lagendijk, and H. Sips. Fast 802.11 link adaptation for real-time video streaming by cross-layer signaling. In *Proc. International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.

[5] I. Haratcherev, J. Taal, K. Langendoen, R. Lagendijk, and H. Sips. Automatic IEEE 802.11 rate control for streaming applications. *Wireless Communications and Mobile Computing, special issue on Radio Link and Transport Protocol Engineering for Future-Generation Wireless Mobile Data Networks*, accepted for publication, 2005.

[6] A. Kamerman and L. Monteban. WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal*, pages 118–133, Summer 1997.

[7] A. Ortega and M. Khansari. Rate control for video coding over variable bit rate channels with applications to wireless transmission. In *Proceedings of the 2nd IEEE International Conference on Image Processing (ICIP)*, October 1995.

[8] IEEE standard 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[9] IEEE standard 802.11a supplement. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. High-speed Physical Layer in the 5 GHz Band, 1999.

[10] IEEE standard 802.11b supplement. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Higher-speed Physical Layer in the 2.4 GHz Band, 1999.

[11] A.J. van der Vegt. Auto Rate Fallback Algorithm for the IEEE 802.11a Standard. Technical report, Utrecht University, 2002.