LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE FRANCHE-COMTE

EA 4269

VAAL, Video Adaptation at Application Layer and Experiments using DCCP

Wassim Ramadan — Eugen Dedu — Julien Bourgeois

Rapport de Recherche n° RR 2011–????

THÈME 4 – Octobre 2010

VAAL, Video Adaptation at Application Layer and Experiments using DCCP

Wassim Ramadan, Eugen Dedu, Julien Bourgeois

Thème 4

OMNI

Octobre 2010

Abstract: More and more high definition and high quality videos are nowadays stored on Internet but they are not accessible for everybody because high and stable bandwidth is needed to stream them. A lower bitrate encoding is usually chosen but it leads to lower quality too. A solution is to use a TCP-friendly transport protocol and let the application itself choose for each user the bitrate of the video which matches the network bandwidth. This paper presents VAAL, a simple and efficient method designed to ameliorate user video experience. It involves only the application layer on the sender. The main idea of VAAL is that the sender application monitors the network conditions by checking transport protocol buffer overflows and adapts the video bitrate accordingly. Experimental results show that the video adaptation using VAAL performs very well compared to the currently widely-used static encoding.

Key-words: Video streaming, content adaptation, TCP-friendliness, DCCP.

Laboratoire d'Informatique de l'Université de Franche-Comté,
Antenne de Montbéliard — UFR STGI,
Pôle universitaire du Pays de Montbéliard,
25200 Montbéliard Cedex (France)
Téléphone : +33 (0)3 81 99 47 43 — Télécopie +33 (0)3 81 99 47 91

VAAL, Adaptation de la Vidéo au niveau de la couche Application et des Expérimentations avec DCCP

Résumé : De nos jours, de plus en plus des vidéos de haute définition et de haute qualité sont stockées sur Internet. Mais elles ne sont pas accessibles à tous car une bande passante élevée et stable est nécessaire pour les acheminer à leurs destinations. Un encodage bas débit est généralement préféré, mais il conduit aussi à une moins bonne qualité. Une solution consiste à utiliser un protocole de transport *TCP-friendly* et laisser l'application choisir elle-même, pour chaque utilisateur, le *bitrate* de la vidéo qui s'adapte le mieux à la bande passante du réseau. Cet article présente VAAL, une méthode simple et efficace, conçue pour améliorer l'expérience vidéo de l'utilisateur. Elle ne concerne que la couche application de l'émetteur (serveur). L'idée principale de VAAL est que l'application émettrice surveille l'état du réseau en vérifiant les débordements de tampon du protocole de transport et adapte le *bitrate* vidéo en conséquence. Les résultats expérimentaux montrent que l'adaptation vidéo effectuée par VAAL donne de très bon résultats comparée à l'encodage statique actuellement répandu.

Mots-clés : Video streaming, adaptation de la vidéo, TCP-friendliness, DCCP.

Laboratoire d'Informatique de l'Université de Franche-Comté,
Antenne de Montbéliard — UFR STGI,
Pôle universitaire du Pays de Montbéliard,
25200 Montbéliard Cedex (France)
Téléphone : +33 (0)3 81 99 47 43 — Télécopie +33 (0)3 81 99 47 91

I INTRODUCTION

Multimedia streaming services over Internet, as well as the demand for higher quality from final clients are in constant progression. New video standards like HD and 3D are asking for more bandwidth. Available bandwidth variation has also to be taken into account so that buffering time can be shortened. Watching such videos on a wireless network is not a comfortable experience, because there are various wireless network technologies with different characteristics, and they can change over time. Hence, available bandwidth is not always stable because of many reasons (interferences, mobility etc.) Also sharing the bandwidth with several users could make the available bandwidth lower than the bitrate required for the video.

Additionally, more and more network applications, for example real-time media like audio and video streaming, can accept a certain level of losses. If they use TCP (Transmission Control Protocol), they have to pay the price of full reliability, with great latency. On the other hand, UDP (User Datagram Protocol) lacks congestion avoidance support. RTP (Real-time Transport Protocol) [1], while being a widely-used protocol for multimedia streaming, is an application protocol ; as such, it is put on top of a transport protocol, such as TCP or UDP, hence it does not cope with transport protocol problems.

Another promising protocol for these applications is DCCP (Datagram Congestion Control Protocol), recently standardised as RFC4340 [2]. It can be seen as TCP minus reliability and in-order delivery of packets, two key points in video streaming, or as UDP plus congestion control. For our purposes, two interesting points of DCCP are that it allows to choose the congestion control used during communication, and that it uses acknowledgements. Among the currently three standardised congestion control protocols, TFRC (TCP-Friendly Rate Control) is the most adapted to video streaming [3]. Also, the acknowledgement packets give useful information to the sender, such as the lost packets and ECN marks.

For the above protocols, especially DCCP and TCP, video transmission is controlled at the network layer and the application is not involved at all. For video streaming in a network with highly variable bandwidth during a connection, an adaptation of the video to the network characteristics is very important. A cooperative approach between application layer and network layer can improve the video quality perceptible by the final user.

The adaptation method we purpose (VAAL) uses transport protocol buffer overflow as a solution to find out the available bandwidth. Each n fixed seconds, the server application computes the number of packets which failed to be written to socket buffer. This number is used to control the video bitrate afterwards. A high number means smaller bandwidth and smaller bitrate. Zero error indicates either a stable or more bandwidth, so bitrate of sent video can be increased.

The above adaptation, known in the literature as "rate adaptive video control", can be done by controlling some other video parameters, such as number of frames per second (FPS) and image size. These possibilities generate a heterogeneous video at the client, and the client application must be capable to recreate a readable video from it.

The three parameters presented above allow to optimise the *network* part of the video streaming. For yet better results, these methods could be *coupled with other methods*. For example, a well-known problem with losses in wireless networks is that they cannot be differentiated from congestion losses, hence the sender reduces the throughput while it should not [4]. Another example useful on lossy links is the FEC (Forward Error Correction) ([5] for example). Also, *video-specific techniques* allow for example to prioritise [6] or retransmit [7] only important packets (I packets in an MPEG-encoded video) on the server side.

In this context, our paper is, to our knowledge, *one of the rare papers which analyse the video content adaptation* ([8] is one of them), and *the first one which uses DCCP in real experiments in wireless networks*. Moreover, this is the first time that the *buffer overflow is proposed for video adaptation*. Our solution is *very simple to deploy*, as only the application on the sender side needs to be modified (does not need to modify the receiver, nor the transport protocol). As a corollary, our method works with any transport protocol which has a congestion control.

This paper is organized as follows. Section II presents related work for video adaptation. Section III presents our VAAL method and its implementation on GNU/Linux. Performance of VAAL is evaluated through real experiments in section IV. Finally, section V concludes this article.

II RELATED WORK

The idea of transmitting video in a network-friendly manner by changing only the endpoints is not new. Several methods can be used for that. One is to selectively transmit packets, based on available bandwidth [6], or selectively retransmit packets, based on its importance (I frames are to be always retransmitted, contrary to P and B frames, for example) [7]. Another method is content adaptation of the video streaming, based on the available bandwidth for example. As our paper fits into this category, we present here works using this method.

Clearly, one of the ISO layers should take care of the re-encoding. We classify the current works based on the layer which takes care of that.

One approach is that the application layer stays unchanged, and *the lower layers change*. In [9] the application sends video layers encoded with H.264/SVC video codec. The IP layer receives all the video layers and sends only those which are network-friendly. The bandwidth is estimated through regular packet probes.

Another approach is that *both layers change* : the lower layers give feedback on the network conditions to application layer, which acts accordingly (for example, it adapts the video bitrate). From the application perspective, the feedback can be push (the application asks for information) or pull (the network wakes up the application and informs it).

A pull solution is iTCP [10], which adds to TCP an active (network-driven) event-based feedback mechanism (such as re-

transmit timeout triggered and a third duplicated acknowledgement (dupack) received), which can be used by the application to adapt to network conditions. It has the drawback that, at receiver side, a lost packet blocks the receiver application until its retransmission arrives. DCCP for example does not suffer from this problem, since it is not a reliable protocol. When used with TFRC, it becomes more appropriate to video streaming because it has a smoother throughput curve.

VTP [11] is a new application protocol : it has drawbacks inherent to congestion control at application level, it needs a modification to the receiver application, and there is a risk that it is not TCP-friendly. They made experiments on linux.

[12] does the same, but at the transport level, by creating a new transport protocol.

Finally, a third approach is that *only the application layer changes*, and the application uses the information provided by lower layers. In [13], the application uses multi-view video encoded with H.264/SVC. It periodically reads available bandwidth from DCCP layer. It sends as much layers as it can, without exceeding the bandwidth. However, this paper is video-oriented and does not analyse any network issues at all.

Back in the year 2000, before the creation of DCCP, [14] too used re-encoding (by changing the quantizer parameter of the encoder) to adapt the video to network bandwidth. But to find out the network bandwidth, they implemented a congestion control at application level using RTCP feedback. Some drawbacks are that this solution depends on RTCP, the calculus is not precise, because e.g. it is done at the application level, and most important there is a risk of non-TCP-friendliness each time a new congestion control is conceived.

In our previous work [15] DCCP protocol was used too, with real traces, but in simulation. That paper does not focus on network issues, like this one, but on the complete streaming architecture, including an RTP mixer between the server and the client. As a side note, the reencoding (content adaptation) was done in the mixer, which needed to have access to DCCP data through a cross-layer mechanism.

The nearest works to ours are [8] and [16], which analyse DCCP video adaptation through ns2 *simulations* with real traces.

Our method too fits into this category. We think the best level to perform adaptation is the application level. We also think that if a rate adaptation method is to be deployed, then it should have minimum changes to existent operating systems and applications.

III   VAAL ADAPTATION

III.A   VAAL explanation

As shown in Fig. 1, the application writes packets to the transport protocol socket buffer at a rate equal to the current video bitrate. The transport protocol has a congestion control which gives the rate at which the packets leave the machine and enter the network. VAAL adjusts the video bitrate to the rate estimated by the transport protocol. Thus, VAAL algorithm is divided in two steps, presented in Fig. 2 : discovery of the network conditions (available bandwidth based on buffer over-
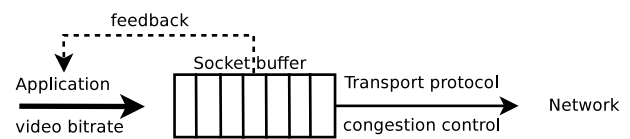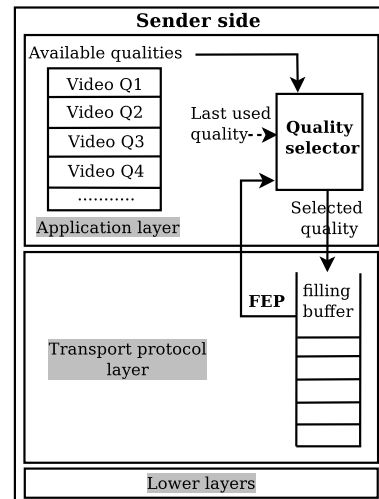

FIGURE 1 – Video data flow on the sender side.

FIGURE 2 – VAAL method on the sender side.

flow) and the action to be done (the quality selection). These two steps are executed each period of n fixed seconds.

The discovery of network conditions works as following. When the transport protocol cannot send more packets to the network (lower layers), they are buffered. If the buffer becomes full, the new packets generated by application will fail to be written (buffer overflow). Thus, VAAL monitors the available network bandwidth through the transport protocol socket buffer overflows when the application tries to write a packet in it. During each period, VAAL computes the percentage of these failed packets, which we will call *failure error percentage* (FEP). Consequently, FEP is an indication of the network conditions : the bigger the FEP, the less the available bandwidth.

The quality selection (adaptation) works as following. VAAL starts the connection by sending the highest video quality available from the video source. At the end of each period, VAAL reads the FEP (given by the first step) and acts like this :

– If FEP is null (no packets failed when written to buffer) VAAL chooses the next higher quality level (higher bitrate) unless the quality is already the highest.
– Elsewhere, if FEP is tolerable (smaller than 5%), the quality is maintained at the same level. ITU.T G.1070 [17] recommends that the end-to-end IP packet loss rate in video streaming should be less than 10%. Hence, we chose a threshold of 5% of packet loss rate at the sender buffer (FEP $< 5\%$), the other 5% being left to handle the network losses.
– Finally, when FEP is greater than 5% and unless the lowest quality is already in use, VAAL searches for the next
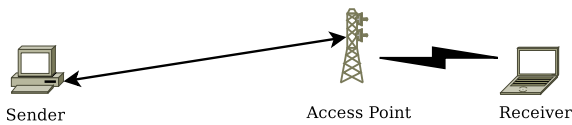
FIGURE 3 – Network topology used for experiments.

lower quality q' which fulfills :

$$q' \leq q(1 - \mathrm{FEP})r \qquad (1)$$

where q is the current quality. In this formula, $q(1-\mathrm{FEP})$ represents the bandwidth available for the period which has just ended, while r is a parameter which allows us to choose a quality with a bitrate different than the available bandwidth.

Note that VAAL requires a transport protocol with a network congestion control, no matter which one. Also, VAAL is especially useful in video conferencing (video on-the-fly) because there is no need to re-encode the video, just changing the encoding rate.

III.B Implementation

We have implemented VAAL video adaptation at the application layer on a GNU/Linux machine with 2.6.31 kernel (without any change to system kernel). It uses DCCP as transport protocol together with TFRC as congestion control. The FEP is computed each two seconds. Since at 95% loss rate VAAL maintains the bitrate, we chose $r = 1.05$, i.e. we tolerate up to 5% of lost packets.

IV EXPERIMENTAL RESULTS

Fig. 3 shows the real network used to realize tests with the program presented in the previous section. A video streaming connection is made between a sender, connected to an access point (AP) via a wired cable, and a receiver, connected to the same AP via a wireless connection (see appendix for more information). The video streaming uses a real video in four qualities 3Mb/s, 2Mb/s, 1Mb/s and 512kb/s. The video has 180s.

All the tests use ten concurrent flows, and they use identical algorithms (i.e. all of them use adaptation, or all of them use some fixed quality). This allows to see what happens when multiples flows sense the available bandwidth, especially to check if this leads to a wide oscillation in performance. In fact, ten flows at maximum bitrate exceed the bandwidth (10*3Mb/s = 30Mb/s, which clearly exceeds the bandwidth provided by a classical 54Mb/s wireless network). We have tested also for five concurrent flows but due to limited number of pages we cannot present their results here. In brief, their results are similar to ten flows, but since there are fewer flows, they get a higher bandwidth, so VAAL is able to choose a higher quality for each of them.

Moreover, two series of tests are done. For the first series, all flows are present at the same time (called in the following *flows without gap* series). In the second, each flow starts 30 seconds after the beginning of the previous flow, except the first one which starts at time 0 (called in the following *flows with gap*
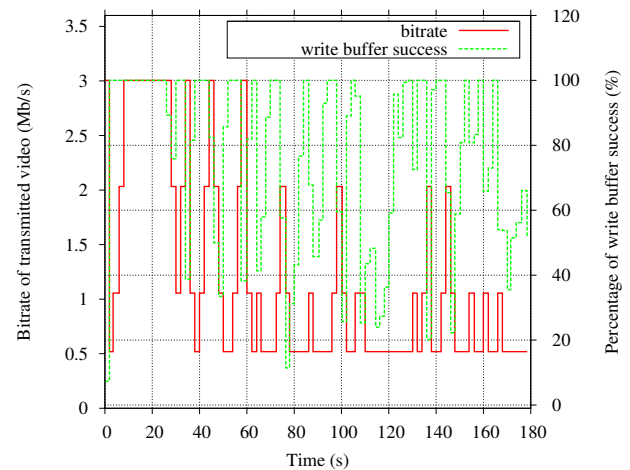

FIGURE 4 – Quality variation for ten flows without gap, flow 1.

series). For both series, each flow waits for a random number between 0 and 4 seconds, to prevent that each two seconds all flows change the bitrate at the same time.

Each test is repeated three times but only one representative result is presented here. Note that there is no retransmission for lost packets in all our tests.

We present two results : the bitrate adaptation takes well into account the DCCP buffer feedback, and VAAL outperforms the widely-used static encoding.

IV.A Quality variation

In this section, we discuss the quality variation made by VAAL. As seen before, every two seconds VAAL looks at the failure error percentage (FEP) of DCCP buffer to decide if it has to increase, maintain or decrease its bitrate. We present here results for ten flows without gap. For better visualisation, we show buffer *success* percentage, which is simply 1 – buffer failure percentage.

IV.A.1 Ten flows without gap

In this test ten flows are running and using the available bandwidth. Fig. 4 shows the results for the first flow of this test. It can be seen that when the buffer success rate is very low the quality chosen by VAAL is very low (e.g. at the second 38). On the contrary, when the failure rate is null, the quality is increased slowly (switched to the available higher bitrate every two seconds, e.g. at seconds 40, 42 and 44). It can be noticed also that the quality is often between 1Mb/s and 512Kb/s. The reason is that the available bandwidth is shared between ten flows which reduces significantly the part of each flow. Another thing is the frequency of quality switching : the quality varies often depending on the available bandwidth. In this way VAAL insures that the quality is a function of the bandwidth. The greater the available bandwidth, the better the quality. Finally, there is no best fixed quality, so the adaptation is useful.

IV.B Adaptation performance

In order to find out if the adaptation is useful, we compare VAAL with the same application but without adaptation. We
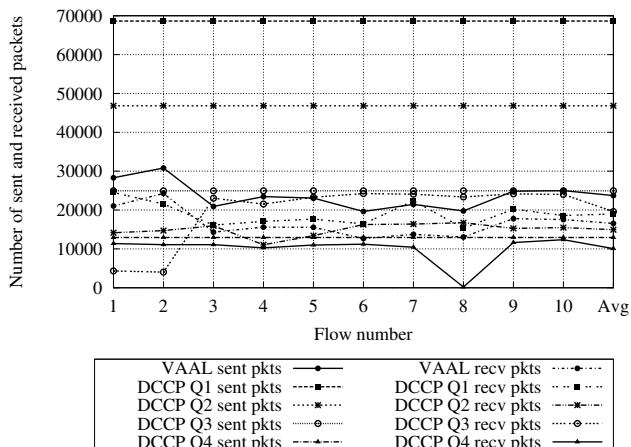
FIGURE 5 – Comparison of number of sent and received packets for ten flows without gap.

FIGURE 6 – Comparison of number of sent and received packets for ten flows with gap.

consider that if a new method is able to maximise the number of received packets while minimising the difference between the number of sent and received packets, it will ameliorate the received video quality. Thus, we compare on the one hand the number of received packets, and on the other hand the number of lost packets, in average.

The results for VAAL are taken from the same tests done before. We also did the same test series for DCCP without video adaptation, separately for each of the four qualities (Q1=3Mb/s, Q2=2Mb/s, Q3=1Mb/s and Q4=512kb/s). In the following figures, the x axis represents the flow number, followed by their average. The number of sent and received packets is put on the y axis (and they use the same point type to distinguish them more easily).

Note that, even if all the curves are put on the same graph, the execution is done at different times. Also, even if the curves use lines for better visualization, the flows are independent.

IV.B.1   Ten flows without gap

The result of this test is shown in Fig. 5. It can be seen that the average number of received packets for VAAL and DCCP without video adaptation is very close to each other. On the other hand, when the number of generated and dropped packets is taken into account we can conclude that DCCP without video adaptation (quality Q1 and Q2) performs very bad because of their high rate of dropped packets (e.g. about 50000 dropped packets out of 70000 for 3Mb/s, and 25000 out of 47000 for 2Mb/s). VAAL is clearly better. The reason of this high number of dropped packets on DCCP buffer is that the rate of generated packets at application layer is much higher than the network capacity. This high rate of lost packets affects considerably the video quality at the receiver side (high number of dropped packets leads to poor video quality on the receiver side).

IV.B.2   Ten flows with gap

These tests allow to compare the performance in a dynamic situation where the number of flows varies over time (the num-
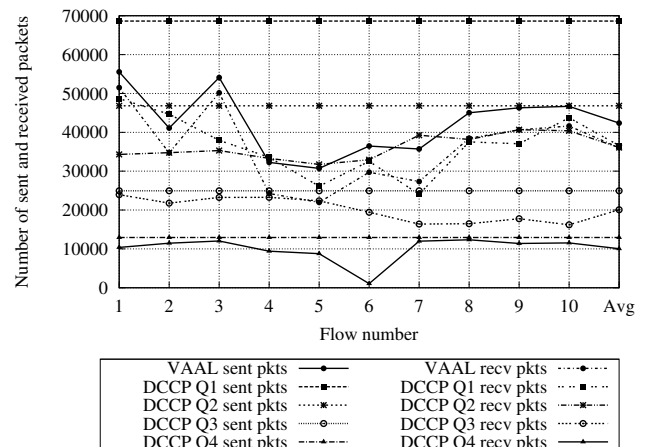
ber of concurrent flows vary between 1 and 6 at any moment). As already specified, there are about thirty seconds between the beginning of each two consecutive flows. Results are shown in Fig. 6. Same conclusion can be made : with video adaptation (VAAL) the bandwidth is efficiently used. For example, in Fig. 6, flows number 1, 2 and 3 use often the highest quality because they have enough bandwidth. Flows number 4, 5, 6 and 7 have less bandwidth (because the number of flows is greater during their life span), so they use lower quality and so on. In the other case, without adaptation, packets are always generated at the same rate, which does not take into account the available bandwidth, and it will risk losing a high number of packets (as seen previously), even before being sent on the network.

Another comparison between Fig. 5 and 6 shows that when bandwidth is shared between a smaller number of concurrent flows, application can chooses a better quality. VAAL knows to find the nearest best quality in both cases.

V   CONCLUSIONS

This paper presents a simple but powerful method (VAAL) to adapt the content of video streaming using the buffer overflow method on the server at the application layer. Intuitively, this method should lead to much better video streaming performance. The real experiments confirm this hypothesis, i.e. the bitrate used during the adaptation is shaped by the available network bandwidth, and it generally leads to either much fewer network losses or avoids a under-utilisation of the network bandwidth. Moreover, the use of a transport protocol (DCCP in our implementation) with a congestion control (TFRC) guarantees the TCP-friendliness of our method, and TFRC makes it video streaming friendly.

Future works will include to implement and test other similar methods, and will use also video quality metrics. Our final goal is to show that content adaptation on server application is the most appropriate method, not only in performance terms but also in implementation and practical terms, to cope with dy-

namic network bandwidth in cases such as video conferencing and small size video servers.

ACKNOWLEDGEMENTS

RÉFÉRENCES

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP : A Transport Protocol for Real-Time Applications," RFC 3550, Internet Engineering Task Force, Jul. 2003.

[2] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," RFC 4340, Mar. 2006.

[3] S. Floyd, E. Kohler, and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) congestion control ID 3 : TCP-friendly rate control (TFRC)," RFC 4342, Mar. 2006.

[4] W. Ramadan, E. Dedu, and J. Bourgeois, "EcnLD, ECN loss differentiation to optimize the performance of transport protocols on wireless networks," in *International Conference on Ultra Modern Telecommunications & Workshops (ICUMT), WMCNT workshop*, ser. 1. Saint Petersburg, Russia : IEEE, Oct. 2009, pp. 1–6.

[5] Y.-C. Su, C.-S. Yang, and C.-W. Lee, "Optimal FEC assignment for scalable video transmission over burst error channel with loss rate feedback," *Signal Processing : Image Communication*, vol. 18, no. 7, pp. 537–547, Aug. 2003.

[6] E. GÃ¼rses, G. B. Akar, and N. Akar, "A simple and effective mechanism for stored video streaming with TCP transport and server-side adaptive frame discard," *Computer Networks*, vol. 48, pp. 489–501, Jul. 2005.

[7] ÃrpÃ¡d HuszÃ¡k and S. Imre, "Selective retransmission of MPEG video streams over IP networks," in *International Symposium on Communication System Networks and Digital Signal Processing (CSNDSP)*, ser. 5, Patras, Greece, Jul. 2006, pp. 125–128.

[8] T. Haukaas, "Rate adaptive video streaming over wireless networks," Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway, Jun. 2007.

[9] P. A. Chaparro, J. Monteiro, C. T. Calafate, J. Alcober, J.-C. Cano, and P. Manzoni, "Supporting scalable video transmission in MANETs through distributed admission control mechanisms," in *Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, ser. 18, M. Danelutto, J. Bourgeois, and T. Gross, Eds. Pisa, Italy : IEEE, Feb. 2010, pp. 238–245.

[10] J. Khan and R. Zaghal, "Symbiotic rate adaptation for time sensitive elastic traffic with interactive transport," *Computer Networks*, vol. 51, no. 1, pp. 239–257, Jan. 2007.

[11] A. Balk, D. Maggiorini, M. Gerla, and M. Y. Sanadidi, "Adaptive MPEG-4 video streaming with bandwidth estimation," in *International Workshop on Quality of Service in Multiservice IP Networks*, ser. 2, M. A. Marsan, G. Corazza, M. Listanti, and A. Roveri, Eds. Milano, Italy : Springer, LNCS 2601, Feb. 2003, pp. 525–538.

[12] M. I. Kazantzidis, "Adaptive multimedia in wireless IP networks," Ph.D. dissertation, University of California, Los Angeles, USA, 2002.

[13] N. Ozbek, B. Gorkemli, A. M. Tekalp, and T. Tunali, "Adaptive streaming of scalable stereoscopic video over DCCP," in *IEEE International Conference on Image Processing (ICIP)*, ser. 14, vol. 6, San Antonio, TX, USA, Sep. 2007, pp. 489–492.

[14] N. Wakamiya, M. Murata, and H. Miyahara, "TCP-friendly video transfer," in *Internet Quality and Performance and Control of Network Systems*, A. L. Chiu, F. Huebner, and R. D. van der Mei, Eds., vol. 4211, no. 1. SPIE, 2001, pp. 25–35.

[15] S. Linck, E. Mory, J. Bourgeois, E. Dedu, and F. Spies, "Video quality estimation of DCCP streaming over wireless networks," in *Euromicro Conference on Parallel, Distributed and Network-based Processing*, ser. 14. MontbÃ©liard, France : IEEE, Feb. 2006, pp. 405–412.

[16] A. Lie and J. Klaue, "Evalvid-RA : trace driven simulation of rate adaptive MPEG-4 vbr video," *Multimedia Systems*, vol. 14, no. 1, pp. 33–50, Jun. 2008.

[17] ITU-T, *Opinion model for video-telephony applications*, International Telecommunication Union Recommendation G.1070, Apr. 2007.

A EXPERIMENT PARAMETERS

The real network parameters used for the experiments were the following :

Parameter name	Parameter value
Experiments place	In building
Packet size	500 Bytes
PC1 (sender) wired card product	82567LM-3
PC1 (sender) wired card technology	Gigabit Connection
PC1 (sender) wired card vendor	Intel Corporation
Wired bandwidth	100Mb/s
PC2 (receiver) wireless card product	BCM4311
PC2 (receiver) wireless card techn.	802.11b/g
PC2 (receiver) wireless card vendor	Broadcom Corporation
PC1&2 OS	Linux (Ubuntu 64bits)
PC1&2 OS kernel	2.6.31 generic
DCCP	Included in the kernel
Access point product	LINKSYS
Access point technology	Wireless-G
Wireless bandwidth	54Mb/s
Distance (AP ↔ PC2)	50cm
Shared with other users	no

Laboratoire d'Informatique de l'université de Franche-Comté
UFR Sciences et Techniques, 16, route de Gray - 25030 Besançon Cedex (France)

LIFC - Antenne de Belfort : IUT Belfort-Montbéliard, rue Engel Gros, BP 527 - 90016 Belfort Cedex (France)
LIFC - Antenne de Montbéliard : UFR STGI, Pôle universitaire du Pays de Montbéliard - 25200 Montbéliard Cedex (France)

http://lifc.univ-fcomte.fr