Video quality estimation of DCCP streaming over wireless networks

Sébastien Linck, Emmanuel Mory, Julien Bourgeois, Eugen Dedu, François Spies
Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC)
CNRS FRE 2661
BP 21126 25201 Montbéliard Cedex, France
{FirstName.LastName}@pu-pm.univ-fcomte.fr

Abstract

This paper describes a streaming architecture simulation model above Network Simulator 2 (NS2) which allows to define specific transport properties. Multimedia contents are specific because they are time-dependent and they can undergo small deterioration if necessary. We simulate such a congestion control that has the ability to decrease the multimedia quality in case of network congestion in order to decrease packet losses and packet delivery delays. We integrate this video congestion control inside DCCP (Datagram Congestion Control Protocol) and TFRC (TCP Friendly Rate Control). The transcoding of the multimedia contents is realized thanks to the NetMoVie simulation model which is an RTP mixer. We compare the adaptive transport solution to the classic transport solution without any adaptive mechanism. The Peak Signal-to-Noise Ratio (PSNR) of the received multimedia contents is measured and compared for better visualization.

1. Introduction

At the transport level, several solutions exist for multimedia streaming over Internet. Some of them do not use any congestion control, such as RTP/UDP, others use control congestion for static files (which are the same all the time), such as HTTP/TCP. Even if more appropriate propositions have been put forward, they have not replaced the current solutions. DCCP is useful because it offers internal mechanisms which allow to define new congestion control strategies. Indeed, DCCP allows to implement and to compare strategies adapted to the transport of multimedia contents. One of these strategies is the adaptation of a video to the available bandwidth. This allows the stream to respect real time constraints. This kind of operation can only be realised by a mixer such as defined in [19].

Being able to simulate a video on demand (VoD) architecture, comprising a server, a mixer and a client, open up real possibilities for the optimization and comparison of elaborate strategies in a real context of flow concurrence. The congestion control used by the video must be TCP-friendly, use the allocated bandwidth as best it can and have the best visual aspect. The originality of our test bed is that it evaluates the final quality of video streamed into NS2. This is done by integrated new modules of real content streaming and PSNR (Peak Signal-to-Noise Ratio) calculation into NS2.

All these properties can be extracted from our simulation model of multimedia streaming and this allows to give more accurate results when analyzing congestion control protocols for example.

The article is organized as follows. Section 2 gives some background information on wireless multimedia streaming and on transport protocols. Section 3 describes our simulation test bed. Section 4 studies two examples of simulations. Section 5 presents other work in the same area. The last section concludes the article and expounds some of our future work.

2. Background

2.1. Multimedia streaming and adaptation

2.1.1. Wireless multimedia streaming. RTP (Real-time Transport Protocol) and RTCP (Real Time Control Protocol) [19] are now the standard of streaming multimedia contents. RTP transmits the data while RTCP controls the RTP stream. Between the RTP server and the RTP client, two intermediate systems can be placed: the mixer and the translator.

The mixer receives RTP packets, possibly changes the data, combines the packets and then forwards new RTP packets. A mixer can modify the data, for example, it can change the quality of the sound.

The translator forwards RTP packets leaving their synchronization source identifier intact.

Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'06)
1066-6192/06 $20.00 © 2006 IEEE

2.1.2. Content adaptation. Multimedia content streaming over wireless networks is facing four challenges: mobility, shared, limited and variable bandwidth. When a client moves, he may change wireless access points (Base Transceiver Station or WLAN Access Point) and multimedia contents must be redirected as quickly as possible. As the bandwidth always fluctuates, multimedia contents have to be adaptable to the available bandwidth at a given moment. This quality modification must be as smooth as possible in order to avoid wide quality variations.

Moreover, clients use a wide range of multimedia devices and a client connected to a GPRS network with a smart phone will not be able to visualize the same multimedia content as a client connected to an 802.11g network with a laptop. This heterogeneity implies that one must be able to stream a wide variety of multimedia contents adapted to each case.

In the case of RTP streaming, the adaptation can be done either in a server or in a mixer. The best place to perform the content adaptation is the mixer because it can be placed the nearest possible to the client. This is useful to avoid latency when adapting the contents.

2.2. Transport (DCCP/TFRC/RTT)

Two transport protocols dominate nowadays: TCP and UDP. DCCP [12] is a recent transport protocol (it is being standardized by IETF) sharing characteristics from both of them: It has congestion control mechanisms like TCP, and it is unreliable like UDP.

DCCP separates the transport of packets from the flow congestion control (CC). Each flow can choose the most appropriate CC. Currently, two CC mechanisms are provided: TCP-like [8] and TFRC [9].

TCP-like, resembling TCP, uses a congestion window. The window increases when there is no packet loss, and decreases by half when there are losses. The congestion window as well as the bandwidth have abrupt changes and are not appropriate for video streaming.

TFRC uses an equation for the bandwidth. The equation is used regularly, for example each RTT. It allows moderate bandwidth changes and is more appropriate to video streaming. Therefore, this is the solution chosen for our approach. The equation, given in [9], is based on TCP Reno CC:

$$bw_{est} = \frac{s}{RTT\sqrt{\frac{2p}{3}} + RTO \times 3p(1+32p^2)\sqrt{\frac{3p}{8}}}$$

where s is the packet size, RTT is the round trip time in seconds, RTO is the retransmission timeout and p is the loss event rate (number of lost packets divided by number of sent packets).

In the implementation of DCCP in NS2 [14], the RTO is set to 4*RTT, as suggested in [9]. The equation becomes:

$$bw_{est} = \frac{s}{RTT\left(\sqrt{\frac{2p}{3}} + 12p(1+32p^2)\sqrt{\frac{3p}{8}}\right)}$$

We discuss the importance of the parameters involved in the equation:

RTT If the RTT does not correspond to the real RTT, the estimated bandwidth is not accurate. In wireless networks, packet losses frequently appear without being caused by congestion. To cope with this, the 802.11 protocol uses ARQ (Automatic Repeat reQuest), sending a lost packet several times on the wireless channel until it is received. This article presents a method to eliminate the dead time in MAC retransmission.

p If the losses are not correlated to a congestion event, the estimated bandwidth is not accurate. As previously said, losses in wireless networks are generally not correlated to a congestion event.

2.3. 802.11 MAC ARQ

802.11 is a protocol which relies on ARQ (Automatic Repeat reQuest). After sending a packet, the network card awaits acknowledgement from the receiver's network card. If it does not arrive, then the sender's network card will consider that the packet was lost and retransmit it. There is a limit in the number of retransmissions.

Packet loss is frequent in wireless networks. It generally appear when an external interference occurs in the network, but also when the mobile clients exit from the area covered by the network. Interference is known to be temporary and to appear at random times.

A detailed explanation of the 802.11 standard can be found in [6].

3. Simulation testbed

3.1. General architecture

The general architecture of our simulation test bed is composed of two main parts: the mixer and the client.

Figure 1 presents the model of the mixer as it has been modeled into NS2. Three kinds of video data can be used as input of the mixer:

- Generated data: The mixer generates its own data by using various algorithms. The distribution of the various kinds of images (I, P and B) is given by Gismo [11].

Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'06)
1066-6192/06 $20.00 © 2006 IEEE

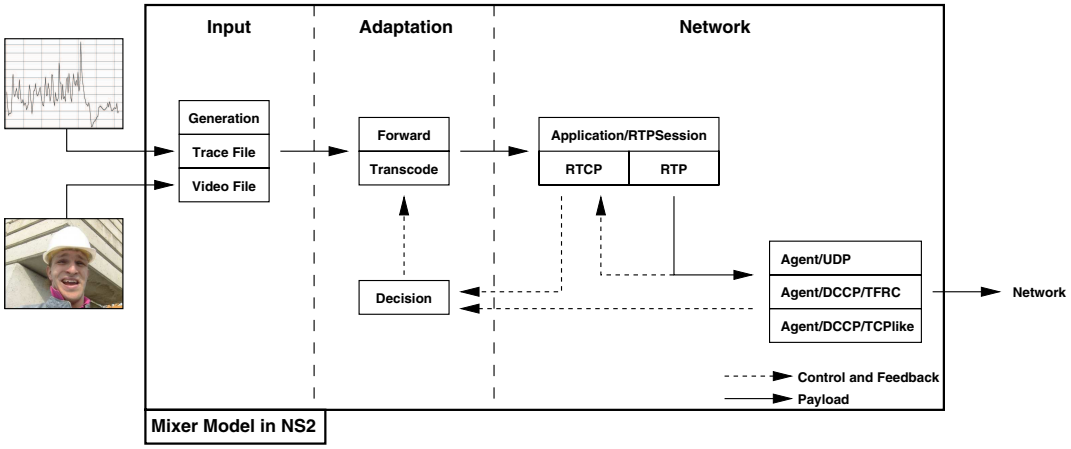


Figure 1. Mixer model in NS2.

- Real traces: During a real transmission, the size of packets sent over the network is analyzed and the characteristics of the packets are stored so that they may be used in NS2.
- Real streaming: This mode makes it possible to use real videos in NS2. The packetization is carried out by the mixer. The packets are sent through the different NS2 modules and the video is really transmitted between the server and the final client.

The input is sent to the core of the mixer, that is to say to the adaptation module, which decides either simply to forward the stream or to transcode it in order to fit the available bandwidth better. The packets are sent to the network module which comprises an RTPSession application which manages the RTP and RTCP agents, and the transport agent, for example, UDP or DCCP.

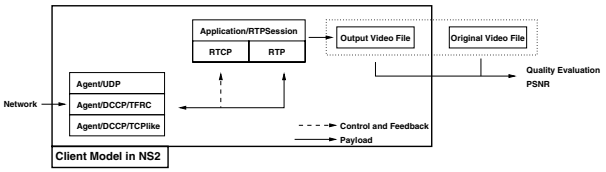


Figure 2. Client model in NS2.

Figure 2 presents the model of the client as it has been modeled into NS2. The client receives the packets from the network, and he can reconstruct the video exactly as if he had played it. This video is output to a compressed video file which can be compared with the original video. Finally, the PSNR can be calculated according to see exactly what the effect of losses or jitter is on he resulting video.

This simulation test bed allows us to test different strategies to stream multimedia contents with various transport protocols. Besides, it also allows us to see the effects of the network problems on the streamed contents clearly. Indeed, during multimedia streaming the lost packets will not have the same effect on the video quality. Some packets will seriously affect the visual quality of the video whereas others will not. This difference depends on various parameters like the type of lost packet, the time the packet is lost in the group of pictures (GOP), etc. In fact, if the last P-frame of a GOP is lost, the quality will not be affected because, just after this image, an intra image will be decoded. As the last P-frame of a GOP and the first I-frame of the following GOP do not have any time dependence, only one frame will be damaged. Another example is when a packet is lost on a P-frame just before a camera movement. The resulting image will be damaged but the camera movement will delete this error. That is why it is necessary to calculate the PSNR and not just to count the lost packets to evaluate the resulting video quality and this is the aim of our test bed.

3.2. Extensions to NS2

Compared to the original version of NS2, we have done the following extensions:

- Currently, NS2 does not contain DCCP. Mattsson's patch [14] has been used for the simulations.
- However, the patch does not work over wireless links. We have modified this patch[1] in order to work with wireless links.

[1]The new patch can be found at `http://lifc.univ-fcomte.fr/~dedu/ns2/`.

Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'06)
1066-6192/06 $20.00 © 2006 IEEE

- We have also modified the DCCP code to add and to use the option that stores the MAC retransmission time, as shown in section 4.

3.3. The mixer

3.3.1. Architecture of the mixer. An RTP mixer [19] receives RTP packets from one or more sources, possibly changes the data format, combines the packets in some manner and then forwards a new RTP packet.

This mixer has been developed in the NetMoVie project [2], which is part of a larger project named MoVie. The mixer is not only RTP-compliant but has also extended functionalities like on-the-fly transcoding or adaptability of the contents to the constraints of the network.

It is located the closest to the client in order to react as soon as possible to the variation of the bandwidth. As the clients are connected to a wireless network, the mixer should be ideally located in the bridge between the wired and the wireless network.

The mixer modifies the transmitted data in order to adapt them to the client or to optimize the available bandwidth. The mixer is also able to deal with several types of coding, like hierarchical video or MPEG codings ones.

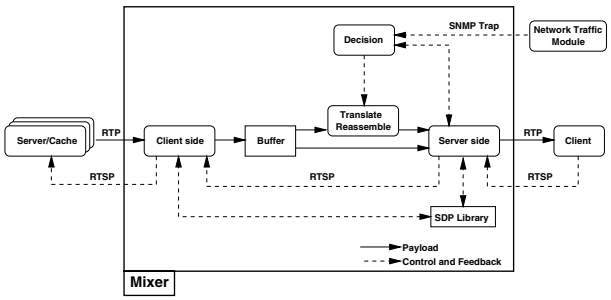


Figure 3. Internal architecture of our mixer.

3.3.2. The modules of the mixer. Figure 3 shows the various modules of the mixer:

Client side This module allows the mixer to be connected to the video servers.

Server side This module consists of the implementation of a complete RTP/RTSP server. The mixer is seen like a server from the clients' point of view.

Buffer The buffer is used to store a certain number of images before beginning the streaming to the client.

Transcode/reassemble This module is the element which makes it possible to adjust the quality of the stream according to the various constraints which act on the transmission. One of the major points of this module is the choice of changing the policy of adaptation.

Decision This module takes into account all the parameters which are given to it: information feedback from the server module (RTCP reports, for example) or information on the available bandwidth coming from the network module. Afterwards, it chooses the most appropriate video for the client according to the available bandwidth and the available video quality from the servers.

3.3.3. Description of a typical session.

1. A client connects to the mixer by the intermediary of the server module and requests the visualization of the video.
2. The request is transmitted to the decision module which will ask the global architecture for the available quality for the required video.
3. The decision module will choose the video whiwh is the most adapted to the client's characteristics and to the constraints of the network.
4. The client module requests the chosen video from the selected server. In the best case, a video the quality of which meets the needs, is directly available in the system and no transcoding or adaptation is necessary. If it is not the case, the decision module chooses the right transcoding method.
5. As soon as the stream is received by the server module, it is sent directly to the client.
6. If a quality change is necessary, the decision module asks the video server if a more adapted video is available. But in order to adapt the quality as soon as possible, it asks the transcoding module to change the quality of the stream while waiting for a new one.
7. As soon as the new video is available, the mixer stops the transcoding and streams the new video.

4. Case study

The Network Simulator [15], version 2.28, frequently used in research, is used for simulations.

4.1. NS2 scenario

4.1.1. Propagation models in NS2. Currently, three wireless propagation models are implemented in NS2 [16]: Free space, Two ray ground and Shadowing. The first two

Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'06)
1066-6192/06 $20.00 © 2006 IEEE

models are of "all or nothing" type: If distance d between the mobiles is smaller than a certain value, all packets sending are received. If d is greater, no packet is received. These are not appropriate in our case, since we need retransmissions from time to time.

In the shadowing model, packets are always received for $d \leq s1$, always lost for $d \geq s2$ and received with a probability for $s1 < d < s2$. This last one is the most suitable because it resembles more to real wireless links.

4.1.2. Simulations. A wired streaming video server, an access point (AP), and a mobile streaming client are created for the simulation. Note that both AP and mobile use retransmission times. This simulation corresponds, to a man walking on the street while watching News on Demand (NoD). The mobile moves according to the following scenario (figure 4):

1. The mobile moves away linearly from the AP.
2. At $t = 100s$, it stops moving and stays motionless during $50s$.
3. Then it goes back to its initial position and the simulation ends at $t = 250s$.

Between the stillness time, the mobile is located in the edge of the covered area of the AP, where most of the retransmissions appear.

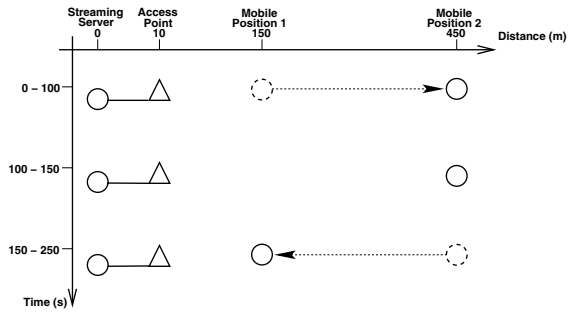


Figure 4. The simulation scenario.

4.1.3. Test protocol. We transfer three times the same video with the same movement scenario. Only the transport protocol differs from one simulation to another. First, we use RTP over UDP, the original video streaming transport protocol. Then we use DCCP/TFRC with video adaptation and at last we propose to study: DCCP/TFRC with RTT modification.

4.2. Results

4.2.1. Throughput. Figure 5 shows three stages of the mobile movement. During the first $50s$ the throughput is smaller than the available bandwidth because the video is made of fixed images or less movements. After this, the throughput better fits to the available bandwidth.

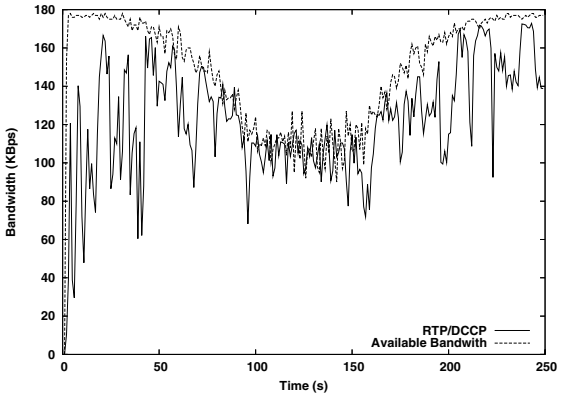


Figure 5. Throughput comparison.

4.2.2. Packet loss. In figure 6(a), during the $250s$ of RTP/UDP video transmission, 2063 packets are lost. With DCCP/TFRC, this number is reduced to 180 packets only. In the UDP case, most of these losses appear during the stillness time, when the available bandwidth is smaller than the video bitrate.

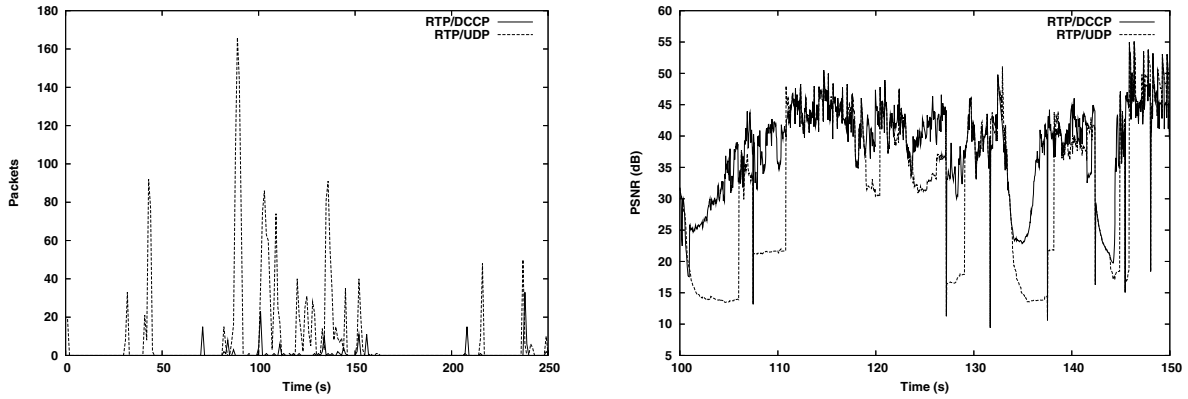
4.2.3. PSNR. To compare the videos received by the client, we use the Peak Signal-to-Noise Ratio (PSNR) which is a standard video quality estimation. For each simulated transport protocol, the original video and the received one are compared in figure 6(b). Because of the TFRC and the video adaptation, we show that the video read by the client on his mobile player has a better quality. The received videos are available on-line at `http://mortimer.pu-pm.univ-fcomte.fr/pdp2006/`.

4.3. RTT modification in DCCP

Due to the unreliability of wireless propagation, 802.11 allows MAC retransmission. Hence, 802.11 transforms a network with losses and predictable delay into a network with no losses and variable delay. On the other hand, packet losses are generally due to interferences. As they are supposed to be *temporary*, the RTT should not be influenced by them. We therefore propose a mechanism to remove the time lost by these MAC wireless retransmissions.

Our solution for DCCP/TFRC is based on the same principle as in [5] for TCP Vegas. An option, called `rets`, is added to DCCP header. Each wireless network card has a timer. The timer is initialised to the value of the `rets` field of the packet for the first transmission of a packet (after the

Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'06)
1066-6192/06 $20.00 © 2006 IEEE



(a) Packet loss.

(b) PSNR comparison.

Figure 6. Packet loss and PSNR comparison results.

backoff). Each time it sends a packet on the wireless link, the value of the timer is stored in the `rets` field. Thus the timer reflects the *exact* time loss due to retransmissions. When the source receives a packet, it takes the appropriate action, for example it subtracts the value of the `rets` option from the RTT of the packet.

The remainder of this section details this mechanism in DCCP.

4.3.1. Retransmission time computing. As specified before, each network card has a timer. Each time a new packet is processed, its `rets` is used to initialise the timer. This allows to take cumulated time losses into account, for example in the case of an ACK packet already containing the lost time value of the corresponding *data* packet. During each (re)transmission, the value of the timer is stored in the `rets` option of the packet.

The MAC-level fragmentation does not influence our mechanism. Indeed, when an IP packet is fragmented the time loss is null if each fragment arrives at destination without retransmission.

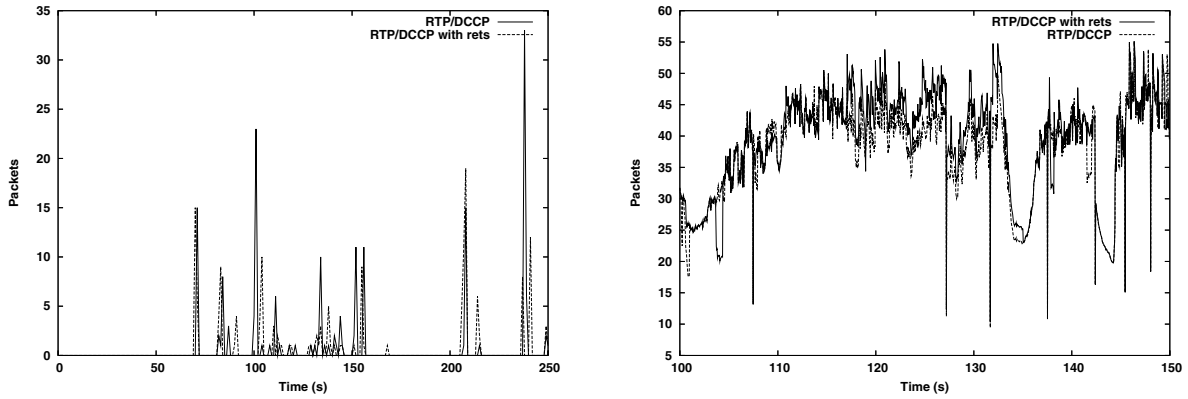
4.3.2. Using the time information. Our `rets` DCCP option has only one field, containing a time value. If the field has 2 bytes and the measurement unit is the time slot $t_s = 20\mu s$ of 802.11 backoff calculation, then the field will overflow at a time $t = 65536 \times t_s = 65536 \times 20\mu s \approx 1.3s$. In the 802.11b standard the maximum Contention Window (CW) is 1023 packets, hence 2 bytes are sufficient. If the field has 4 bytes, it will overflow after $t \approx 65536 \times 1.3s \approx 1$ day, which is largely sufficient. The source sets this field to zero. During the trip, the field may be modified by the bridge wired-wireless.

This mechanism allows incremental deploying. It gives useful values only if the sender, the receiver and the AP know about it. If the sender or the client is not aware of this option, it is not activated because of the DCCP Option negotiation [12]. Otherwise, it adds the option and sets the field to 0. If the AP and/or the receiver do not know this option, the sender either receives no option, or an option with value 0, which does not change anything either.

4.3.3. Actions taken. Contrary to [5], in DCCP/TFRC it is up to the receiver to take appropriate actions for congestion control. For instance, we propose that a DCCP/TFRC receiver use the corrected RTT in its formula to estimate the RTT. thanks to this estimation the sender will use a more adapted sending rate. This last information sent to a video server would allow to better know the network bandwidth and so to send the appropriate transcoded flow.

4.3.4. Results. In figure 7(a) there are fewer packet drops on the curves including the RTT modification than on the RTT/DCCP curve. Thanks to this modification, the number of packet losses decreases by 10%.

Figure 7(b) shows that DCCP with RTT modification (RTP/DCCP with `rets`) is mainly equivalent to the other curve except at the time 100 sec and between 115 sec and 135 sec. In this interval, the modification offers an improvement of around 10% because of packet losses at time 115 sec which lead to a serious temporal propagation of an error.



(a) Packet loss.

(b) PSNR comparison.

Figure 7. Packet loss and PSNR comparison results.

5. Related work

5.1. Video streaming using DCCP/TFRC

DCCP adds congestion control to UDP and it has successfully been used in video streaming. [24] presents an implementation of TFRC in Linux, a codec and a video conferencing system with low latency, combining these elements. It divides the video system in three components: the codec, the DCCP module, and the algorithm deciding the video quality to use. The variables used for quality changing are: resolution, JPEG quality and frame rate.

5.2. TFRC in wireless links

TFRC uses a formula based on TCP Reno [9]. Therefore, over wireless links it has the same drawbacks as TCP, the most notable is that a packet loss is considered as a congestion event. Several approaches have been proposed to adapt TFRC to wireless links.

[3] analyses losses in wireless links. The authors propose multiple connections for a video stream and deduce the following rule: "Keep increasing the number of connections until an additional connection results in an increase of end-to-end RTT or packet loss rate". Based on the RTT variation, the number of connections n is increased by α/n or decreased by β, where α and β are constant.

5.3. Removing MAC retransmission times

The negative effect of MAC retransmission on TCP is treated in [17], where the TCP connection between a wired machine and a wireless machine is divided into two TCP connections by the AP located in the middle. The AP buffers data received from the wired end and retransmits it to the wireless end if it was not received. Also, the time spent in the AP is subtracted from the TCP timestamp option. Contrary to the method proposed in this article, the time spent at the AP is not accurate (the timestamp granularity depends on the source machine [10]), the AP needs to buffer data and it works only with the TCP timestamp option.

5.4. Video transcoding

Three categories of video transcoding that modify the bitrate of the streamed video have been developed [13]. The first one is referred as closed loop transcoding or Cascaded Pixel Domain Transcoder (CPDT) [23]. The video is completely decoded, possibly modified and then encoded, this is the solution chosen for our mixer. The second category called Open Loop Transcoding (OLT) [7] do not completely decodes the stream but stops to the DCT phase. This solution saves CPU time but the resulting quality is not as good as the CPDT solution. Finally, the third one, is an intermediate solution that pushes the decoding process deeper than OLT. This category is named DCT Domain Transcoder (DDT) [25] and an implementation has been realized [18].

It is also possible to transcode by changing the resolution of the video [21, 1, 20] or by modifying the image frequency [22, 4].

6. Conclusion and future work

This paper proposes a complete video streaming architecture which includes a mixer combining a server and a

Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'06)
1066-6192/06 $20.00 © 2006 IEEE

client, and a mobile client. This paper demonstrates that this model can help to optimise and to implement a dedicated congestion control protocol.

The measured improvements of our solution compared to the classical video streaming one are significant. Two types of results have been presented, for the network (throughput and packet loss) and for the visualisation of received video (PSNR). The PSNR results of our solution is really better than the classical solution RTP/UDP.

Wireless networks are recent technologies and most of the current protocols, especially the transport ones, do not include any of the specificities of the MAC layer of WiFi network. We want to propose new solutions of control congestion adapted to video streaming which take into account the constraints of wired and wireless networks. Thanks to our simulation model, it will be faster and easier to propose new efficient strategies for delivering multimedia content.

7. Acknowledgements

This work is a project funded by: EU, french ministry of research, DRIRE, Franche-Comt´e Council and CAPM.

References

[1] N. Bjork and C. Christopoulos. Transcoder architecture for video coding. In *IEEE Trans. Consum. Electron. vol. 44 no. 1*, pages 88–98, 1998.

[2] J. Bourgeois, E. Mory, and F. Spies. Video transmission adaptation on mobile devices. *Journal of Systems Architecture*, 49(10-11):475–484, Nov. 2003.

[3] M. Chen and A. Zakhor. Rate control for streaming video over wireless. In *INFOCOM*, volume 2, pages 1181–1190, Hong Kong, Mar. 2004. IEEE Computer and Communication Societies Press.

[4] M.-J. Chen, M.-C. Chu, and C.-W. Pan. Effi cient motion-estimation algorithm for reduced frame-rate video transcoder. In *IEEE Transactions on Circuits and Systems for Video Technology Vol. 12, No. 4*, Apr. 2002.

[5] E. Dedu, S. Linck, and F. Spies. Removing the MAC retransmission times from the RTT in TCP. In M. Al-Akaidi and L. Rothkrantz, editors, *Euromedia Conference, Workshop on Distributed Multimedia Databases and Multimedia Adaptation*, pages 190–193, Toulouse, France, Apr. 2005. Eurosis.

[6] D. Dhoutaut. *tude du standard IEEE 802.11 dans le cadre des rseaux ad hoc : de la simulation l'exprimentation*. PhD thesis, INSA, Lyon, Dec. 2003.

[7] A. Eleftheriadis and B. Anastassiou. Constrained and general dynamic rate shaping of compressed digital video. In *International Conference on Image Processing (ICIP)*, 1995.

[8] S. Floyd and E. Kohler. Profi le for DCCP congestion control ID 2: TCP-like congestion control. IETF draft, available at `http://www.ietf.org/internet-drafts/draft-ietf-dccp-ccid2-10.txt`, Mar. 2005.

[9] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol specifi cation, Jan. 2003. RFC 3448.

[10] V. Jacobson, B. Braden, and D. Borman. TCP extensions for high performance, May 1992. RFC 1323.

[11] S. Jin and A. Bestavros. Gismo: a generator of internet streaming media objects and workloads. *ACM SIGMETRICS Perform. Eval. Rev.*, 29(3):2–10, 2001.

[12] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). IETF draft, available at `http://www.ietf.org/internet-drafts/draft-ietf-dccp-spec-11.txt`, Mar. 2005.

[13] Z. Lei and N. Georganas. Rate adaptation transcoding for precoded video streams. In *Proceedings of ACM Multimedia*, Juan-les-Pins, France, Dec. 2002.

[14] N.-E. Mattsson. A DCCP module for ns-2. Master's thesis, Luleå University of Technology, Sweden, May 2004. Available at `http://www.ns.dccp.org/thesis.htm`.

[15] Network simulator — ns-2. `http://www.isi.edu/nsnam/ns/`.

[16] The NS manual. `http://www.isi.edu/nsnam/ns/ns-documentation.html`.

[17] K. Ratnam and I. Matta. Effect of local retransmission at wireless access points on the round trip time estimation of TCP. In *The 31st Annual Simulation Symposium*, pages 150–156, Boston, MA, USA, Apr. 1998.

[18] S. Roy and B. Shen. Implementation of an algorithm for fast down-scale transcoding of compressed video on the itanium. In *Proceeding of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2002.

[19] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications, 2001. RFC 1889.

[20] T. Shanableh and M. Ghanbari. Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats. In *IEEE Transactions on Multimedia, Vol. 2, No. 2*, June 2000.

[21] J. Xin and C.-W. L. M.-T. Sun. Digital video transcoding. In *Proceedings of the IEEE, Vol. 93, No. 1*, Jan. 2005.

[22] J. Youn, M.-T. Sun, and C.-W. Lin. Motion vector refi nement for high-performance transcoding. In *IEEE Transactions on Multimedia Vol. 1*, Mar. 1999.

[23] J. Youn, M.-T. Sun, and J. Xin. Video transcoder architectures for bit rate scaling of H.263 bit streams. In *ACM Multimedia*, pages 243–250, Nov. 1999.

[24] T. Young. High quality video conferencing. Honours project, University of Waikato, Hamilton, New Zealand, Dec. 2003.

[25] Q.-F. Zhu, L. Kerofsky, and M. B. Garrison. Low-delay, low-complexity rate reduction and continuous presence for multipoint videoconferencing. In *IEEE Transactions on circuits and systems for video technology Vol. 9 No 4*, June 1999.