2007 International Symposium on Nonlinear Theory and its Applications
NOLTA'07, Vancouver, Canada, September 16-19, 2007

Trajectory Tracking Control for Video Streaming Application

Lynda Zitoune and Hugues Mounier and Véronique Vèque

Institut d'Electronique Fondamentale, Bat 220
Université paris-Sud XI, 91405 Orsay cedex
Email : lynda.zitoune,hugues.mounier,veronique.veque@ief.u-psud.fr

Abstract—Based on control theoretical approaches, we develop a new adaptation mechanism for transmiting video streaming application over Internet. In order to satisfy the video requirements we propose to use a nonlinear control method that adjusts the source bite rate in accordance to the network resource availability. Our method relying on trajectory tracking control, ensures network stability and avoids source bit rate oscillation. It also provides low loss and bounds the transmission packet delay. Trajectory Tracking is a promising method to dynamically manage the network resources that can avoid stravaging the competing flows.

1. Introduction

Today, video streaming applications like Video on Demand (VoD) are the very popular applications running on the Internet. These applications have witnessed significant evolution over the past few years, from content delivery to progressive download. The first technology allows clients to download compressed videos and to play them back locally. The second, more flexible one, allows clients to watch the video as the content of files are being downloaded. In the second case, the traffic generated by such applications needs service guarantees, also known as Quality of Service guarantees in terms of bandwidth, loss, delay and jitter (delay variation) [1].

In order to satisfy the demands of these applications, an appropriate network service should be provided over Internet. The main challenge is to develop adaptive schemes aimed at dealing with the network resource variation, the clients heterogeneity and providing low loss and bounded packet delay. This can be achieved if the packets are controlled over the network, namely in the limit nodes. Several solutions are proposed for adapting streaming flows depending on where they are applied [2]. Receiver based adaptation scheme is the most popular. The video stream is encoded and stored in the server either in layered-coded video or in version-coded video. These pre-encoded videos are transmitted to the edge router. In the layered-coded scheme, the video is coded on a base layer and several enhancement layers. The edge router forwards the base layer to all clients and a subset (possibly all) of the enhancement layers to clients depending on their access rates [3]. In version-coded video strategy, multiple versions are created. The edge router forwards the appropriate version to its clients [2, 3]. In both cases, and depending on the clients resources availability, the edge router adds/removes enhancement layers (in layer-coded case) or replaces corrupted video frames by the corresponding video frames from the immediately lower version (in version-coded case). These adaptive strategies lead to drastic short-term changes, causing packets losses, delay violations and network behavior oscillation which affect the competing traffic. The development of effective forward error correction algorithms (FEC) and evolution of error concealment techniques allows to reduce the requirements on the packet losses, but delay requirements remain very strict [4].

In this paper, we propose a new method for controlling the edge router bit rate in order to smooth the short-term changes and to bound the packet delay. Our method is based on a nonlinear approach of theoretical control named *flatness based trajectory tracking* that stabilizes the general network behavior and ensures a dynamic adaptation of the video stream depending on resource availability.

In the rest of this paper, we describe our target environment and explain the functionality of our trajectory tracking control in section 2. Next, we introduce the development of the approach and it implementation in sections 3,4. In section 5, we present the simulation results and we conclude our work in section 6.

2. Target environment

Our target environment is a video server that plays back a video stream (on layered or version form) for many heterogeneous clients. The clients and the server are connected through the Internet. As depicted in figure (1), the functional configuration of a video streaming server is composed of two entities. A storage block stores video sequences represented by files (layers or versions), and a rate information block keeps information about the predefined rate of each sequence. This block enforces the server to transmit with a predefined rate according to the clients capacity and network resource availability [4].

At the receiver side, we assume that we are able to estimate the client buffer availability on a time interval. Our aim is to adapt the rate of the video stream to this buffer availability and to deal with the drastic short-term rate changes causing bursts and delay violation by smoothing these bursts. This buffering distribution allows a perceptual quality maximization while minimizing rapid, distur-

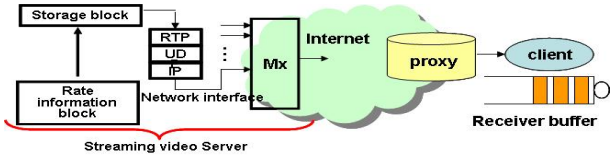

FIG. 1 – Target environment

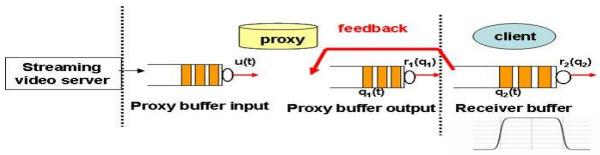

FIG. 2 – Network control model

bing changes in quality.

Because of the stored video has a predefined transmission rate which is not compatible with any adaptation control method, we propose to use an intermediate proxy. The proxy collects the video transmitted with its intrinsic rate and performs a rate adaptation of the video basing on the receiver feedback control. By applying our regulation on a proxy, we offer an adaptable, modular and protocol independent solution. By adaptable, we mean that the proxy can be reconfigured based on local characteristics of other clients. By modular, we mean that the proxy offers the ability to implement our algorithm, which is independent of signaling or multimedia transport protocols.

Proxy adaptive scheme was also proposed to deal with resource variation and clients heterogeneity. This solution consists in changing the transport protocol of a video or translating the video to another code, for example converting MPEG coded video to H.261 one. However, neither the problem of rapid disturbing changes nor resource adaptation have been solved. Thus, it is necessary to develop a simple transmission service able to follow the buffers variation in a smooth manner.

3. Trajectory tracking approach

As a matter of introduction to our approach, we consider the very simple case of a network with one streaming server, a single receiver interconnected by a proxy (figure 2). We suppose that the proxy collects the totality of packets generated by the server in its buffer input. These packets are served with the same rate as their incoming rate, denoted $u(t)$ and stored in the proxy output buffer (queue q_1). Packets stored in output buffer are released according to some service $r_1(q_1)$ to the receiver (queue q_2). At the receiver, the packets are played-back with some service bit rate $r_2(q_2)$.

Here we propose to control the proxy input bit rate $u(t)$ in order to respect the receiver buffer availability which we have modelled by a reference trajectory denoted as $q_{2r}(t)$ (figure 3). Thus, the feedback control is placed between the receiver buffer and the output buffer of the proxy. The controller adjust the $u(t)$ so that the packets released from

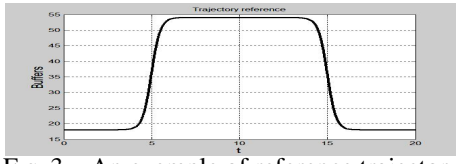

FIG. 3 – An exemple of reference trajectory

q_1 must be accepted in q_2. In order terms, the controller ensures that $q_2(t)$ tracks $q_{2r}(t)$, mainly when transition between lack/availabilty of buffers, to avoid losses, delay variation and bit rate oscillations.

Video streaming is identified as an application that generates enormous data quantities. For modelling this type of application we choose to use a fluid flow model which is the most appropriate for such bulk transfer.

4. Implementation of the tracking trajectory

4.1. Fluid Flow model

In the fluid flow paradigm, the physical evidence is that the rate of accumulation of packets in the buffer is the difference between the packet inflow rate and the packet outflow rate. So for the model (figure 2), we obtain two differential equations describing the length queues variations $(\dot{q}_1, \dot{q}_2)$ (1).

$$\begin{aligned} \dot{q}_1(t) &= u(t) - r_1(q_1(t)) \\ \dot{q}_2(t) &= r_1(q_1(t)) - r_2(q_2(t)) \end{aligned} \tag{1}$$

with

$q_1(t), q_2(t)$	: buffers size (packets).
$u(t)$	: proxy (output buffer) input bit rate (packets/sec).
$r_1(q_1(t))$	: proxy output service rate (packets/sec).
$r_2(q_2(t))$	: receiver play-back rate (packets/sec).

The positivity of the buffers queue lengths as well as their maximum capacity are considered by describing the outflow rates $r_1(q_1), r_2(q_2)$ in terms of the contents of the buffers q_1, q_2 respectively (see [5]).
We take $r_i(q_i) = \frac{\mu_i q_i}{a_i + q_i}$ which is (as demonstrated in [5]) a positive bounded function of the load q_i and a monotonically increasing one. The parameter μ may be interpreted as the maximal processing capacity of the router. This relation is obtained by supposing a linear relation between the residence time (or queueing delay) and the buffer queue length.

The model (1) is rewritten as (2) :

$$\begin{aligned} \dot{q}_1(t) &= u(t) - \frac{\mu_1 q_1(t)}{a_1 + q_1(t)} \\ \dot{q}_2(t) &= \frac{\mu_1 q_1(t)}{a_1 + q_1(t)} - \frac{\mu_2 q_2(t)}{a_2 + q_2(t)} \end{aligned} \tag{2}$$

Developing such a control scheme can be decomposed in two steps : 1. Design of the reference trajectory of the

so-called flat outputs (see subsection 4.2 below) ; off-line computation of the open loop controls. 2. Inline computation of the complementary closed loop controls in order to stabilize the system around the reference trajectories [6, 7].

Why is this two step design better suited than a classical stabilization scheme ? The first step obtains a first order solution to the tracking problem, while following the model instead of forcing it (like in a usual pure stabilization scheme). The second step is a refinement one, and the error between the actual values and the tracked references will be much smaller than in the pure stabilization case [7, 8].

4.2. Flatness Control

The model (2) is flat with $q_2(t)$ as a flat output. In other words, we get a complete parametrization of the system in terms of q_2 and of a finite number of its derivatives. Thus, $u(t)$, as well as $q_1(t)$ are nonlinear expressions of $q_2(t)$ and its derivatives, as explicitly demonstrated below :
The first equation of (2) gives :

$$u(t) = \dot{q}_1(t) + \frac{\mu_1 q_1(t)}{a_1 + q_1(t)} \tag{3}$$

The last equation of (2) yields :

$$q_1(t) = a_1 \frac{\dot{q}_2(t) + \frac{\mu_2 q_2(t)}{a_2+q_2(t)}}{\mu_1 - \dot{q}_2(t) - \frac{\mu_2 q_2(t)}{a_2+q_2(t)}} \tag{4}$$

which may be rewritten as :

$$q_1(t) = a_1 \frac{\dot{q}_2(t)(a_2 + q_2(t)) + \mu_2 q_2(t)}{(\mu_1 - \dot{q}_2(t))(a_2 + q_2(t)) - \mu_2 q_2(t)} \tag{5}$$

We assume the case of $M/M/1$ file for which $a_1 = a_2 = 1$ [5], so

$$q_1(t) = \frac{\dot{q}_2(t)(1 + q_2(t)) + \mu_2 q_2(t)}{(\mu_1 - \dot{q}_2(t))(1 + q_2(t)) - \mu_2 q_2(t)} \tag{6}$$

the first derivative of q_1 may be calculated as

$$\dot{q}_1(t) = \frac{\mu 1(1 + a_2)^2 \ddot{q}_2(t) + \mu_1 \mu_2 \dot{q}_2(t)}{(\mu 1(1 + a_2) - \dot{q}_2(t)(1 + a_2) + \mu_2 q_2(t)^2} \tag{7}$$

Replacing $q_1(t)$ and $\dot{q}_1(t)$ with their values in (3), we obtain :

$$\begin{aligned} u(t) &= \dot{q}_2(t) + \frac{\mu_2 q_2(t)}{1 + q_2(t)} \\ &+ \frac{\mu 1(1 + a_2)^2 \ddot{q}_2(t) + \mu_1 \mu_2 \dot{q}_2(t)}{(\mu 1(1 + a_2) - (1 + a_2)\dot{q}_2(t) + \mu_2 q_2(t))^2} \end{aligned} \tag{8}$$

Thus, for a reference trajectory $q_{2r}(t)$, the proxy input bit rate defined by the equation (9)

$$\begin{aligned} u(t) &= \dot{q}_{2r}(t) + \frac{\mu_2 q_{2r}(t)}{1 + q_{2r}(t)} \\ &+ \frac{\mu 1(1 + a_2)^2 \ddot{q}_{2r}(t) + \mu_1 \mu_2 \dot{q}_{2r}(t)}{(\mu 1(1 + a_2) - (1 + a_2)\dot{q}_{2r}(t) + \mu_2 q_{2r}(t))^2} \end{aligned} \tag{9}$$

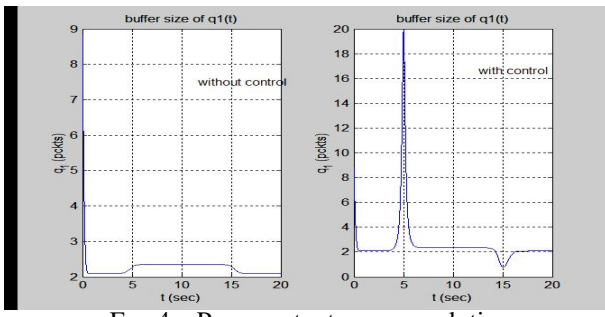

FIG. 4 – Proxy output queue evolution

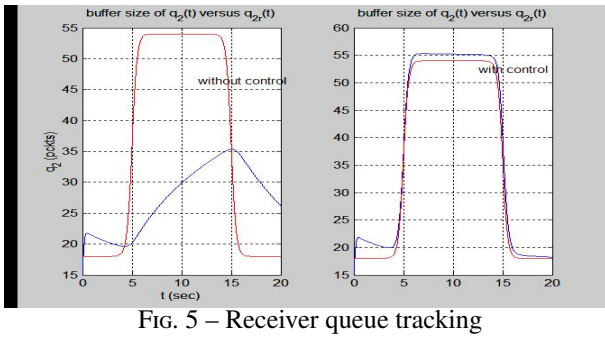

FIG. 5 – Receiver queue tracking

ensures the open loop tracking of $q_2(t)$. Some closed loop scheme must be added to ensure tracking in a practical case when the system is not stable. Note that the closed loop control is not addressed here due to space constraints, but related work may be found in [9] for interested readers.

5. Simulation and results

The simulation of the flatness control scheme requires the integration of the system dynamics (2) with the input packets rate $u(t)$ calculated by the control law (9) (for an $M/M/1$ queue). We choose the reference trajectory of the receiver queue q_{2r} as depicted in figure (3) obtained by the relation $q_{2r}(t) = a + b(\tanh(c(t-5)) + \tanh(-c(t-15)))$. a, b are parameters that determine the available buffer size. c is a parameter used to adjust the transition between these quantities. The simulation results are obtained using the so-called explicit forward Euler scheme with a time step $d = 0.01sec$, $\mu_1 = 140$ and $\mu_2 = 100$.

As illustrated in figures, the open loop control law (fi-

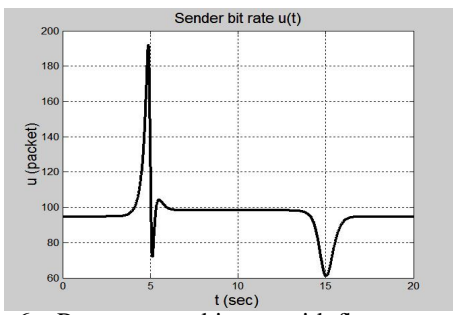

FIG. 6 – Proxy output bit rate with flatness control

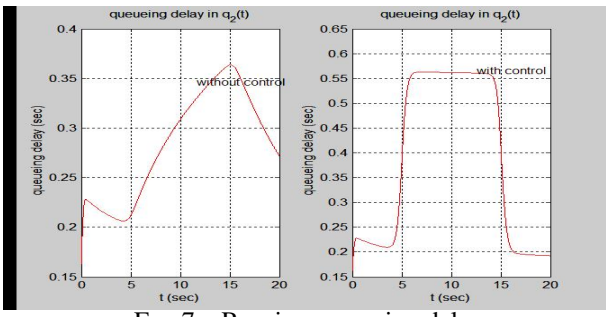

FIG. 7 – Receiver queueing delay

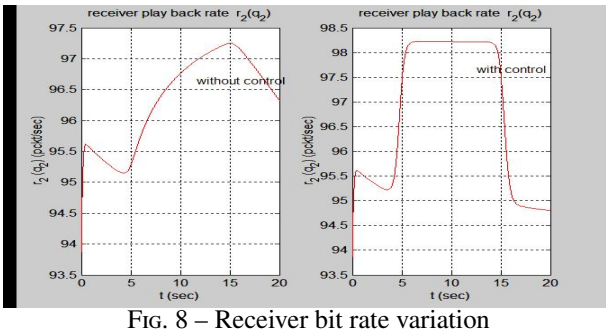

FIG. 8 – Receiver bit rate variation

gure 6) ensures the tracking of the reference flat output (figure 5 right). The model is assumed to be perfect and stable, as a result there is no need for closed loop control in order to stabilize the system around the reference trajectory. By controlling the proxy input bit rate, we have arrived to limiting its output queue length (figure 4 right). Thus,comparing to figures at the left, our approach is very efficient for managing network resources by maximizing the buffers utilization, smoothing the service delay and bit rate oscillations (see 5,7,8). Figures on the left are obtained by $u = \frac{\mu_1\mu_2 q_2}{\mu_1+(\mu_1-\mu_2 q_2)+\mu_2 q_2}$ (solution the equilibrium model of 2). As a result our *Active Queue Management* proposed by trajectory tracking method advocated here provides guarantees for critical traffic and may be used for dynamical provisioning of transmission service.

6. Concluding remarks

To ensure performance guarantees for streaming applications, we have developed reactive control policy which adapts the source rate to the network state variations. The proposed method called *trajectory tracking* deals with drastic short-term rate changes and limits the traffic in order to respect the time constraint. we show the contribution of the reactive control and the dynamic regulation using purely control theoretic approaches which stabilize the network and avoid undesirable oscillations for the transmission of such critical flows. In future work, we will extend our control method to consider several video servers and clients interconnected in computing manner.

References

[1] W. Ashmawi, R. Guérin, S. Wolf, and M.H. Pinson. On the impact of policing and rate guarantees in diffserv networks : a video streaming application perspective. In ACM, editor, *ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 83–95, San Diego, CA, USA, August 2001.

[2] O. Layaida and D. Hagimonte. Dynamic adaptation in distributed multimedia applications. Technical report, INRIA, August 2002.

[3] Adaptive video streaming in lossy networks : versions or layers ? Radulovic,i. and frossard,p. and verscheure,o. *IEEE International Conference on Multimedia and Expo ICME '04*, 3 :1915– 1918, June 2004.

[4] Y. Koucheryavy, D. Moltchanov, and J. Harju. A top-down approach to vod traffic transmission over diffserv domain using af phb class. In *ICC 2003, IEEE International Conference on Communications*, volume 26, pages 243–249, Seattle, May 2003.

[5] V. Guffens, G. Bastin, and H. Mounier. Fluid flow network modeling for hop-by-hop feedback control design and analysis. In *Internetworking 2003*, San Jose, CA, June 2003.

[6] P. Martin and N. Petit. Commande des systèmes non linéaires, le point de vue des systèmes plats. Cours, module thématique m2, Ecole des Mines, France, 1997-1999. Notes de cours.

[7] M. Fliss, P. Levine, P.and Martin, and P. Rouchon. A lie-backlund approach to equivalence and flateness of nonlinear systems. *IEEE, Transaction Automatic Control*, 44 :922–937, 1999.

[8] H. Mounier, V. Vèque, and L. Zitoune. *Algebraic Methods in Flatness, Signal Processing and State Estimation*, chapter Flatness and GPI for rate control in the DiffServ architecture, pages 119–130. ISBN 968-5785-333. Innovacion Editorial Lagares de Mexico, 2003.

[9] L. Zitoune, H. Mounier, and V. Vèque. Application du suivi de trajectoires dans l'ingénierie de trafic internet. Ief report series, IEF, Université Paris Sud, France, Mars 2007.