

# Satellite Terminal Quality of Service Management with AQM Control

Romain Delpoux, Pascal Berthou, Frédéric Gouaisbaut, Yann Labit

# ▶ To cite this version:

Romain Delpoux, Pascal Berthou, Frédéric Gouaisbaut, Yann Labit. Satellite Terminal Quality of Service Management with AQM Control. 15th Ka Band Conference, Sep 2009, Cagliari, Italy. pp.1. hal-00435757

# HAL Id: hal-00435757 https://hal.science/hal-00435757

Submitted on 24 Nov 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

### SATELLITE TERMINAL QUALITY OF SERVICE MANAGEMENT WITH AQM CONTROL

Romain Delpoux<sup>1,3</sup>, Pascal Berthou<sup>1,2</sup>, Frédéric Gouaisbaut<sup>1,2</sup>, Yann Labit<sup>1,2</sup> <sup>1</sup> CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France <sup>2</sup> Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France <sup>3</sup> Chalmers University, Göteborg, Sweden {rdelpoux, berthou, fgouaisb, ylabit}@laas.fr

#### Abstract

The standardization of a Return Channel via Satellite (DVB-RCS) and the satellite community efforts in term of interoperability over the last few years are expected to play, in a near future, a decisive role in Next Generation Networks (NGNs) through the integration of Satellite networks as an alternative to terrestrial networks like DSL (Digital Subscriber Line) in low terrestrial infrastructure areas. Furthermore, the advance of distributed multimedia applications like voice over IP and videoconference implies some new requirements to guarantee the Quality of Service (QoS). It concerns a limited transmission delay, a weak jitter, a minimal loss rate and a guaranteed bandwidth.

The recent NGN architectures prone the adaptation of the diffserv architecture to satellite systems. In this paper, we consider an architecture compliant with the SatLabs recommendation and propose a mechanism to improve the QoS management in Return link Satellite Terminals (ST). The goal of this work is to regulate a priori the Diffserv AF queue and avoid the overflood. Considering the AF queue sheared between UDP based multimedia applications and TCP based data transfer, the main idea consists in controlling TCP streams to guarantee transmission capacity of UDP packets. The most constraint application can then enjoy low buffers time delay and very few losses.

In order to solve this problem, we design a congestion control mechanism based on Active Queue Management (AQM) techniques by using control theory. To this end a fluid model of TCP connexion originally designed for wire-networks is proposed for satellite networks. Then, the design of a robust proportional integral (PI) and a robust dead time based controllers are investigated. To avoid AF queue over flooding, TCP packets are voluntary dropped in the ST according to regulation rules. TCP connection throughput is then controlled and limited to protect UDP streams against unnecessary drops. The different methods are then simulated on matlab and NS2-Simulator and compared to a classical DropTail mechanism.

#### 1. Introduction

Several commercial DVB-RCS based networks are already deployed and many efforts are done in order to enhance interoperability. Most recent commercial deployments provide either Internet access or mesh connectivity over a transparent geostationary satellite. Fixed bandwidth contracts are generally offered to consumers thanks to a simple resource management scheme. It simplifies admission control, reduces cost and gains experience while waiting for the standardization of finer resource management strategies and equipment. A lot of work on IP over satellite remains particularly in the Quality of Service (QoS) field and the next step is, obviously, to take benefits from DVB-RCS dynamic allocation schemes and IP QoS architectures to cope with the satellite delay and the scarce uplink resources.

This article deals a part of the ST, more precisely, it contributes to the QoS (Quality of Service) management of the return link, a central problem in satellite network (compared to wire network) due to the satellite delay and the scare uplink resources. Instead of over-sizing the connection, which could be very expensive, we aim at reaching an optimal exploitation of uplink resources [3].

In order to solve this problem, we design a congestion control mechanism (Active Queue Management (AQM)). This mechanism is equivalent to a controller designed by using the Automatic methods. Two AQM are described in this article, a robust proportional integral (PI) and a robust dead time based controllers. These mechanisms are applied to an existing model of the TCP protocol for wire network described in [4] that we adapt to the satellite network. The different method are simulated on NS2-Simulator and compared to the DropTail mechanism.

This paper is composed of three parts. The first part presents the context of the satellite network. The second part deals with the model of TCP and the design of both controllers. Finally, the third part presents the simulations and the results in order to show the behavior of this method. The last part concludes on the model evaluation and the future work.

#### 2. Context of the satellite network

#### 2.1. Satellites networks

#### 2.1.1. Satellite Access Scenario

The satellite access scenario is a typical Satellite Networks Architecture described in [3]. It consists of a geostationary satellite interconnected to terrestrial stations (Satellite Terminal) network. Satellite Terminals (STs) provide single PC or Local Area Network (LAN) with access to network, while Gateways (GWs) allow connection with Internet core network. The satellite network resources are managed by a Network Control Center (NCC). The uplink access from each ST is managed through DVB-RCS (Digital Video Broadcasting-Return Channel Satellite) interface, whereas transmissions form GWs are implemented through DVB-S (DVB-Satellite) interfaces.

#### 2.1.2. QoS Architecture

The QoS is managed in the ST at two levels : MAC (Medium Access Control) and IP (Internet Protocol).

Two classes of services are implemented at the MAC layer, DVB-RT (Real-Time) dedicated to applications with high temporal constraints (VoIP) and DVB-NRT (Non Real-Time) dedicated to more tolerant applications, or even not affected by delay. (Peer to Peer, FTP...). The QoS architecture proposed at the IP layer divides the traffic in three class of service: BE (Best-Effort) which guarantees nothing, AF (Assured Forwarding): ensures a relative QoS, EF (Expedite Forwarding): guarantees low end to end delays.

Here we consider the AF (Assured Forwarding) class of service of the IP layer, which is a class where a relative QoS should be ensuring. The transport protocols considered in this work are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). TCP is used for the NRT applications (Peer to Peer, FTP...), and UDP for time-sensitive applications (VoIP ...). The data applications are stored in the BE and AF queue while multimedia applications in the EF and AF queue. Consequently, AF queue receives data and multimedia applications which present a problem. In other words, TCP and UDP flows come in the same queue. When congestion occurs packets are lost and impact the TCP and UDP connections. TCP connections reduce their throughput and UDP (multimedia) connections experience a lower quality.

The goal of this work is to regulate a priori the AF queue and avoid queue over-flooding. Therefore, it is necessary to keep place in the buffer for UDP packets in priority, as TCP packets can be retransmit contrary to UDP and reduce the buffer size, while keeping reasonable transmission capacity, i.e. having as few losses as possible (and then protect UDP packets).

So we have to model the AF queue and regulate it in order to respect the conditions above-cited.

#### 2.2. TCP

It is a "end-to-end" communication protocol, which means that a direct link between the source and the destination is established. The main characteristic of this protocol is to certify the data reception by the receiver using mechanism based on acknowledgement. If a packet is lost, the sender should send it again. Thus TCP assures the transmission of the entire informations.

TCP is a general purpose protocol, and does not make assumption on the network used. To find the maximum transmission throughput, TCP probes the network until reaching the limit. This is the role of slow start and congestion avoidance mechanisms.

#### 2.2.1. Congestion Avoidance Algorithms

Congestion avoidance algorithm have been developed to regulate the flow rate the closer as possible to the "transmission limit". In order to transmit the maximum of informations and avoid network congestion. The basic hypothesis of this algorithm is to consider that a packet lost is synonym of congestion. The principle of the algorithm is to control the rate of each sources function of traffic state. The principle is simple, each source increases progressively their output flow. This increases takes place until a packet loss occurs. This means that congestion is detected somewhere in the network. Thus the flow is decreased enough in order to go out of the congestion state. The following sub-section give an outline of the Additive Incease Multiplicative Decrease (AIMD) algorithm implemented in TCP.

#### 2.2.2. Algorithm

- The source sends *W* packets.
- The receiver acquits the received segment and the source acts in consequence:
  - If the flux is transmitted with success, the source increases its size:  $W \leftarrow W + 1$
  - If there is a loss, the source should retransmit its data and reduce its congestion window. There is principally two kind of loss identification, indications by Timeout (TO) and indications by duplicate acknowledgement (3DupAck)
    - \* If the source did not receive the acknowledgement, TO:  $W \leftarrow 1$ .

\* If the source receive three duplicate acknowledgement, 3DubAck:  $W \leftarrow W/2$ .

The time of one exchange correspond to a way return, i.e. one RTT (Round Trip Time).

#### 3. TCP modeling and controllers design



Figure 1: The considered system

Following the algorithm developed in the last paragraph we clearly recognize that the overall system is an interconnected feedback system as described by figure 1. Thus feedback control principles appear to be an appropriate tool for the analysis and the design of AQM strategies. The main principle is to drop intentionally TCP packets before the router queue becomes full so that the source can prevent the congestion, by reacting to the losses with the congestion avoidance mechanism. The goal is to optimize the data transmission maintaining a high stream at the buffer level. AQM detects network congestion, packet losses of incipient congestion, and inform traffic sources. Sources reacts and decrease the congestion window to avoid buffer saturation. This way to prevent router congestion is an active research subject, see for example [1] and references therein. We propose in this paper an AQM to enhance IP QOS on the return channel.

#### 3.1. A fluid-flow Model of TCP congestion control mechanism

In order to use control theory, we propose to introduce a mathematic model which has been developed in [7] and [6]. This model is based on two assumptions: the traffic is considered fluid-flow and the losses are described by a Poisson Process. Moreover we consider N homogeneous sources. That means for example, when a loss occurs, all of the TCP connexions react to the loss. Off course, considering a single loss, the hypothesis is improper, but considering multiple losses and because TCP as been designed to be fair, this hypothesis becomes accptable. This is designed to guarantee the equity between TCP connections and modify the windows size of each sources simultaneously.

The first assumption implies that the congestion window increases in a continue way instead of step increase. It increases by one every RTT and hence the continuous increase is represented as dt/RTT. The second assumption models the packets loss occurrences. We assume idealized behavior, i.e. we model the losses as Poisson streams represented as  $-\frac{W(t)}{2}dN(t)$ . Then the evolution of the congestion window size W can be described as follow:

$$dW(t) = \frac{dt}{RTT} - \frac{W(t)}{2}dN(t)$$
(1)

by noting that dN(t) is defined as:

$$dN = \begin{cases} 1, \text{ if losses arrivals} \\ 0, \text{ otherwise} \end{cases}$$
(2)

This equation reflects the "Additive Increase Multiplicative Decrease" aspect of TCP. The first term corresponds to the additive increase part, which states that the windows size will increase by one every RTT. The second term corresponds to the multiplicative decrease part, which halves the window size for each arrival of a loss. Remark that we use a simplified model, which ignores the TCP slow start mechanism, that start at the beginning of a connection, and time-outs. Effectively, some measures realized on Ourses-Project [8] using a DVB-S2/RCS system using a Ka-Band link, show that timeouts barely occur, and TCP work most of the time as congestion avoidance instead of a slow start at the beginning of the connection.

#### 3.2. The system

Using stochastic differential analysis of the equation (1), and considering the simplifications above cited, [7] have developed a dynamic model of the TCP behavior. In this model, we consider a system in which there is a single congested router with a transmission capacity of C. Associated with this router is an AQM that is characterized by a packet discard function  $p(\cdot)$  that

takes as its argument an estimate of the average queue length at the router, and the average congestion window size. The proposed model from [4] is then of the form:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)}{2} \frac{W(t-R(t))}{R(t-R(t))} p(t-R(t))$$

$$\dot{q}(t) = -C(t) + \frac{N(t)}{R(t)} W(t)$$
(3)

where  $\dot{x}$  denotes the time-derivative and

 $W \doteq$  average TCP window size (packets),

 $q \doteq$  average queue length (packets) of the AF queue in the ST,

 $R \doteq$ round-trip-time =  $\frac{q}{C} + T_p$ (secs),

 $C \doteq link$  capacity (packets/sec),

 $T_p \doteq$  propagation delay (secs),

 $\dot{N}$   $\doteq$  load factor (number of TCP sessions),

 $p \doteq$  probability of packet mark, which takes values only in [0,1].

The dynamic TCP behavior is modeled by a non-linear time delay systems which can be complicated to analyse from a control theory point of view. That is the reason why we are only interested in the design of an AQM around an equilibrium point  $(W_0, q_0, p_0)$ .

To linearize model (3) we first assume that the number of TCP sessions and the link capacity are constant i.e.,  $N(t) \equiv N$  and  $C(t) \equiv C$ . Taking (W,q) as the state and p as an input, the operating point  $(W_0, q_0, p_0)$  is then defined by  $\dot{W} = 0$  and  $\dot{q} = 0$  so that

$$\begin{cases} \dot{W} = 0 \Rightarrow W_0^2 p_0 = 2\\ \dot{q} = 0 \Rightarrow W_0 = \frac{R_0 C}{N}, \ R_0 = \frac{q_0}{C} + T_p \end{cases}$$
(4)

Moreover, we ignore the dependance of the time-delay argument t - R on the queue-length q, and assume it fixed to  $t - R_0$ .

We obtain finally the linearized model (5) around equilibrium point defined by (4):

$$\begin{cases} \delta \dot{W}(t) = -\frac{N}{R_0^2 C} \Big( \delta W(t) + \delta W(t-h) \Big) \\ -\frac{1}{R_0^2 C} \Big( \delta q(t) - \delta q(t-h) \Big) - \frac{R_0 C^2}{2N^2} \delta p(t-h) \delta \dot{q}(t) = \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) \end{cases}$$
(5)

where  $\delta W \doteq W - W_0$ ,  $\delta q \doteq q - q_0$  represent the state variables and  $\delta p \doteq p - p_0$  the input. This can be rewriten as (6):

$$\begin{bmatrix} \delta \dot{W}(t) \\ \delta \dot{q}(t) \end{bmatrix} = \begin{bmatrix} -\frac{N}{R_0^2 C} & -\frac{1}{CR_0^2} \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} + \begin{bmatrix} -\frac{N}{R_0^2 C} & \frac{1}{CR_0^2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta W(t-h) \\ \delta q(t-h) \end{bmatrix} + \begin{bmatrix} -\frac{C^2 R_0}{2N^2} \\ 0 \end{bmatrix} \delta p(t-h)$$
(6)

In order to use the robust framework from the control theory, let modify the system (6) as an interconnected system as Figure (2) by isolating  $\Delta(s) = \frac{2N^2s}{R_0^2C^3}(1 - e^{-sR_0})$ . The nominal system is then described by:

$$\begin{bmatrix} \delta \dot{W}(t) \\ \delta \dot{q}(t) \end{bmatrix} = \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} + \begin{bmatrix} -\frac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix} \delta p(t-h) + \xi(t)$$
(7)

Where  $\xi(t) = Hz(t)$  represent the uncertainties input.



Figure 2: Block diagram of a linearized AQM control system

Notice that in Figure (2), C(s) represents the control law implemented at the router level. We aim at designing C(s) to ensure the closed-loop stability and the performances objectives. To this end, we propose to use the result of [4] that gives the condition and the proof for stabilization. C(s) should stabilizing the delayed nominal plant and gain-stabilizing the uncertainties  $\Delta(s)$ . The linearized AQM control system is stable if C(s) stabilized the delayed nominal plant, and  $\Delta(s)$  is gain stabilized, i.e., the product of  $\Delta(s)$  and the closed loop transfer function of the nominal plant with the controller is less than one (using the generalized Nyquist criterion). We get a suitable model where we can apply Automatic feedback control in order to control the TCP dynamic. The development of command law will be detailed in the next section . We have to use methods which take in consideration the control delay by respecting the robustness conditions on the uncertainties.

#### 3.3. The Controllers

We present two different AQM strategies using automatic control law to get an anticipatory congestion detection and control capability but also achieve satisfactory control performance in terms of the queue length dynamics (or equivalently the queueing delay). The first method use the well know Proportional Integral (PI) feedback control adapted to the TCP network in [4]. The second method is a predictive controller introduced in [5] for a system with control delay that we have adapted to TCP network. For the study of both controllers, we assume that the RTT is constant.

#### 3.3.1. PI controller

In this section, we propose to design a classical PI controller adapted to system of the form (7), defined in [4]. C(s) is the transfer function of the controller. L(s) is the open-loop transfer function of the model.

$$C(s) = K_{PI} \frac{\frac{s}{z} + 1}{s} \quad L(s) = \frac{\frac{K_{PI}C^2}{2N} \left(\frac{s}{z} + 1\right) e^{-sR_0}}{s \left(s + \frac{2N}{R_0^2C}\right) \left(s + \frac{1}{R_0}\right)}$$
(8)

We define z such that the dominant pole is cancelled  $(z = 2N/R_0^2C)$ . The crossover frequency is  $\omega_g = \frac{\beta}{R_0}$  where  $\beta$  is chosen to set the phase margin. Then  $K_{PI} = \omega_g z \left| \frac{j\omega_g + \frac{1}{R_0}}{\frac{C^2}{2N}} \right|$  in order to meet the crossover frequency condition  $|L(j\omega_g)| = 1$ . We then calculate the desired phase loop by choosing  $\beta$  which lead to a positive phase margin.

#### 3.3.2. Predictive controller

We consider the state space model (7) where  $A = \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}$  and  $B = \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}$ . In order to implement a static feedback, we need to know  $\begin{bmatrix} \delta W(t+R_0) \\ \delta q(t+R_0) \end{bmatrix}$ :

$$\begin{bmatrix} \delta W(t+R_0)\\ \delta q(t+R_0) \end{bmatrix} = e^{A.R_0} \begin{bmatrix} \delta W(t)\\ \delta q(t) \end{bmatrix} + \int_0^{R_0} e^{A\theta} Bp(t-\theta) \, d\theta.$$
(9)

A static state feedback is then proposed as:

$$p(t) = F \begin{bmatrix} \delta W(t+h) \\ \delta q(t+h) \end{bmatrix} = F \left( e^{Ah} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} + \int_0^{R_0} e^{A\theta} Bp(t-\theta) \, d\theta \right).$$
(10)

where F is an  $m \times n$  matrix to be found

Notice that we can rewrite the controller as follow, enhancing the infinite dimensional feature of the proposed control law.

$$p(t) = F_{state} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} + \int_0^{R_0} F_p(\theta) p(t-\theta) \, d\theta.$$
(11)

Where  $F_{state} = Fe^{AR_0}$  and  $F_p(\theta) = Fe^{A\theta}B$ . This lead to the closed-loop system:  $\begin{bmatrix} \delta \dot{W}(t) \\ \delta \dot{q}(t) \end{bmatrix} = (A+BF)\begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix}$  and the characteristic equation of the closed-loop system is then: det(sI - (A+BF))

The controller (11) is not usable for the modeling using Matlab/Simulink or NS2-simulator directly. We develop the equation (11) we arrived to the following form which is implementable:

$$p(t) = F_{state} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} + C_1 e^{\lambda_1 t} \begin{bmatrix} \int_0^t e^{-\lambda_1 \theta} p(\theta) \, d\theta - \int_0^{t-R_0} e^{-\lambda_1 \theta} p(\theta) \, d\theta \end{bmatrix} + C_2 e^{\lambda_2 t} \begin{bmatrix} \int_0^t e^{-\lambda_2 \theta} p(\theta) \, d\theta - \int_0^{t-R_0} e^{-\lambda_2 \theta} p(\theta) \, d\theta \end{bmatrix}$$
(12)

This form of the controller is implementable with simulations software.

#### 4. Results and applications

#### 4.1. Network typology

We consider the network topology consisting of 6 TCP sources and 1 UDP source, with the same propagation delay connected to a destination node through a router (see figure 3). All this sources together arrives to the satellite connection, which has a bigger propagation delay and a smaller link capacity, where the phenomenon of congestion collapse appears. It is then necessary at this node to control the stream with the help of an AQM.



Figure 3: The network topology

We have adapted the parameters to the conditions of the satellite network, i.e. the link capacity C = 128 packets/sec<sup>1</sup>, the propagation delay  $T_p = 0.7$  secs and the number of sources N = 6 sources. The operating point is chosen such that the average queue length  $q_0 = 35$  packets. Using the equations (4) we obtain a RTT  $R_0 = 0.9734$  secs, an average TCP window size  $W_0 = 20.77$  packets and a probability of packet mark  $p_0 = 0.0046$ .

Both PI controller and Predictive controller have been previously simulated on Matlab/Simulink and the stability checked. However, in this article we will present the simulations with NS2-Simulator only. We will make a comparison between the DropTail, which can be considered as the simplest AQM, the PI and the predictive controllers. For each simulation we have depicted the congestion window, the queue size, the propagation delays, and the packets losses.

The DropTail queue management is already available in the NS library. We only need to specify the queue management type when we create the link between 2 sources. When the queue is filled up to its maximum capacity, the newly arriving packets are dropped until the queue has enough room to accept incoming traffic. The probability of packet mark for the PI and the predictive controllers is computed at each packet arrival, which defines the sampling frequency by calculating the time between each arrivals. The probability is a value between 0 and 1. We draw a random variable v. If p is smaller than v we keep the packet, if p is bigger, we drop the packet. With DropTail, when the queue is filled to its maximum capacity, the newly arriving packets are dropped until the queue has enough space to accept incoming traffic.

#### 4.2. First Experimentations: TCP behavior

AQM		DT	PI	PC	
Mean(pkts)			136.2	36.8	30.6
Stand.dev.(pkts)			45.8	28.8	27.7
$Average \ queueing \ delay(secs)$			1.7	0.99	0.94
	AQM	DT	PI	PC	, Y
	Transmitted(pkts)	25702	25590	2551	.3
	Dropped(pkts)	225	236	286	;
	Percentage(%)	0.88	0.92	1.1	

Table 1: Statistics on the queue length for the three AQM (left) and on TCP packet losses (right

In this part, we make a simulation with TCP only. With this simulation, we observe the TCP performances (see table 1). As expected, the DropTail AQM as a larger average queue size.

<sup>&</sup>lt;sup>1</sup>corresponds to a 0.512 Mb/s link with an average packet size 500 bytes

The window size of each sources increase until a packet loss occurs. As we know that the RTT is function of the queue size  $(R(t) = T_p + \frac{q(t)}{C})$ , this method leads to bigger propagation delays. Thanks to the Integral part of the PI controller, which cancels the static state error, the average queue size is closed to the fixed reference (35 packets). So we have reasonable delays because the queue size does not grow. About the predictive controller, the average queue size in less than the reference, but the standard deviation is smaller than the PI's standard deviation. From the losses point of view (see figure 4) we observe that the number of losses is quiet similar, around 1%. We can also point that with DropTail, losses occur at the moment where the queue is full, then when congestion occurs, all the packets are dropped. This can pose a problem to solve the congestion problem, because the router is already congested and we will have to resend all the dropped packets. Using the two others method, the buffer is never congested, and packets are lost continuously one by one. It will be easier to resend the lost packet.



Figure 4: Queue size and losses on NS with the three AQMs for TCP connection only

About the figures (4), we can also point out that thanks to the regulation the queue size is more stable for the two automatic based AQM than for the DropTail. This behavior is thus reflected on the propagation delay. With DropTail delays exceed two seconds, while they are around one second for the two others. With this simulation, we can conclude that the AQM using automatic control are much more efficient in term of delay.

#### 4.3. Second Experimentations: TCP behavior with UDP perturbations

In this simulations, UDP flux arrive with a flow rate of 128kbits. A first UDP connection occurs between the  $40^{th}$  and the  $160^{th}$  seconds. As second one between the  $80^{th}$  and  $120^{th}$  seconds.



Figure 5: Queue size and losses on NS with the three AQMs for TCP and UDP connections

AQM	DT	PI	PC
Transmitted(pkts)	5120	5120	5120
Dropped(pkts)	29	23	0
Percentage(%)	0.57	0.45	0

Table 2: Statistics on UDP packet losses for the three AQM

The Figure (5) and the Table (2) points that using DropTail, if the perturbation occurs when congestion occurs, the buffer is full, so all the UDP packets will be lost. Using this controller,

the connection will be cut during the congestion phenomena. With the PI controller some UDP packets are dropped during the connection, which will lead to a lower connection quality, but still acceptable. Otherwise using the predictive controller, none of the UDP packets are lost, then the UDP connection will be completely transmit.

#### 5. Conclusion and future work

In this paper, we have exposed two different AQMs based on automatic control that we compare to the DropTail AQM. This method is very interesting in term of delay regulation. Furthermore, it leads to a lower delay and which oscillates less than DropTail AQM. Concerning the PI controller, we obtain a zero study-state regulation error, and the method is very easy to implement in *ns*. Regarding the predictive controller, the implementation is much more complex. The behavior is similar to the PI controller behavior, however none of the UDP packets are lost which is one of most important goal of this work, despite that we have a study state error. Conversely, the DropTail AQM poses a huge problem if the UDP packets arrive when the buffer is full.

In order to improve the predictive controller, it could be useful to implement an integral part, which will cancel the study-state error. One of the next objective is to implement the two controllers on a platform (Platine [2]) to make more experimentations.

Another method that could be used for this problem is to have two queue (one for UDP and one for TCP) with different priorities. However in this scheme, we can ask the question of the necessity of AF queue, it is like if we had in the AF queue EF and BE queues. Moreover, some problems of TCP instability could occur, because TCP packets will be blocked when UDP packets arrived, so the congestion phenomena will be accentuate (c.f. 4.3).

## References

- [1] Y. Ariba, Y. Labit, and F. Gouaisbaut. Congestion control of a single router with an active queue management. *International Journal on Advances in Internet Technology*, 2(1), 2009.
- [2] C. Baudoin, Mathieu Dervin, Pascal Berthou, Thierry Gayraud, Frédéric Nivor, Baptiste Jacquemin, Didier Barvaux, and J. Nicol. PLATINE: DVB-S2/RCS enhanced testbed for next generation satellite networks. In *International Workshop on IP Networking over Nextgeneration Satellite Systems (INNSS07)*, volume ISBN-13: 978-0387754277, page 11p., 07 2007.
- [3] T. Gayraud and P. Berthou. A qos architecture for dvb-rcs next generation satellite networks. *EURASIP Journal on Wireless Communications and Networking*, ID 58484 doi:10.1155/2007/58484:200–300, 2007.
- [4] C. V. Hollot, V. Misra, D. Towsley, and W. Gong. Analysis and design of controllers for aqm routers supporting tcp flows. *IEEE Trans. on Automat. Control*, 47:945–959, jun 2002.
- [5] A. Z. Manitius and A. W. Olbrot. Finite spectrum assignment problem for systems with delays. *IEEE Trans. on Automat. Control*, AC-24:541–553, aug 1979.
- [6] V. Misra, W. Gong, and D. Towsley. Stochastic differential equation modeling and analysis of tcp windowsize behavior. Technical report, University of Massachusetts, October 1999.
- [7] V. Misra, W. Gong, and D Towsley. Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. In *SIGCOMM*, pages 151–160, August 2000.
- [8] I. Tou, M. Gineste, T. Gayraud, and P. Berthou. Quality of service evaluation in satellite systems. *International Workshop on Satellite and Space Communications 2008 (IWSSC 2008)*, pages 133–137, oct 2008.