An Optimal AQM Controller for the DiffServ Architecture

Leonardo R. Augusto  
Institute of Computing  
State University of Campinas  
Brazil

Nelson L. S. da Fonseca  
Institute of Computing  
State University of Campinas  
Brazil

Michele M. A. E. Lima  
Federal University of de Pernambuco  
Caruaru  
Brazil

Abstract—The Assured Forwarding Per Hop Behavior of the Differentiated Services Architecture (DiffServ) was proposed to provide throughput differentiation in the Internet. In this paper, we use a non-rational approach to develop an optimal AQM controller for this PHB. The design considers both adaptive and non-adaptive traffic. Simulations show that the proposed controller produces goodput per connection higher than those produced by existing proposals.

I. INTRODUCTION

Multimedia applications have specific Quality of Service requirements which cannot be guaranteed under the “best effort” service model of the Internet. To cope with such requirements, the DiffServ architecture was standardized by the Internet Engineering task force (IETF) to support these applications. In the DiffServ architecture, network traffic from different flows is clubbed in different Per Hop Behavior (PHB) according to their QoS requirements. One of the PHBs standardized is the Assured Forwarding (AF) [2] which promotes throughput differentiation.

This paper focuses on AF DiffServ networks using token bucket [3] at the edge for coloring packets. Packets are marked green when rate is within the negotiated value; otherwise they are marked red. In case of congestion, the chances of dropping (or ECN bit setting [4]) red packets is higher as compared to green ones. However, in networks with TCP flows, bandwidth provisioning is also complex due to the TCP congestion control mechanism which reacts slowly to bandwidth availability changes. Moreover, end-user throughput depends on other flows and on the propagation delay. Therefore, a bandwidth differentiation mechanism should coordinate with TCP congestion control mechanism to provide guarantees to different flows.

In [8], an Active Rate Management (ARM) mechanism was introduced for adaptively setting token bucket parameters, in order to achieve minimum guarantee throughputs in various network conditions. In [8], it is demonstrated that when combined with two-level PI-AQM [9], the ARM mechanism is able to maintain minimum throughput at or above target rates and to respond to fluctuations in traffic characteristics. But the response time taken, in all the cases analyzed in [8] is quite long. Agrawal et. [1] al. proposed a Linear Quadratic Regulator based on Proportional-Integral-Derivative (PID) controller for AQM to improve the response time. The 'D' component improves the stability and the responsiveness of the system [20].

This paper introduces a novel AQM controller based on Optimal Control Theory instead of using Classical Control Theory. Controllers based on Classical Theory, such as PI and PID, are frequently sensible to model uncertainties which does not happen as much with controllers based on Optimal Control Theory. Moreover, using optimal control strategy the desired system behavior is formulated in terms of a cost function that is minimized. The use of an objective function allows the designer to specify exactly the control objectives, that once determined, are optimally met.

The novelty of the proposed approach lies in the use of non-rational controllers to prove stability with respect to the delay component of the system [2]. It is well-known that delay independent control presents performance limitations in the presence of long delays [3], [4] [5]. Stability and performance objectives are completely expressed as Linear Matrix Inequalities (LMIs), requiring the solution of a single convex problem with low computational cost for the determination of the controller parameters values.

Although the same system of equations used to derive the PI-AQM controller was used to derive the optimal controller, the plant utilized in this work represents the system in greater detail. The advantage of representing the complete dynamics of the system in the plant is that the stability of the plant implies on the stability of the system.

II. THE ARM/AQM DIFFSERV ARCHITECTURE

To support the AF PHB, ARM and AQM mechanisms need to cooperate so that Minimum Guaranteed Rates (MGR) are provided. The ARM mechanism is located at the edge of the network and it is responsible for setting the token bucket parameters to assure the MGRs whereas AQM mechanism are placed at the network core routers and it drops packet in case of congestion or incipient congestion.

A network with m routers at the network edge, each router acting as a traffic aggregator of $N_j$ TCP flows. Leaky Buckets mark the aggregate of these TCP flows with leaky rate $A_j(t)$ and have a bucket capable of holding $b_j \gg 1, j = 1, \ldots, m$ tokens. Leaky rates are adjusted by an ARM controllers so that MGRs are guaranteed. The output flow of the edge routers feed a single router at the network core which has an output...
capacity $C$ and buffer size $q$. Let $r_j(t)$, for $0 < j < m$, be the transmission rate of $j^{th}$ aggregate and $\tilde{r}_j$ the guaranteed rate for that aggregate. The network is said to be overprovisioned in case $(\sum_{j=1}^{m} \tilde{r}_j < C)$ and exactly provisioned when $(\sum_{j=1}^{m} \tilde{r}_j = C)$. Moreover, if $(\sum_{j=1}^{m} \tilde{r}_j > C)$ the network is considered underprovisioned.

Let $f_{ij}^j(t)$ be the fraction of $r_j(t)$ marked as green and $f_{ij}^j(t) = 1 - f_{ij}^j(t)$ the fraction of $r_j(t)$ marked as red. The green fraction is given by $f_{ij}^j(t) = \min\left(1, \frac{\tilde{r}_j}{r_j(t)}\right)$. An ARM mechanism which mark packets as green according to the rule $r_j(t) < \tilde{r}_j \Rightarrow f_{ij}^j(t) = 1$ and $r_j(t) > \tilde{r}_j \Rightarrow f_{ij}^j(t) = 0$ is said to fully color at equilibrium.

Let $p_g$ denote the discarding (marking) probability of green packets and $p_r$ denote the same probability for red packets. Note that $0 < p_{g}(t) f_{ij}^j(t) < p_{r}(t)(1 - f_{ij}^j(t)) < 1$ which means that the discarding probability by the AQM mechanism of green packet is always lower than those of red packets. An AQM mechanism which discard packets according to the following rules $\tilde{p}_o < 0 \Rightarrow \tilde{p}_i = 0$ e $\tilde{p}_o > 0 \Rightarrow \tilde{p}_o = 1$ is called non-overlapping.

In [6], fundamental relationships were proved for the assurance of the MGRs. In networks where the ARM mechanism fully color the packet flow and the AQM mechanism is non-overlapping the MGRs are guaranteed. Moreover, in overprovisioned networks, the transmission rates can exceed the guaranteed value.

### III. An ARM/AQM Diffserv Architecture

In this section, the system of equation describing the dynamics of the ARM/AQM Diffserv system is introduced as well as the target operational points derived.

Let $W_j(t)$ denote the window size of TCP connections composing the $j^{th}$ flow and $R_j(t) = T_j + \frac{q(t)}{C}$ the round trip time faced by packet of this flow where $T_j$ represents the propagation delay and $\frac{q(t)}{C}$ the delay due to queueing at the core router. The sending rate $r_j$ can be expressed as $r_j(t) = \frac{N_j W_j(t)}{R_j(t)^2}$ and the window size variation given by:

$$\dot{W}_j(t) = 1 \frac{R_j(t)}{r_j(t)} \frac{R_j(t)}{2R_j(t)} p(t - R_j(t)) \tag{1}$$

In this equation, the first term is due to the additive increase whereas the second term is associated to the multiplicative decrease of the TCP congestion control mechanism. The discarding probability $p(t - R_j(t))$ is given by $p = \frac{\tilde{r}_j}{r_j(t)} p_r(t - R_j(t)) + \left(1 - \frac{\tilde{r}_j}{r_j(t)}\right) p_g(t - R_j(t))$

The variation of the core router queue length is computed as the difference between the link capacity and the aggregate arrival rate at the router, expressed as:

$$\dot{q}(t) = -C + m \sum_{j=1}^{m} \frac{N_j W_j(t)}{R_j(t)} + \omega_q(t) \tag{2}$$

Equation (2) differs from the one introduced in [7] by the introduction of non-adaptive flow such as UDP, yielding a more realistic system model.

The derivation of AQM controllers needs to distinguish different network scenarios. In over and in exact provisioned networks, there is available capacity to assure the MGRs. Thus, the aim is to avoid packet discard of green packets, keeping the queue length close to a target equilibrium value which yield low delay value. In under provisioned networks, it can be necessary to discard green packets. The compromise is to enlarge the target queue length to accommodate green packets for ensuring the MGRs. In line with that, two operational points were derived: one for scenarios where the network is either over provisioned or exactly provisioned and the other when the network is under provisioned. Actually, our solution adopts two different controllers, one for each of the mentioned cases.

To adhere to the non-overlapping behavior, the discard probability of green packets should be zero in over provisioned as well as in exactly provisioned scenarios whereas the discard probability of red packets should vary from zero to one. Thus, only variations of the values of the variables $W_{0j}$, $q_0 e p_{r0}$ impact the equilibrium point.

Moreover, in under provisioned scenarios the discard probability of red packets should be equal to 1 whereas the discard probability of green packets should vary between 0 and 1. Thus, only the value of the variables $W_{0j}$, $q_0 e p_{r0}$ impact the equilibrium point. The derivation of these points are described next.

### Equilibrium point for over and exactly provisioned networks

The derivation of the equilibrium point for exactly provisioned scenario is considered. Linearizing the system of equations (1) and (2) around the equilibrium point $(W_{0j}, q, p_{r0})$, it follows that:

$$\dot{x}_{1j}(t) = -\frac{N_j m_j}{C R_j} \left[ x_{1j}(t) - x_{1j}(t - R_0) \right]$$

$$\dot{x}_{2j}(t) = m_j \frac{N_j}{R_j} x_{1j}(t) - \frac{1}{R_j} x_{2j}(t) \tag{5}$$

where $R_j$ guaranteed rates is provided o each aggregate $j$, $r_j = C$, $m_j \frac{N_j W_j(t)}{R_j(t)} + \omega_q(t)$, $r_j = 12916$ pkts/sec.

To derive the equilibrium point, some network parameter values were considered. The capacity of the link was set to $C = 155Mbps$ (38750 packets of 500 bytes per second). The number of TCP flows per aggregate at the edge is $N = 413$ and the round trip time $R = 256ms$. The target queue length was set to one quarter of the bandwidth delay product yielding $q = 2480$ packets, $(q = C.R = 9920)$. The propagation delay used was $T = 0.2$ sec [8]. Three aggregate at the edge were
considered, \((m = 3)\), and the TCP window size \(W = 8\) was adopted to avoid unnecessary timeouts. Choosing \(r = 12916\) packets/sec and \(A = 2583\) packets/sec, Equation (1) can be solved and \(p_r = 0.0390625\) obtained.

**Equilibrium point for under provisioned networks**

Linearizing the system of equations (1) and (2) around the equilibrium point \((W_0, q, p_g)\) for the under provisioned case, it follows:

\[
\dot{x}_{1j}(t) = \frac{N_j}{C R_j^2} [x_{1j}(t) - x_{1j}(t - R_0)] - \frac{1}{R_j^2} C x_{2j}(t)
\]

\[
+ \frac{1}{R_j} C x_2(t - R_0) - \frac{C A_0 R_j}{2N_j^2 m} u(t - R_0)
\]  

(6)

\[
\dot{x}_2(t) = \frac{m N_j}{R_j} x_{1j}(t) - \frac{1}{R_j} x_2(t)
\]  

(7)

The values of \(C, m, r, W \in T\) are the same adopted for the previous scenario. Moreover, \(R = 320\) ms; \(p_r = 1, e p_g = 0\), since all red packets should be discarded in this case. The round trip time was enlarged given the under provisioning of resources.

From \(p_g, p_r \in r\) values, it follows that \(A = 2513\) packets per second, the number of TCP flows per aggregate is \(N = 516\) and the resulting target queue length \(q = 6200\) packets.

**IV. DESIGN OF OPTIMAL AQM CONTROLLER TO THE ARM/AQM DIFFSERV ARCHITECTURE**

The AQM control system proposed in this paper, dsH2-AQM, is actually composed by two controllers, one for each equilibrium point derived in the previous section. The controller \(H2_o\) and \(H2_r\) compose a non-overlapping AQM system, i.e., at any time only one of them is operational.

Optimal control Theory was used to design dsH2-AQM. The desired system behavior is formulated in terms of a cost criterion that is minimized. The use of a cost or objective function allows and forces the designer to specify exactly the control objectives of the system, that once determined are optimally met. Furthermore if the system model and the knowledge about the initial state are of sufficient quality, the performance of the controller optimal control is practically the best that can be achieved. In this section, the desired system behavior is formulated as a cost minimization problem with an objective function exactly specifying the control objectives.

Most of AQM policies based on Control Theory [5], [9] consider only the current dynamic information and consequently do not compensate explicitly for long feedback delays. It is well-known that delay independent control has a limit on performance in the presence of large delays, compared with delay-dependent control [3], [4]. In the network context, this means that by using a delay dependent approach the networks resources can be efficiently utilized [5]. In this paper, the congestion control system is presented in space frame form as a linear delay system, with a non-rational approach used to derive the optimal controller. The use of a non-rational approach not only assures network efficiency in the presence of long delays but also overcomes the difficulty involved in incorporating the matrix multiplier into the problem to prove stability with respect to the delayed component of the system as this is the major challenge in designing a rational controller for linear delay systems [2].

The synthesis of the controller presented here is based on results introduced in [2], where the design and synthesis of non-rational controllers for linear delay systems are expressed and solved as LMIs. The form of the controller was carefully chosen to reproduce the structure of the plant of a linear delay system. System stability is then analyzed through the connection of the system plant and the controller. Such a representation makes possible the use of delay independent stability analysis, which creates synthesis conditions that can be expressed and solved as LMIs.

Although the design of dsH2-AQM considers the same system of equation used to derive PI-AQM, the plant used represents the system in greater detail, and also takes into account the presence of non-adaptive traffic such as UDP. The advantage of representing the complete dynamics of the system in the plant is the fact that the stability of the plant implies on stability of the system. The systems of equation can be expressed in state space as:

\[
\dot{x}(t) = A_0 x(t) + A_1 x(t - R_0) + B_u w(t) + B_u u(t); (8)
\]

\[
z(t) = C z(t) + D_{zu} u(t);
\]

\[
y(t) = C_y x(t - R_0) + D_{yu} w(t);
\]

Matrices \((A_o, B_u, C_o, D_u)\) are defined considering the state space vector \(x(t) \in \mathbb{R}^n\), the input \(u(t) \in \mathbb{R}^m\), the input noise \(w(t) \in \mathbb{R}^r\), the target output \(z(t) \in \mathbb{R}^p\) and the measured output \(y(t) \in \mathbb{R}^p\).

Now, consider that such system is connected to a controller defined as:

\[
\dot{x}(t) = \hat{A}_0 \hat{x}(t) + \hat{A}_1 \hat{x}(t - R_0) + \hat{B} y(t);
\]

\[
u(t) = \hat{C}_u \hat{x}(t) + \hat{C}_1 \hat{x}(t - R_0) + \hat{D}_y y(t);
\]  

(9)

Considering the linearized equations (3) and (5) for the controller \(H2_r\) and (6) (7) for the controller \(H2_o\), the state vector of the system (8) can be defined as \(x(t) = \begin{bmatrix} m & x_{1o}(t) & x_{2o}(t) \end{bmatrix}\).

These \(m + 1\) completely represent the ARM/AQM system. The next step is to determine the controller (9) matrices in order to stabilize the system (8), and minimize specific measures \((y(t))\) of the reference output \(z(t)\). For that, it is necessary to define the target performance goals for both \(z(t)\), and \(y(t)\).

Given the similarity between the linearized systems of both controllers, the matrices that capture their dynamics are the same except for matrix \(B_u\).

The elements of matrix \(A_0\) contain the coefficient of the delay independent variables, related to the values of \(W\) and \(q\).
The first row contains \( \frac{\partial f}{\partial q} \) and \( \frac{\partial f}{\partial w} \) and the second row \( \frac{\partial f}{\partial t} \) and \( \frac{\partial f}{\partial z} \). Matrix \( A_1 \) incorporates the system delay. The elements of the first row are defined by \( \frac{\partial f}{\partial q} = \frac{\partial f}{\partial w} \) and the elements of the second row are null since \( \frac{\partial f}{\partial t} = \frac{\partial f}{\partial z} = 0 \).

In this way matrices \( A_0 \) and \( A_1 \) can be defined as:

\[
A_0 = \begin{bmatrix}
-\frac{N \mu^2}{m^2 R_0} & -\frac{1}{R_0 C} & 0
\end{bmatrix},
\quad A_1 = \begin{bmatrix}
-\frac{N \mu^2}{m^2 R_0} & \frac{1}{R_0 C} & 0
\end{bmatrix}
\]

Matrix \( B_u \), is related to the discard probability. \( H_2 \), involves \( p_r \), whereas \( H_2 \), involves \( p_y \). The matrices \( B_u \) of these controllers are given by:

\[
B_u = \begin{bmatrix}
-\frac{C \mu_a R_0}{2 N^2 s^2 \
\frac{2 N^2 s^2 R_0}{2 N^2 s^2}
\end{bmatrix},
\quad B_u = \begin{bmatrix}
-\frac{C \mu_a R_0}{2 N^2 s^2}
\end{bmatrix}
\]

Matrix \( B_w \), gives the system noise. For both controllers, the level of noise is chosen to be equal to 0.2C, i.e., 20% of the link capacity can carry UDP traffic. Such noise level is clearly an upperbound to those found in the Internet where roughly 83% of the bytes are carried by the TCP protocol [8]. Matrix \( C_y \) establishes that \( q \) is the variable to be controlled which is measured in the previous RTT period. \( D_{yw} \), weights the measured noise which is usually adopted as 10% of the values in \( B_u \) in the controller design:

\[
B_w = \begin{bmatrix}
0 & 0
0 & 0.2C
0 & 0.02C
\end{bmatrix},
\quad C_y = \begin{bmatrix}
0 & 1
\end{bmatrix},
\quad D_{yw} = \begin{bmatrix}
0 & 0.02C
\end{bmatrix}
\]

The objectives of the controllers dsH2-AQM are defined in the matrices \( C_{z_0}, C_{z_1} \) and \( D_{z_2} \). The objectives of both controllers are bandwidth guarantee and jitter minimization. These objectives can be achieved by minimizing the TCP window size (W) variation and the length of the queue (q). For that, it is necessary to define matrices \( C_{z_0}, C_{z_1} \) and \( D_{z_2} \) to be equal to the matrices \( A_0, A_1 \) and \( B_u \) of the linearized system (8), respectively.

The synthesis of the optimal controller involves the approach introduced in [2], in which linear delay systems are designed and solved as Linear Matrix Inequalities (LMIs).

Once performance goals have been defined, it is necessary to integrate system (8) with the controller, as follows:

Let \( x(t) \) be the augmented state vector which contains the state vector \( x(t) \) and the controller state vector \( \hat{x}(t) \):

\[
x(t) = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}
\]

The connection of system (8) with the controller (9) yields the linear delay system:

\[
\dot{x}(t) = A_0 \hat{x}(t) + A_1 \hat{x}(t - R_0) + B_w(t);
\]

\[
z(t) = C_0 \hat{x}(t) + C_1 \hat{x}(t - R_0) + D_w(t);
\]

where:

\[
A_0 = \begin{bmatrix}
A_0 & B_u \hat{C}_0 \\
0 & A_0
\end{bmatrix},
\quad A_1 = \begin{bmatrix}
A_1 + B_u \hat{D} C_y & B_u \hat{C}_1 \\
B C_y & A_1
\end{bmatrix},
\quad B = \begin{bmatrix}
B_w + B_u \hat{D} D_{yw} \\
B D_{yw}
\end{bmatrix},
\quad C_0 = \begin{bmatrix}
C_2 & D_{zu} \hat{C}_0 \\
D_{zu} \hat{C}_1
\end{bmatrix},
\quad D = D_{zu} D_{yw};
\]

Stability of system (10) is assured by using Theorem 4-b from [2].

In the solution obtained, matrices \( A_1 \) and \( \hat{C}_1 \) are approximately zero, and, can be ignored. As a result, the elimination of the delay terms in the system leads to rational controllers. When possible, delay elimination is the optimal solution for an \( H_2 \), norm minimization problem [2]. Thus, an optimal solution to the congestion control problem was found. The transfer functions in the frequency domain of the controllers corresponding to the performance goals are given by:

\[
C_{H2\text{out}}(s) = \frac{0.045793(s + 0.0005042)}{(s + 1.07e04)(s + 1.056e-07)};
\]

\[
C_{H2\text{in}}(s) = \frac{0.030116(s + 0.0003085)}{(s + 1.031e04)(s + 1.283e-07)};
\]

For a digital implementation of such controllers, it is necessary to choose a sampling frequency, so that a representation in the \( z \)-domain can be obtained. It is necessary to define the appropriate sampling frequency that not only avoids needless packet loss but also stabilizes the queue length. To determine the proper sampling frequency, values ranging from 10% to 50% of the link capacity, \( C_0 \), were investigated. The frequency chosen was 10% of the value of the link capacity, since the controllers obtained using this range of frequency differ just a little. Such value is common in the literature [10]. The transfer functions in the \( z \)-domain are:

\[
C_{H2\text{out}}(z) = \frac{a(z + 1)}{z + b} = \frac{2.4822^{-6}(z + 1)}{z + 0.1598};
\]

\[
C_{H2\text{in}}(z) = \frac{a(z + 1)}{z + b} = \frac{1.6677^{-6}(z + 1)}{z + 0.1417};
\]

The transfer function connecting \( \delta p \) (\( \delta p = p - p_0 \)) and \( \delta q \) (\( \delta q = q - q_0 \)) can be converted into difference equations at discrete times \( kT \), where \( T = \frac{1}{f} \), resulting in \( \delta p(kT) = a[\delta q(kT) + \delta q((k-1)T)] - b \delta p((k-1)T) \). Such representation is used to implement the algorithm and that computes the discard probability of the controllers \( H_2 \), and \( H_2 \), shown in Fig. 1.

V. NUMERICAL EXAMPLES

In this section, a comparison involving the dsH2-AQM dsPI-AQM [7] and RIO [11], is presented. Simulations were conducted using the ns simulator. Scenarios with only adaptive sources and with both with adaptive and non-adaptive sources were simulated. A dumbbell topology involving a link with
dsH2-AQM-Discard-Probability()

\[ p_0 \leftarrow \text{values defined for the equilibrium point}; \]
\[ p \leftarrow a(q - 2q_0 + q_{old}) - b p_{old} + p_0(1 + b); \]
\[ p_{old} \leftarrow p; \]
\[ q_{old} \leftarrow q; \]

Fig. 1. Algorithm for computing the discard probability

capacity of 155Mbps, buffer size of 12400 packets and propagation delay \((T)\) of 0.192 sec was used to derive the examples. Packet size is 500 bytes and the receiver window size was set to 1000 in order to avoid restrictions on the sender window due to flow control. The scenarios involve either 3 or 4 leaky buckets at the edge. The parameters for equilibrium point of the controller \(H_2\), are \(q_0 = 2480\) packets, \(RTT = 0.256\) sec, \(p_g = 0.0\), \(p_r = 0.0390625\) whereas for the controller \(H_{2g}\), they are \(q_0 = 4960\) packets, \(RTT = 0.320\) sec, \(p_g = 0.0\), \(p_r = 1\).

Traffic aggregates were generated by the traffic generator TrafficGen [12] and they were composed of both long (FTP) and short range (HTTP) duration flows. A distribution with a Lognormal body (88%) and Pareto tail (12%) was used to generate the size of web traffic. The Lognormal has mean 7247 and standard deviation 2876. The mean of the Pareto distribution was 10558 and the shape is 1.383 [13]. For FTP traffic file sizes an exponential distribution with mean of 512KB was used.

Queue length and goodput per connection are shown as functions of the network load which was varied from 0.4 to 1.0. The mean number of timeouts per connection and packet loss are not shown due to space limitation and since they impact the goodput.

The first two experiments involves only responsive flows and three aggregate flows at the edge. In the other two experiments, a fourth UDP aggregate flow was added considering that roughly 17% of the Internet traffic is carried by the UDP protocol [8].

A. Experiment 1

In the first group of experiments, only adaptive traffic is used. The MGRs values in a 20% over provisioned network are \(x_1 = 70.5\)Mbps, \(x_2 = 17.0\)Mbps e \(x_3 = 42.5\)Mbps and they are the same used in in [6]. The queue length produced by both controllers \(dsH2\) and \(dsPI\) vary as a function of the load. Conversely, the queue length produced by RIO does not vary as much as the ones produced by the other two controllers (Fig. 2). Note that the queue produced by \(dsH2\) is at most 36% of the buffer size. The goodput per connection produced by \(dsH2\) can be 37% greater than the one produced by \(dsPI\) for loads of 0.4 and 43% higher than the one given by RIO. For loads of 0.9 these differences are, respectively, 5% and 10% (Fig.3).

B. Experiment 2

In this experiment, the network is under provisioned and the MGRs are increased by 20% of the value used in the first experiment. It can be seen in Fig. 4 that the queue produced by \(dsH2\) can be 34% longer than the one produced by \(dsPI\) and 123% longer than the one produced by RIO. Nonetheless, such queue length is at most 57% of the buffer size, which does not yields to unnecessary packet loss. The highest value of queue length happens under loads varying from 0.9 and 1.0. The queue length produced by \(dsH2\) leads to goodput 16% higher than the one given by \(dsPI\) and 40% higher than the one given by RIO (Fig. 5).
C. Experiment 3

In this experiments, an under provisioned network was considered. Again, the MGRs of the aggregates are increased by 20% in relation to the scenario in Example 1 but under the influence of UDP traffic. The queue length produced by dsPI and RIO are similar (Fig. 6) and quite shorter than the one produced by dsH2. The queue produced by dsH2 is longer than those produced by the other two controllers under loads of 1.0. Such queue length avoids unnecessary losses and better utilization of network resources which has a significant impact on the goodput per connection. The goodput produced by dsH2 is higher than those produced by the other two controllers. For instance, for loads of 0.8 the goodput produced by dsH2 is 38% and 78% higher the one produced by dsPI and RIO, respectively (Fig. 7).

D. Experiment 4

In this experiments, the UDP load was increased to 0.3 which is both higher than the one found in the Internet and higher than the one dsH2 was designed for. The objective is to assess the performance of dsH2 even under scenarios in which the noise exceeds the one considered in the design. The queue produced by dsH2 can be, respectively, 100% and 150% longer than those produced by dsPI and RIO (Fig. 8) which assures higher goodput values per connection (Fig. 9). Even under non-responsive load higher than the one dsH2 was designed to operate, it produces goodput values 31% and 52% higher than those given by dsPI and RIO, respectively (Fig. 7).

VI. CONCLUSIONS

Guaranteeing bandwidth is certainly one of the major challenges in packet switched networks. In [6], a framework for providing guarantees was proposed for the AF DiffServ networks which involve ARM controllers at the edge and AQM controllers at the network core. This paper introduced an AQM controller based on Optimal control Theory. The advantage of using this theory is the introduction of the control objectives in the controller design. Results derived via simulation show that the dsH2 controller outperforms the dsPI and the RIO controllers.

REFERENCES