On Improving Bandwidth Assurance in AF-based DiffServ Networks Using a Control Theoretic Approach

XiaoLin Chang and Jogesh K. Muppala[1]
Dept. of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: muppala@cs.ust.hk

Abstract

The Assured Forwarding (AF) based service in a Differentiated Services (DiffServ) network fails to provide bandwidth assurance among competing aggregates under certain conditions, for example, where there exists a large disparity in the round-trip times, packet sizes, or target rates of the aggregates, or there exist non-adaptive aggregates. Several mechanisms have been proposed in order to address the problem of providing bandwidth assurance for aggregates, using only the knowledge gathered at ingress routers. In this paper, we present a control theoretic approach to analyze these mechanisms and explore the reasons when they fail to achieve bandwidth assurance under some circumstances. Then we propose a simple but robust controller for this problem, namely, the Variable-Structure Adaptive CIR Threshold (VS-ACT) mechanism. We validate the analysis and demonstrate that VS-ACT outperforms several other mechanisms proposed in the literature over a wide range of network dynamics through extensive simulations.

Keywords- Differentiated Services, Assured Forwarding Service, Bandwidth Assurance, Proportional-Integral Control, Adaptive, Marking Threshold

1. Introduction

The Differentiated Services (DiffServ) [1] approach is proposed as a scalable mechanism to address the insufficiency of the traditional Internet infrastructure in providing adequate Quality of Service (QoS) support. This is especially important to satisfy an ever-increasing number of diverse applications, each with different QoS requirements. Assured Forwarding (AF) [2] Per Hop Behavior (PHB) is one of the DiffServ forwarding mechanisms standardized by Internet Engineering Task Force (IETF), which offers different forwarding assurances to different customers based on their profiles. Recently, several research studies on the AF-based service performance within the current DiffServ framework, such as in [3]-[5], have brought forth some of the shortcomings of this service. In particular the AF-based service fails to provide bandwidth assurance (i) when there is a large difference in round-trip times (RTTs), packet sizes, target rates, or the number of micro-flows among the competing aggregates; (ii) when there exist extremely aggressive non-adaptive flows. The main reasons for this failure include (i) the TCP congestion control algorithm; (ii) non-adaptive flows showing no response to congestion indication. Several mechanisms such as in [5] and [10]-[23] have been proposed in order to alleviate this problem.

This paper focuses on applying a control theoretic approach to analyze and design mechanisms, which are employed at ingress routers in order to improve bandwidth assurance for aggregates based only on the knowledge, gathered at the ingress routers. We refer to such mechanisms as ingress-based mechanisms in this paper. These mechanisms improve bandwidth assurance in the AF-based DiffServ networks by *indirectly* influencing the rate allocated to flows/aggregates at congested routers through marking some packets as IN/OUT at the ingress routers. In this paper we assume that ingress routers have only output queues and only core routers may be congested. Discussions about ingress-based mechanisms in combined input and output queuing (CIOQ) switches are given in Section 8. We use routers and switches interchangeably in this following. Control theoretic approaches have been widely used to analyze and design mechanisms to improve the performance of various software systems [7], including *intelligent* AQM schemes such as in [5], [18] and [24], which improve network QoS by directly controlling flows at congested routers. Recently the authors in [17] designed an ingress-based mechanism by using the feedback control theory. However, to the best of our knowledge, no other work has considered a control-theoretic analysis of existing ingress-based mechanisms. In addition as shown in our simulation results, the fixed-gain controller [17] faces performance degradation in dynamic networks. A network is considered dynamic in this paper when, for example, there are changes in flow characteristics or changes in traffic load or changes in the network resources.

In this paper we present a generic Nonlinear Proportional-Integral (NPI)-type controller structure, where the proportional and integral gains are not static. Using this controller structure, we analyze some existing ingress-based mechanisms and explore the reason for when they fail to achieve bandwidth assurance. Then we develop an ingress-based mechanism, which is a self-tuning PI controller for adapting the *marking threshold*, namely, Variable-Structure Adaptive CIR Threshold (VS-ACT) in order to improve bandwidth assurance. Here, CIR represents the Committed Information Rate,

[1] Corresponding author. Tel: +852-23586978; fax: +852-23581477.

defined in the Service level Agreement (SLA) [1]. The *marking threshold* of a priority for an aggregate is the average rate of data transfer allowed at this priority level. In Time Sliding Window Three Color Marker (TSW3CM) [6], there are two marking thresholds, CIR and Peak Information Rate (PIR). Extensive simulations are carried out in this paper to investigate the dynamic (*transient* and *steady-state*) behavior of VS-ACT. *Transient behavior* captures the responsiveness and efficiency of a mechanism in reacting to the changes in the network conditions. The transient performance metrics considered here include the settling time and overshoot. *Steady-state behavior* captures the performance of the control system after the transient response settles. The steady-state performance metrics considered include steady-state error and sensitivity. Simulation results confirm our analysis and demonstrate that VS-ACT outperforms several other adaptive mechanisms in terms of the *transient* and *steady-state* performance in the process of improving bandwidth assurance over a wide range of network dynamics. VS-ACT achieves these by on-line adjustments to the controller gains based on the system states rather than on network parameters such as the maximum RTT, the number of active long-lived TCP flows, or the link capacity. The system states, used in this paper, include the deviation of the low-pass filtered average arriving rate from the aggregate's CIR and the change of the deviation.

The main contributions of this paper include: (i) applying a control theoretic approach to analyze some existing adaptive ingress-based mechanisms; (ii) using the control theoretic approach to design a simple but robust controller for improving bandwidth assurance; (iii) performing extensive simulation studies in support of our analysis and investigating the performance of VS-ACT.

The rest of this paper is organized as follows. First, we discuss some related work on improving bandwidth assurance in Section 2. Then we present a generic NPI-type controller structure and use it to analyze some existing mechanisms in Section 3. In Section 4, the VS-ACT mechanism is presented. We study the performance of VS-ACT and compare it with other mechanisms in Section 5. Finally we present the conclusions in Section 6.

2. Related Work

Fig.1 shows the framework of a Differentiated Services network. In this framework, the routers are divided into two categories, core routers and edge routers (including ingress routers and egress routers). Ingress routers are responsible for marking the DiffServ Codepoint (DSCP) of all the incoming packets according to the marking threshold. The core routers do not have any per-flow state and just differentiate packets based solely on the DSCP marking of the packets. Ever since Clark proposed the AF-based service framework by using RED with in/out (RIO) mechanism in [8], extensive performance studies of the AF service in this framework have been carried out. In order to improve the performance of AF service in this framework, some researchers design *intelligent* AQM schemes at core routers, for example in [5] and [18]. In this paper we only consider intelligent schemes designed at edge routers to improve bandwidth assurance with the assumption that no *intelligent dropping*/*scheduling schemes* are implemented at core routers. This assumption is consistent with the DiffServ approach of pushing the complexity to the network edges in order to keep the core of the network simple and scalable. Note that the mechanisms employed at ingress routers and the *intelligent* AQM *schemes* at core routers are complementary and can be used in conjunction with each other.

According to the number of levels of drop preference in an AF class, there are two kinds of traffic conditioners: (i) two-color based and (ii) three-color based. Three-color based conditioners, such as TSW3CM, are proposed in order to improve the fair sharing of excess bandwidth. Through simulations, the authors in [9] conclude that utility of three levels of drop precedence in a traffic class depends on the traffic load, the sum of target rates and the available link capacity.

Intelligent traffic conditioners at ingress routers have been proposed in [10]-[17] which use the knowledge gathered at the ingress router to improve bandwidth assurance in the context of the AF-based services. Before we proceed, we define a few terms used subsequently in the paper. We refer to a flow/aggregate as *unsatisfied* when its bandwidth assurance is not achieved; otherwise it is considered *satisfied*. We refer to a flow/aggregate as *conditionally-satisfied* when its bandwidth assurance is achieved by increasing CIR_{Thresh} larger than CIR. The essence of the remedies suggested in [10]-[17] is to increase the allowed maximum low-pass filtered arriving rate of IN traffic of the *unsatisfied* flow/aggregate so that a larger amount of IN traffic of this flow/aggregate is injected into the domain to compensate for the performance loss caused by the dropped/ECN-marked low priority packets. Some authors, for example in [10]-[13], implement this by incorporating TCP flow characteristics (such as RTT, packet size, Retransmission Time-Out) into the computation of marking probabilities. These remedies can be applied only to individual flows or require that all the micro-flows in an aggregate be identical. Some intelligent schemes for aggregates based on aggregate information have been proposed, such as in [14]-[17]. Usually, this aggregate information is the low-pass filtered average arriving rate, which is computed at ingress routers either periodically or upon a packet arrival. In order to reduce the large performance fluctuation and reduce the sensitivity to control parameter settings, the authors in [14] propose the *Memory-based Marking* (MBM), which adjusts marking probabilities (defined as *mp*) upon a data packet arrival by using the current average arriving rate and the previous average arriving rate. Note that the average arriving rate is computed also upon the arrival of a packet. In [15] the authors propose a *Packet Marking Engine* (PME, employed at routers), which uses the periodically estimated average arriving rate as the decision-making factor in the process of adjusting the marking probability periodically. The authors in [16] develop an *Adaptive CIR Threshold* (ACT), which adapts the marking threshold (defined as CIR_{Thresh}) periodically. Note that MBM, PME and ACT utilize feedback control in an *ad hoc* manner. Very little is known about why they work and very little

explanation can be given when they fail under some circumstances. By using classical linear control theory, the authors in [17] propose a fluid flow model for the dynamics in the AF-based DiffServ network and develop the *Active Rate Management* (ARM), which regulates the token bucket rate at ingress routers to guarantee the minimum bandwidth requirement. Token bucket rate is a kind of marking threshold. To our knowledge, ARM is the first mechanism that has been designed based on feedback control theory. It consists of a fixed-gain PI controller and a low-pass filter. ARM also uses the periodically estimated average arriving rate as the decision-making factor in adjusting the token bucket rate periodically. Note that *mp* of a *satisfied* aggregate in PME or the token bucket rate of a *satisfied* aggregate in ARM may be decreased to zero. However, ACT uses CIR as the lower bound of CIR_{Thresh}.

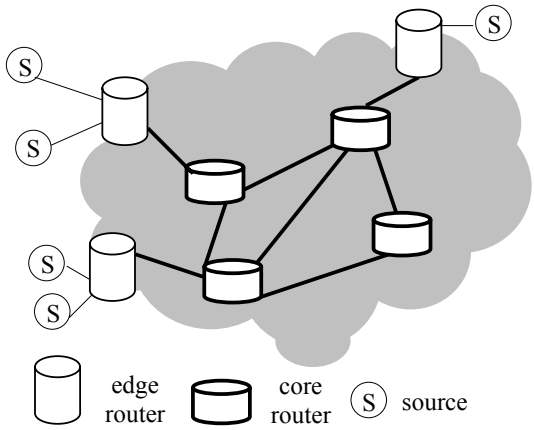


Fig.1. DiffServ architecture

In addition, intelligent mechanisms at ingress routers based on explicit feedback messages, sent from other routers to the ingress routers, have been proposed, which can be classified into two categories, *active* and *passive*. In the passive mechanisms, such as in [19]-[21], control packets are sent periodically by ingress routers in order to trigger other routers to generate feedback messages. The problem caused is that bandwidth is still consumed by the control packets even if there is no performance improvement. In active mechanisms such as in [22] and [23], feedback messages are generated at the core or egress routers. The main problem caused is their limited ability in detecting whether the failure of bandwidth assurance occurs. Thus, the authors in [22] propose to combine the active feedback mechanism with ACT [16].

3. A Generic NPI-type Controller Structure for Improving Bandwidth Assurance

In this section, we first introduce a generic NPI-type controller structure. Then, we use it to analyze several existing adaptive mechanisms, including PME [15], MBM [14], ARM [17], and ACT [16].

3.1. A generic NPI-type controller structure

Without loss of generality, we assume that each aggregate is served by a separate ingress router. The traffic of all aggregates feed into a core router. Fig.2 depicts the closed-loop architecture of the combined *Adjusting-Algorithm*/AQM AF-based DiffServ network, where each bold inner loop denotes an AF-feedback-loop. This loop is invoked at every sampling instant. Each dotted box denotes an ingress router. In the context of control theory, the *Adjusting-Algorithm* is a controller, employed at ingress routers in order to improve bandwidth assurance by increasing the amount of IN priority traffic based only on the knowledge gathered at the ingress router; "*Aggregate i Dynamics*" ($0 \le i \le n$, n is the number of ingress routers) is the plant in the AF-feedback-loop. The sensor aims to measure the arriving rate.

In general, the implementation of a controller requires considering four major inter-related aspects: (i) identify the three basic control-related variables, the *reference value*, *controller input* and *controller output*; (ii) define the controller structure; (iii) design the controller gain adjusting algorithms; (iv) set the control parameters used in the algorithms.

We define the *controller input* at time instant k as $e(k)=\text{CIR}-r_a(k)$, where CIR is the *reference value*, that is, the target rate defined in the SLA. Here, due to the bursty nature of the network traffic and other perturbations, r_a is defined as the low-pass filtered arriving rate. Now the key issue is defining the *controller output*. Some mechanisms such as PME and MBM improve bandwidth assurance by adjusting the DSCP marking probability, *mp*; others such as ARM and ACT improve bandwidth assurance by adjusting CIR_{Thresh}. Since the essential idea of all the mechanisms is to improve the amount of IN priority traffic, the *controller output* should completely determine the allowed maximum low-pass filtered arriving rate of IN priority traffic. When the *controller output* is defined as *mp*, this maximum rate is still affected by r_a. However, when the *controller output* is CIR_{Thresh}, the *controller output* completely determines this maximum rate. Thus, we

choose CIR_{Thresh} as the *controller output* in our generic NPI-type controller structure. Correctly defining the *controller output* benefits the explanation of a mechanism.

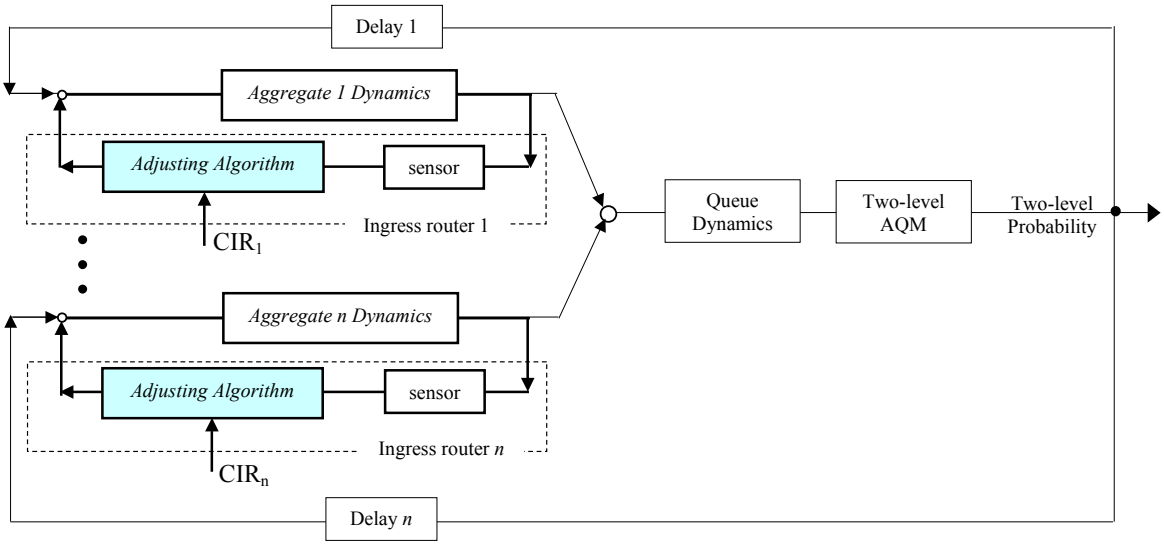


Fig.2. The combined *Adjusting-Algorithm*/AQM AF-based DiffServ network

Now we consider designing the controller structure. The authors in [24] discuss the limitations of applying a pure proportional controller for AQM. Similar limitations exist when a pure proportional controller is applied to adjust CIR_{Thresh}. Adding Integral control can alleviate these limitations. Eq.(1) gives the digital approximate implementation of a *continuous*-time NPI-type controller structure [25], obtained by applying Trapezoidal rule [26].

$$CIR_{\text{Thresh}}(k) = CIR_{\text{Thresh}}(k-1) + K_{\text{p}}(k)\big(e(k) - e(k-1)\big) + K_{\text{i}}(k)\big(e(k) + e(k-1)\big) \tag{1}$$

Eq.(1) is the combination of the proportional control action $CIR_{\text{Thresh}}(k) = CIR_{\text{Thresh}}(k-1) + K_{\text{p}}(k)\big(e(k) - e(k-1)\big)$ and the integral control action $CIR_{\text{Thresh}}(k) = CIR_{\text{Thresh}}(k-1) + K_{\text{i}}\big(e(k) + e(k-1)\big)$. We do not use Derivative (D) control because the network traffic is bursty and Derivative control may amplify the noise. In Eq.(1), $K_{\text{p}}(k)$ and $K_{\text{i}}(k)$ are both controller gains, respectively representing the proportional gain and the integral gain. When $K_{\text{p}}(k)$ and $K_{\text{i}}(k)$ are both constants, Eq.(1) is a linear controller. Although a fixed-gain PI controller is an effective controller for a static system, its ability is degraded in the presence of network parameter variations and disturbances. A controller with varying gains, tuned appropriately, can produce better performance than the fixed-gain controller, while still enjoying the simple structure of the fixed-gain PI controller.

3.2. The rules of tuning controller gains

Designing an efficient NPI controller requires proper tuning of K_{p} and K_{i} in order to produce small *overshoot*, short *settling time*, small *steady-state error* and low *sensitivity* in dynamic networks. *Settling time* reflects how fast the bandwidth assurance is re-achieved after changes in the network conditions. *Overshoot* represents the value of $\frac{r_a^{\max} - \text{CIR}}{\text{CIR}}$, where $r_a^{\max}$ is the maximum value of r_{a} during its transient phase. A large *overshoot* damages the benefits of other aggregates. *Sensitivity* represents how significantly the changes in network resources or traffic characteristics affect the attainment of bandwidth assurance for *unsatisfied* aggregates. *Sensitivity* describes the robustness of the control system with respect to these changes.

K_{p} and K_{i} can be tuned by applying the indirect adaptive control approach [27], which uses the estimated network parameters to update the controller gains. This approach has been used in [28], [29] and [30]. In this paper we attempt the system-state-based self-tuning method to adjust K_{p} and K_{i}, which is an implementation of the direct adaptive control approach [27]. In this method, K_{p} and K_{i} are designed as functions of the system output, r_{a}.

Now we discuss how to use $|e(k)|$ to adjust K_{p} and K_{i} according to the features of the proportional control action and the integral control action. When the characteristics of a *satisfied* aggregate or the environmental conditions (such as characteristics of other aggregates) change, the aggregate's ability in grabbing bandwidth may change. That is, the operating point of CIR_{Thresh} of this aggregate may change. This change must result in changes in $|e(k)|$. Usually, the larger

the $|e(k)|$, the larger the impact of the network conditions on the aggregate. That is, a large |e(k)| usually means that the current CIR_{Thresh} is far away from the new operating point. The proportional control action, $K_p(k)e(k)$, changes CIR_{Thresh} in proportion to the value of $e(k)$ and in the direction, which reduces $e(k)$ [26]. The integral action changes CIR_{Thresh} incrementally, in proportion to the time integral of previous errors. A large K_p and K_i can produce faster transient response with possible instability [27]. Thus, when there is no disturbance, it is reasonable to design K_p and K_i as increasing functions of $|e(k)|$. Then CIR_{Thresh} can quickly reach the new operating point. In addition, the possible instability caused by using large constant K_p and K_i to speed up transient response is alleviated. However, when considering disturbance, the rules of adjusting K_i are becoming complex [27]. Thus, in Section 3.3 we only analyze the proportional gain in each mechanism. As shown in the simulation results in Section 5, this analysis method can give insights into the behaviors of PME, MBM, ARM, and ACT.

3.3. Analysis of some existing mechanisms

We use the above discussions to analyze PME, MBM, ARM and ACT. We have mentioned that r_a is a low-pass filtered average arriving rate. A fluid model for this dynamics is given in Eq.(A.4) in Appendix A. In order to simplify the analysis, we ignore this dynamic. In addition, we ignore the low-pass filter when we analyze ARM. Based on these assumptions, we can map PME, MBM, ARM, and ACT to the NPI-type controller structure in Eq.(1).

3.3.1. PME

PME uses Eq.(2) to update $mp(k)$ at time instant k. η is a positive constant.

$$mp(k) = mp(k-1) + \eta\left(1 - \frac{r_a(k)}{\text{CIR}}\right) \tag{2}$$

By letting $e(k)$=CIR- $r_a(k)$ and $CIR_{\text{Thresh}}=mp(k)\ r_a(k)$, we obtain

$$CIR_{\text{Thresh}}(k) = CIR_{\text{Thresh}}(k-1) + \frac{r_a(k)}{\text{CIR}}\frac{\eta}{2}\big(e(k)-e(k-1)\big) + \frac{r_a(k)}{\text{CIR}}\frac{\eta}{2}\big(e(k)+e(k-1)\big) \tag{3}$$

Thus, $K_p(k) = \frac{r_a(k)}{\text{CIR}}\frac{\eta}{2}$, showing that: (i) When $r_a \le$CIR, the controller is a NPI controller, where K_p is a non-decreasing function of r_a and a non-increasing function of CIR. Thus, an *unsatisfied* aggregate with small r_a or with large CIR has a slow speed of increasing CIR_{Thresh}. (ii) When $r_a >$CIR, the controller is also a NPI controller. It is obvious that K_p in the case of $r_a \le$CIR is always smaller than K_p in the case of $r_a >$CIR. When an aggregate exceeds its CIR through increasing CIR_{Thresh}, the decreasing action of K_p causes the arriving rate to quickly go below CIR. However because the speed of increasing CIR_{Thresh} is slow, it takes a long time to recover back towards the CIR from below. Thus, the average goodput is below CIR for a long time. Thus, it is difficult for the unsatisfied aggregates to approximate the average goodput close to their CIRs. Note that such decreasing and increasing methods result in the small difference in the achieved average goodput among the AF adaptive aggregates, compare to TSW.

3.3.2. Memory-based marker

MBM uses Eq.(4) to update mp. $v(k)$ denotes the average arriving rate estimated at time instant k.

$$\begin{cases} mp(k) = mp(k-1) + \left(1 - \dfrac{v(k)}{\text{CIR}}\right) + \dfrac{v(k-1)-v(k)}{v(k)} & v(k) \le \text{CIR} \\ mp(k) = mp(k-1) + \dfrac{v(k-1)-v(k)}{v(k)} & v(k) > \text{CIR} \end{cases} \tag{4}$$

We rewrite Eq.(4) into Eq.(5) by letting $e(k)$=CIR-$v(k)$ and $CIR_{\text{Thresh}}= v(k)mp(k)$.

$$\begin{cases} CIR_{\text{Thresh}}(k) = CIR_{\text{Thresh}}(k-1) + \left(\dfrac{1}{2}\dfrac{v(k)}{\text{CIR}} + 1\right)\big(e(k)-e(k-1)\big) + \dfrac{1}{2}\dfrac{v(k)}{\text{CIR}}\big(e(k)+e(k-1)\big) & v(k) \le \text{CIR} \\ CIR_{\text{Thresh}}(k) = CIR_{\text{Thresh}}(k-1) + \big(e(k)-e(k-1)\big) & v(k) > \text{CIR} \end{cases} \tag{5}$$

Note that MBM updates mp whenever a packet arrives. Eq.(5) shows that (i) When $v(k) \le$CIR, $K_p(k) = \frac{v(k)}{2*\text{CIR}} + 1$. Thus, the controller is an NPI-type controller. The features of the MBM in this case are similar to PME. The *unsatisfied*

aggregate with smaller r_a or with larger CIR has a slow speed of increasing CIR_{Thresh}. (ii) When $v(k)$>CIR, the controller is a fixed-gain Proportional (P) controller. The static feature when $v(k)$>CIR and the slow increasing feature when $v(k)$≤CIR enlarges the difference in the achieved Average Goodput among the competing AF adaptive aggregates, compared to TSW.

3.3.3. ARM

ARM consists of a fixed-gain PI controller and a low-pass filter. We ignore the low-pass filter. Then it can be mapped to Eq.(6). The token bucket rate is a kind of CIR_{Thresh}. a and b are positive constants.

$$CIR_{Thresh}(k) = CIR_{Thresh}(k-1) + ae(k) - b(k-1) \tag{6}$$

Eq.(6) is a fixed-gain PI controller, where $K_p(k) = \frac{a+b}{2}$ and $K_I(k) = \frac{a-b}{2}$. We have previously mentioned the disadvantage of a fixed-gain controller. But the following simulation results show that ARM produces a fast response in most cases. The main reason is that ARM uses zero as the lower bound of CIR_{Thresh}. We discuss the drawbacks of using zero as the lower bound in Section 4.2.2.

3.3.4. ACT marker

ACT uses Eq.(7) to update CIR_{Thresh} at time instant k. γ and β are both positive constants.

$$\begin{cases} CIR_{Thresh}(k) = CIR_{Thresh}(k-1) + \gamma \mathrm{CIR}\left[1 - \frac{r_a(k)}{\mathrm{CIR}}\right] & r_a(k) \le \mathrm{CIR} \;\&\&\; CIR_{Thresh} < 2.0\mathrm{CIR} \\ CIR_{Thresh}(k) = CIR_{Thresh}(k-1) - \beta \mathrm{CIR}\left[1 - \frac{\mathrm{CIR}}{r_a(k)}\right] & r_a(k) > \mathrm{CIR} \;\&\&\; CIR_{Thresh} > \mathrm{CIR} \end{cases} \tag{7}$$

We rewrite Eq.(7) into Eq.(8) by letting $e(k)$=CIR- $r_a(k)$.

$$\begin{cases} CIR_{Thresh}(k) = CIR_{Thresh}(k-1) + \frac{\gamma}{2}\left(e(k) - e(k-1)\right) + \frac{\gamma}{2}\left(e(k) + e(k-1)\right) & r_a(k) \le \mathrm{CIR} \;\&\&\; CIR_{Thresh} < 2\mathrm{CIR} \\ CIR_{Thresh}(k) = CIR_{Thresh}(k-1) + \frac{1}{2}\frac{\beta \mathrm{CIR}}{r_a(k)}\left(e(k) - e(k-1)\right) + \frac{1}{2}\frac{\beta \mathrm{CIR}}{r_a(k)}\left(e(k) + e(k-1)\right) & r_a(k) > \mathrm{CIR} \;\&\&\; CIR_{Thresh} > \mathrm{CIR} \end{cases} \tag{8}$$

Eq. (8) shows that (i) When r_a≤CIR, the controller is a fixed-gain PI controller, where $K_p(k) = \frac{\gamma}{2}$ and $K_I(k) = \frac{\gamma}{2}$. (ii) When r_a>CIR, the controller is a NPI controller, where $K_p(k) = \frac{1}{2}\frac{\beta \mathrm{CIR}}{r_a(k)}$, a non-increasing function of r_a and non-decreasing function of CIR. Thus, $K_p(k)$ is a non-increasing function of $|e(k)|$. It is noted that the features of the control gains in ACT in this case are opposite to those of PME. This slowly-decreasing feature in adjusting the CIR_{Thresh} of the *conditionally-satisfied* aggregate with large r_a when r_a >CIR is undesirable. In addition, setting γ larger than β in [16] also reduces the speed of decreasing CIR_{Thresh}. In the following sections we use the term *slowly-decreasing* method to represent the use of these two features that cause the slow speed in decreasing CIR_{Thresh} of the *conditionally-satisfied* aggregate. Although the *slowly-decreasing* method can accelerate the attainment of bandwidth assurance for some *unsatisfied* aggregates, it may result in an excessive increase in CIR_{Thresh} of some *conditionally-satisfied* aggregates. This results in the Average Goodput of these aggregates being larger than their CIRs. A serious side effect is that the excessive amount of IN traffic may prevent weak *unsatisfied* aggregates, such as those with large RTT, large CIR and the like, from improving their goodput. It may also result in the unfair sharing of excess bandwidth among aggregates. We see these effects in the simulations presented later.

3.4. Summary

According to the above analysis, we list K_p and K_i of each mechanism in TABLE I. From this table, we can see that either these mechanisms are fixed-gain controllers or the controller gains are adjusted contrary to what is desired. As shown in the following simulation results, a fixed-gain controller performs well in some network situations but performs worse in other situations. However, a controller with an undesirable design of controller gains either results in the bandwidth attainment over CIR or can't improve bandwidth assurance.

TABLE I CONTROL GAINS IN EACH SCHEME

		PME		MBM		ACT		ARM	
For *unsatisfied* aggregate	K_p	$\frac{r_a(k)}{\text{CIR}}\frac{\eta}{2}$	x	$\frac{1}{2}\frac{v(k)}{\text{CIR}}+1$	x	$\frac{\gamma}{2}$	fixed-gain	$\frac{a+b}{2}$	fixed-gain
	K_i	$\frac{r_a(k)}{\text{CIR}}\frac{\eta}{2}$		$\frac{1}{2}\frac{v(k)}{\text{CIR}}$		$\frac{\gamma}{2}$		$\frac{a-b}{2}$	
For *satisfied* aggregate	K_p	$\frac{r_a(k)}{\text{CIR}}\frac{\eta}{2}$	√	1	fixed-gain	$\frac{1}{2}\frac{\beta\text{CIR}}{r_a(k)}$	x	$\frac{a+b}{2}$	
	K_i	$\frac{r_a(k)}{\text{CIR}}\frac{\eta}{2}$		0		$\frac{1}{2}\frac{\beta\text{CIR}}{r_a(k)}$		$\frac{a-b}{2}$	

Note: *x* *represents that the settings of controller gains have adverse effect on performance.*
√ *represents that the settings of controller gains can bring help to the performance improvement.*
Fixed-gain *represents that the controller gains are static.*

4. Variable-Structure PI Controller for Adapting CIR Threshold

In this section we present VS-ACT and discuss some design considerations.

4.1. The VS-ACT mechanism

Based on the above discussions, we develop a Variable-Structure PI controller for adapting CIR_{Thresh}. The initial value of CIR_{Thresh} is set to CIR. The VS-ACT mechanism acts as follows: (i) when r_a<CIR, CIR_{Thresh} is increased step by step until 2CIR; (ii) when r_a>CIR and CIR_{Thresh}≥CIR, CIR_{Thresh} is decreased step by step until CIR. The formula of adjusting CIR_{Thresh} is depicted by

$$\begin{cases} CIR_{\text{Thresh}}(k) = CIR_{\text{Thresh}}(k-1) + \varphi(k)\left[k_{\text{pmin}}\left(e(k)-e(k-1)\right) + k_{\text{imin}}\left(e(k)+e(k-1)\right)\right] \\ CIR_{\text{Thresh}}(k) = \max\left\{\text{CIR}, \min\left\{2\text{CIR}, CIR_{\text{Thresh}}(k)\right\}\right\} \end{cases} \tag{9}$$

where

$$\varphi(k) = \beta\frac{k_{\max}}{1+\exp\left[-\left[e(k)/\eta\right]^2\right]} \quad \text{and} \quad \beta = \begin{cases} 0.75 & |e(k)| < |e(k-1)| \\ 1.0 & \text{otherwise} \end{cases} \tag{10}$$

k_{pmin} and k_{imin} in Eq.(9) and $k_{\max}$ in Eq.(10) are positive constants. Their settings are discussed in Section 4.2.3. η and β are user-defined positive constants.

4.2. Design Considerations

4.2.1. The formula

Fig.3 depicts the block diagram of a AF-feedback-loop system with VS-ACT. It is easy to see that VS-ACT is based on modulating the control output of a fixed-gain PI controller with $\varphi(k)$, which is a modified *sigmoidal* function of $|e(k)|$. The reason for using the modified *sigmoidal* function rather than other kinds of functions such as the *hyperbolic* function or the *piecewise-linear* function is the consideration that (i) the exponential term can produce a fast increase or a fast decrease; (ii) it is much easier to bound the function value when using a smooth *sigmoidal* function; (iii) we need $K_p(k_1)=K_p(k_2)$ when $|e(k_1)| = |e(k_2)|$ when the moving directions at both time k_1 and time k_2 are the same (towards the CIR or away from CIR); thus we make modification to the standard *sigmoidal* function.

Eq.(9) and Eq.(10) show that $K_p(k)$ and $K_i(k)$ are both designed as non-decreasing functions of $|e(k)|$. The motivation of varying $K_i(k)$ proportional to $|e(k)|$ is that varying $K_i(k)$ proportional to $|e(k)|$ can produce fast transient response. In addition, the low-filtered arriving rate can accommodate some disturbance. Even if there is unnecessary accumulation in CIR_{Thresh}, the accumulation may be not large and may be quickly released because $K_i(k)$ is proportional to $|e(k)|$. In order to reduce the unnecessary accumulation due to disturbance and in order to prevent instability, we use small k_{pmin} and k_{imin} when $|e(k)|$ is small and the increasing speed of $K_p(k)$ and $K_i(k)$ is also small when $|e(k)|$ is not large. These are achieved by using η and

"square" in Eq. (10). Note that "square" can speed up the increase of $K_p(k)$ and $K_i(k)$ when $|e(k)| > \eta$. The time delay and the existence of the other aggregates may result in the excessive increase in $K_p(k)$ and $K_i(k)$, leading to instability. In order to reduce the possibly excessive increase/decrease in CIR_{Thresh}, $K_p(k)$ and $K_i(k)$ are varying between $[k_{pmin}, k_{max}+k_{pmin}]$ and $[k_{imin}, k_{max}+k_{imin}]$, respectively.

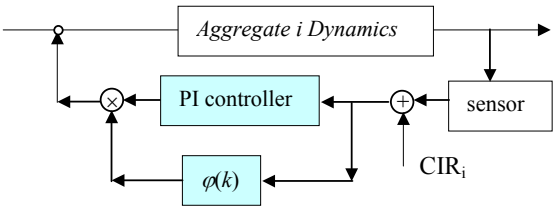


Fig.3. The *i*-th AF-feedback-loop system with VS-ACT controller as the *Adjusting-Algorithm*

Now, we give the reason for using β. The control action of VS-ACT depends on the movement of r_a toward CIR or away from CIR. Fig.4 is an example about the variation of $e(k)$ over time. The controller gains should be smaller in thick curves, where $|e(k)| < |e(k\text{-}1)|$. The reason is that when $e(k_1)=e(k_2)$ and the moving direction of $r_a(k_1)$ is away from CIR and the moving direction of $r_a(k_2)$ is towards CIR, a large K_p at time k_2 may lead to unnecessary increase or decrease in CIR_{Thresh}. We use β to achieve this goal.

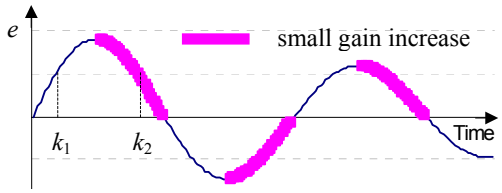


Fig.4. Illustration of function of *e* over time

4.2.2. Using upper and lower bounds

We have mentioned that VS-ACT uses CIR as the lower bound of CIR_{Thresh} and 2CIR as the upper bound. The goal of using the upper and lower bounds is similar to the goal of anti-windup [26] strategies in the classical control theory. It is possible that any further increase in CIR_{Thresh} does not lead to any improvement in bandwidth assurance when CIR_{Thresh} increases past a certain point. If the integration of $e(k)$ continues in this case, the value of CIR_{Thresh} becomes very large without any performance improvement. $e(k)$ then has to be of the opposite sign for a long time to bring the value of CIR_{Thresh} back to its steady-state value when the network conditions are changing. Thus, if there is no upper bound, there may exist an adverse impact on other aggregates improving bandwidth assurance and there may be an adverse impact on the fair sharing in excess bandwidth among aggregates when the network conditions are varying. In addition, when CIR_{Thresh} or *mp* is allowed to be zero, the performance of the aggregate itself in achieving bandwidth assurance and sharing in excess bandwidth is degraded in some situations. The simulations in Section 5.4 illustrate the importance of using upper and lower bounds. Choosing 2CIR as the upper bound is motivated by the multiplicative-decrease feature in the TCP congestion control algorithm.

4.2.3. Setting control parameters

The settings of k_{max}, k_{pmin} and k_{imin} are critical to the performance of the system. Recently the authors in [31] have analyzed the stability of the system in [17], where ARM is employed at the ingress routers and a two-level PI controller is employed as AQM at the core router. We use the same approach to analyze the stability of the system, where (i) the ingress router uses the TSW profiler to provide two-level edge coloring and uses a fixed-gain PI-type marker to adjust CIR_{Thresh}; (ii) RIO is used as AQM at the core router. The details are given in [33]. The salient steps of the analysis are given in Appendix A.1. We derive the sufficient conditions for system stability as given in condition (A.8) in Appendix A.1. Note that using such conditions to derive k_{max}, k_{pmin} and k_{imin}, if not impossible, is hard work, especially when there are a large number of aggregates involved.

However, these conditions provide some theoretical guidelines for selecting k_{max}, k_{pmin} and k_{imin}. When we choose k_{pmin} and k_{imin}, we ignore the transient behavior and focus on the steady-state behavior. In the sufficient conditions derived for

stability, the upper bounds are approximately decreasing functions of RTT and increasing functions of N (the number of micro-flows of an aggregate). Thus, letting k_{max}=0, we choose k_{pmin} and k_{imin}, in the scenario, where (i) each adaptive aggregate has the same characteristics; (ii) the minimum N^- and the large propagation delay RTT^+ are used; (iii) CIR is set to the average of the possible values used in all the simulations; (4) setting the sending rate of non-adaptive aggregates such that the subscription level of the bottleneck link is light. The *subscription level* of a link is defined as the ratio of the sum of CIRs of adaptive aggregates and the sending rates of non-adaptive aggregates to the link bandwidth. We repeat simulations to find out the large values of k_{pmin} and k_{imin} such that the goodput of each unsatisfied aggregate can approximate CIR. The motivation here is that the system with this VS-ACT at ingress routers is locally stable in the range of $N \geq N^-$ and $RTT \leq RTT^+$ when the system (N^- ,RTT^+) with VS-ACT, using (k_{pmin}, k_{imin}) and k_{max}=0, is stable.

We choose the value of k_{max} in order to speed up the transient response without sacrificing the stability. We set the ability of grabbing bandwidth of each aggregate different in a large degree (we do it by assigning them with different propagation delay or with different CIRs) and the subscription level of the bottleneck link is heavy, such as the network scenarios in 5.1.3 and 0. We repeat simulations until the transient response is satisfactory while the steady-state behavior is satisfactory.

5. Simulation Results

We use *ns-2* [32] to evaluate the effectiveness of VS-ACT and compare its performance with TSW, PME, MBM, ACT, and ARM.

The network topology used for simulations is shown in Fig.5. In this figure S_i/D_i ($1 \leq i \leq 10$) is source/destination node; I_i is ingress router; E_1 is egress router; and C_1 is core router. The link delay between E_1 and D_i is 10ms. The capacities and delays of other links are set to 20Mbps and 5ms, respectively. There are 10 aggregates (A_1-A_{10}), where A_i is from S_i to D_i. An adaptive aggregate is defined as consisting only of identical adaptive micro-flows, which respond to congestion. A non-adaptive aggregate is defined as consisting only of identical non-adaptive micro-flows, which do not respond to congestion. We summarize the attributes of each aggregate in TABLE II. We employ UDP sources sending constant bit rate (CBR) traffic as an example of non-adaptive sources. The sending rates of A_9 and A_{10} are both 5.0Mbps. We use TCP sources generating infinite FTP bulk data as adaptive sources. The TCP sources are based on the TCP-Reno implementation.

C_1—E_1 is the bottleneck link and it is *implicitly over-subscribed*. The subscription level is 120%. Here, an *under-subscribed* (*exact-subscribed*) link is referred to as the link where the sum of CIRs of all competing aggregates is less than (equal to) the link capacity; an *implicit over-subscribed* link refers to a kind of *under-subscribed* links where the sum of CIRs of adaptive aggregates and the sending rates of non-adaptive aggregates is larger than the link capacity.

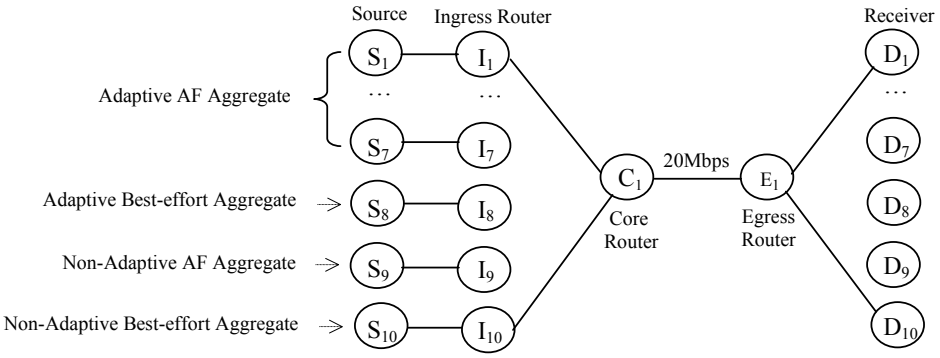


Fig.5. Network topology

Some notations and the corresponding parameters used in the following section are defined in TABLE III. In PME, we find that when η is set to 0.045 PME performs better than setting other values to η in the following simulations. We use the time sliding window (TSW) profiler at the ingress routers when doing simulations with TSW, PME, MBM, ACT and VS-ACT and we use the token bucket profiler when doing simulations with ARM. The input of PME, MBM, ACT, and VS-ACT is computed by using the Exponentially Weighted Moving Average (EWMA) technique with the 1-second period and the weight of 0.8. The arriving rate in the 1-second period is computed by measuring the number of arriving packets over 1-second. We use 1.0 second as the sampling interval to update the marking probability in PME and 2.0s as the time interval to adjust CIR_{Thresh} in ACT and VS-ACT. The reason for choosing 2.0s is that the maximum RTT is in (1.0, 2.0)s in experiment 0. The reason for using 1.0s rather than 2.0s in PME is that the transient response in most experiments is too slow if we use 2.0s. For ARM, the parameters used in the controller are set as suggested in [17]. In [17], the sampling interval for adapting the token bucket rate is set to 1/37.5s and the time interval for computing the average arriving rate is

set to 1.0s. RIO is used as the AQM for all the following simulations. Note that the authors in [17] evaluate ARM by applying the two-level PI controller as AQM, which is different from our paper. The RIO parameters, [q_{min}, q_{max}, p_{max}] for IN and OUT, are set to [150,400,0.02], [80,150,0.1], respectively. Adaptive hosts and network routers are ECN-enabled. The packet size at routers is 1000bytes. Unless otherwise specified, the above settings are used as default values in the following simulations.

In the following we consider both static and dynamic scenarios. By static networks, we mean that network configurations and traffic characteristics is not changed during the whole simulation. The simulations in the static scenarios aim to examine the steady-state behavior of the various mechanisms. The performance metric is the Average Goodput, computed by measuring the number of packets received at the receiver over a specified time period after the network is in the quasi-stable state. The simulations in the dynamic scenarios aim to examine the transient behavior of the various mechanisms. The performance metric is the Average Goodput (estimated per 5.0 seconds) variation in the simulation period.

TABLE II ATTRIBUTES OF AGGREGATES

Aggregate		# of micro-flows	Packet size (bytes)	CIR (Mbps)	Round Trip Propagation Delay (ms)
Adaptive aggregate	A_1	5	1000	2.0	50
	A_2	5	1000	2.0	50
	A_3	5	1000	2.0	50
	A_4	5	1000	2.0	50
	A_5	5	1000	2.0	50
	A_6	5	1000	2.0	50
	A_7	5	1000	2.0	50
	A_8	5	1000	0.0	50
Non-Adaptive aggregate	A_9	1	1000	2.0	50
	A_{10}	1	1000	0.0	50

TABLE III SCHEMES

Mechanism	Parameters
TSW	Time Sliding Window Two Color Marker
MBM	Memory-based Marking
PME	Packet Marking Engine, η= 0.045
ACT	Adaptive CIR Threshold, γ=0.05, β=0.025
VS-ACT	k_{pmin}=0.03, k_{imin}=0.028, k_{max}=0.03, η=0.5Mbps, β=75%

5.1. Static network scenarios: *under-subscribed*

The simulations in this section examine the steady-state performance of each scheme in *under-subscribed* networks. So far we have assumed that all the micro-flows in an aggregate are identical. Thus, when we exclude the impact of non-adaptive aggregates, the main elements that affect the ability of an adaptive aggregate in achieving bandwidth assurance are (i) the number of micro-flows in the aggregate; (ii) CIR of the aggregate; (iii) micro-flow characteristics such as packet size and RTT. We study the impact of each of these attributes on the Average Goodput. We vary one attribute at a time and examine the performance. The range of RTTs, packet sizes and CIRs is chosen according to the simulations in [12]. All the aggregates, A_1-A_{10}, are active. Each simulation lasts 800s. The Average Goodput of each aggregate in one simulation is computed from the 400th second to the 800th second. Each simulation is repeated 10 times, and then a final average is taken over all the runs. In the following, we first present the results for the various cases and then give remarks.

5.1.1. Impact of the number of micro-flows

The number of micro-flows of A_1-A_8 is set to 5, 10, 15, 20, 25, 30, 35 and 15 respectively. Other settings are same as in TABLE II. Fig.6 shows the Average Goodput achieved by A_1-A_7 for each scheme. In the figure, the horizontal line (at 2 Mbps) denotes the target rate to be achieved by each aggregate and 1-7 denote A_1-A_7, respectively.

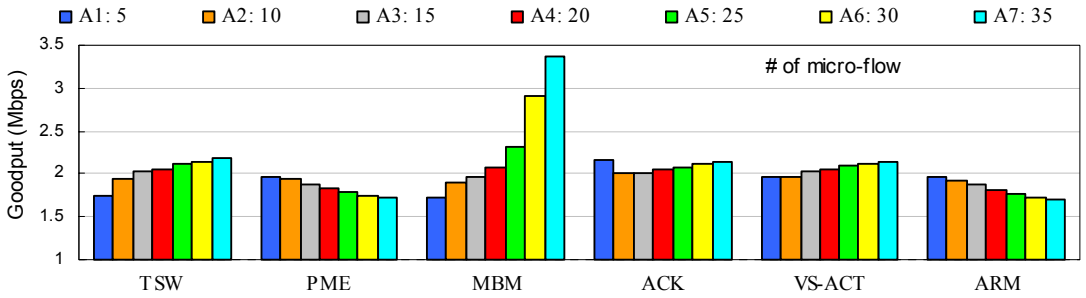


Fig.6. Simulation 5.1.1: Static *under-subscribed* network --- Impact of # of micro-flows

5.1.2. Impact of packet size

The packet sizes of A_1-A_7 are set to 100byes, 300bytes, 500bytes, 700bytes, 1000bytes, 1200bytes, 1500bytes, respectively. Other settings are same as in TABLE II. Fig.7 shows the Average Goodput achieved by A_1-A_7 for each scheme. In the figure, the horizontal line (at 2 Mbps) denotes the target rate to be achieved by each aggregate and 1-7 denote A_1-A_7, respectively. B represents bytes.

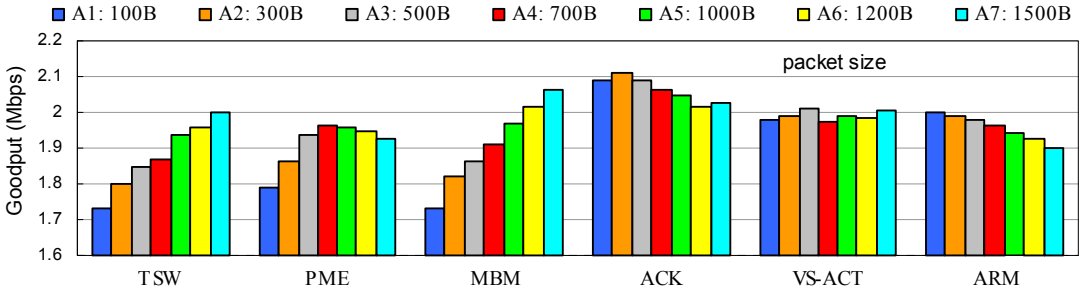


Fig.7. Simulation 5.1.2: Static *under-subscribed* network --- Impact of packet size

5.1.3. Impact of target rate

The target rates of A_1-A_7 are set to 0.5Mbps, 1Mbps, 1.5Mbps, 2Mbps, 2.5Mbps, 3.5Mbps and 4.5Mbps, respectively. So the subscription level is 127.5%. Other settings are same as in TABLE II. Fig.8 shows the Average Goodput Deviation of A_1-A_7 for each scheme. Average Goodput Deviation is defined as [(Average Goodput) – CIR]. Ideally the Average Goodput Deviation should be zero, as represented by the dashed line in the figure.

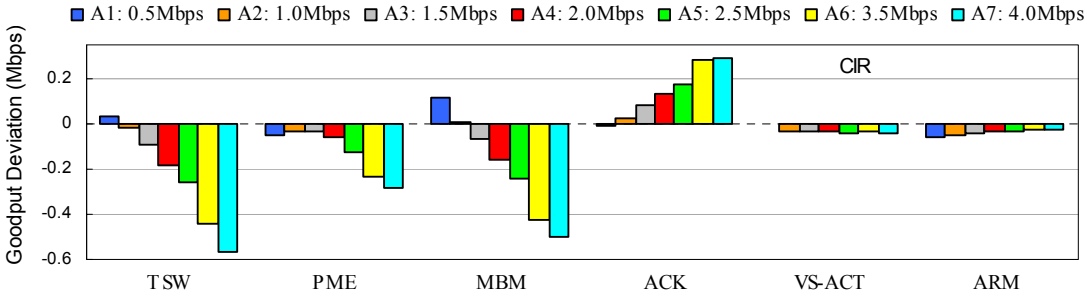


Fig.8. Simulation 5.1.3: Static *under-subscribed* network --- Impact of target rate

5.1.4. *Impact of RTT*

We set RTTs of A_1-A_7 to different values by setting the link delay of E_1—D_i (i from 1 to 7) to 10ms, 50ms, 200ms, 350ms, 500ms, 650ms, and 800ms, respectively. Other settings are same as in TABLE II. Fig.9 shows the Average Goodput achieved by A_1…A_7 for each scheme. In the figure, the horizontal line (at 2 Mbps) denotes the target rate to be achieved by each aggregate and 1-7 denote A_1-A_7, respectively. We repeat the simulations by varying the link delay of S_i—I_i (i from 1 to 7) instead of the link delay of E_1—D_i to set the RTTs of different aggregates to different values. Similar results are obtained. We don't show the results.

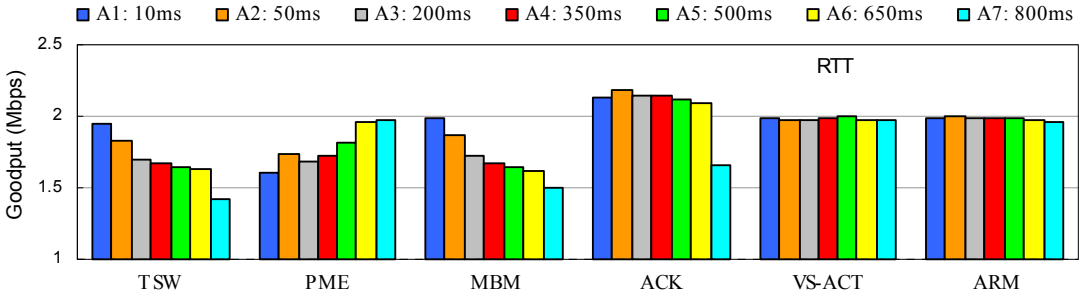
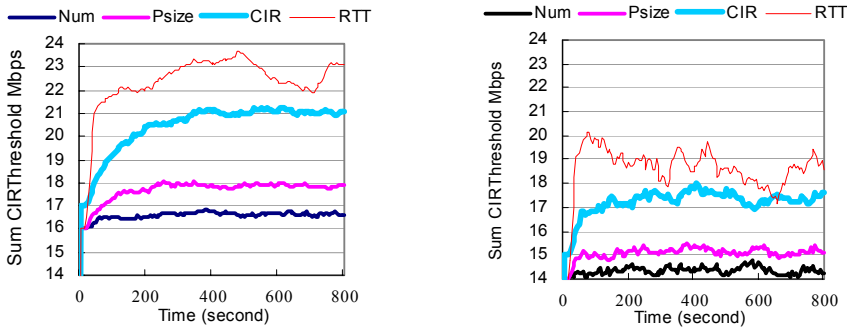


Fig.9. Simulation 0: Static *under-subscribed* network --- Impact of RTT

5.1.5. *Remarks*

The results in Fig.6-Fig.9 show that: (i) When MBM is employed, the Average Goodput of most AF adaptive aggregates in the four experiments can't approximate CIR. Compared to TSW, the Average Goodput of some AF adaptive aggregates is improved, but some is degraded. Consistent with the analysis in Section 3.3.2, there is large difference in the Average Goodput among A_1-A_7. (ii) Compared to TSW and MBM, PME results in smaller difference in the Average Goodput among A_1-A_7, consistent with the analysis in Section 3.3.1. (iii) Due to the fixed control gains, ARM behaves better under some conditions but worse under other conditions. (iv) ACT and VS-ACT perform better than other mechanisms in the four experiments in the term of improving bandwidth assurance. When ACT is applied, A_1-A_7 can achieve their CIRs in Experiments 5.1.1-5.1.3. But in Experiment 0, the Average Goodput of A_7 is far below its CIR while other *conditionally-satisfied* aggregates (A_1-A_6) obtain more than their own CIRs. We use $SumCIR_{Thresh}$ to denote the sum of CIR_{Thresh} of all the aggregates passing through the bottleneck link. Fig.10 (a) and (b) give the $SumCIR_{Thresh}$ variations in the four experiments of ACT scheme and VS-ACT scheme, respectively. Fig.10 (a) explains the performance of ACT in 0, validating the analysis in Section 3.3.4. Fig.10 (b) shows that, when VS-ACT is applied, the $SumCIR_{Thresh}$ of VS-ACT is smaller than that of ACT. Thereby A_7 in Experiment 0 has a greater chance to increase its goodput. The Average Goodput of A_1-A_7 is very close to their CIRs in the four experiments.



(a) ACT (b) VS-ACT

Fig.10. Simulation 5.1: $SumCIR_{Thresh}$

5.2. *Static network*: *exact-subscribed*

Now we investigate the steady-state performance of each scheme in the static *exact-subscribed* network. Only A_1-A_7 are active. The target rates of A_1-A_7 are set to 8.0 Mbps, 4.5 Mbps, 2.5 Mbps, 2.0 Mbps, 1.5 Mbps, 1.0 Mbps, and 0.5Mbps,

respectively. Other settings are the same as in TABLE II. The simulation lasts 800s. Fig.11 shows the Average Goodput Deviation of A_1-A_7 for each scheme. Ideally the Average Goodput Deviation should be zero, as represented by the dashed line in the figure. The Average Goodput at the receiver is computed from the 300ths to the 800ths. Each simulation is repeated 10 times, and then a final average is taken over all the runs. The results of this experiment further confirm the conclusions about TSW, MBM, ACT and VS-ACT made in 5.1. In this experiment, PME and ARM perform better than in Section 5.1. The reason is that the lower bounds of *mp* and CIR_{Thresh} are both 0.0.

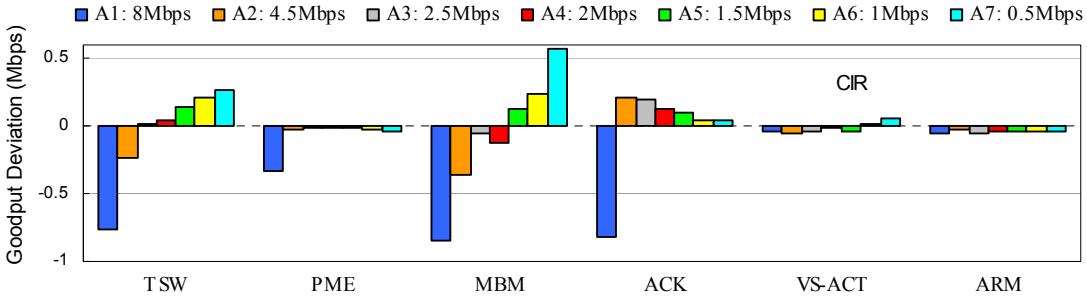


Fig.11. Simulation 5.2: Static *exact-subscribed* network

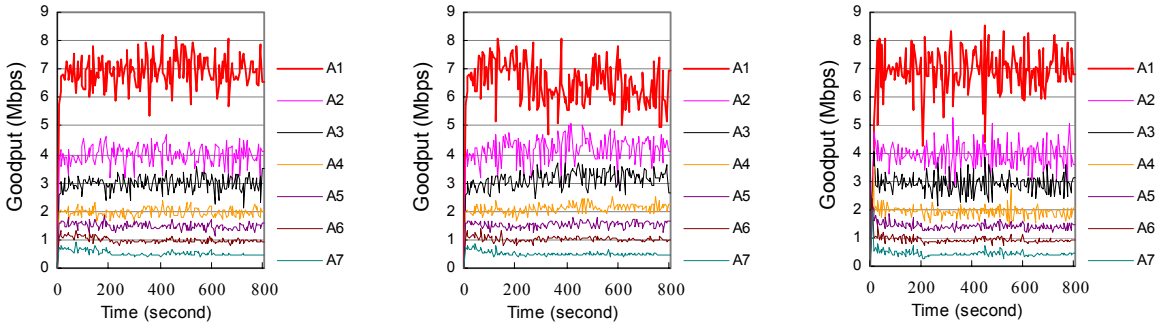
5.3. Dynamic networks

The results in Experiments 5.1 and 5.2 display the failure of TSW, MBM and PME in providing bandwidth assurance in static networks. From this section onwards, we focus on evaluating ACT, ARM and VS-ACT by examining their transient behaviors. We have mentioned earlier that one factor degrading the performance of ACT is the slow decrease in K_p of the *conditionally-satisfied* aggregates with smaller CIR or with larger r_a when r_a >CIR. In Section 5.3.1, we examine the case of "with smaller CIR", that is, the impact of the *conditionally-satisfied* aggregates with smaller CIR on the performance of the *unsatisfied* aggregate. In Section 5.3.2 we examine the case of "larger r_a". We also use the simulations in 5.3.1 and 5.3.2 to show ARM performs better in some network situations but performs worst in other situations.

5.3.1. Varying non-adaptive traffic load

The network is dynamic due to the varying non-adaptive traffic load, which leads to the varying subscription level. The simulation lasts 800s. The sending rates of A_9 and A_{10} are both 0.5 Mbps in [0, 200th]s, 5.0Mbps in [200th, 400th]s, 1.0Mbps in [400th,600th]s and 9.0Mbps in [600th,800th]s, respectively. The target rates of A_1-A_7 are set to 7.0, 4.0, 3.0, 2.0, 1.5, 1.0 and 0.5 Mbps, respectively. A_8, A_9 and A_{10} send best-effort traffic. Other settings are same as in TABLE II. Fig.12 (a), (b) and (c) depict the Average Goodput variation of A_1-A_7 for ACT, VS-ACT and ARM, respectively. They are obtained by measuring the number of packets per 5 seconds. Fig.13 plots CIR_{Thresh} variation and Average Goodput variation of A_1 for ACT and VS-ACT. CIRThresh_A_1_ACT (CIRThresh_A_1_VS-ACT) represents the variation of CIR_{Thresh} of A_1 when ACT (VS-ACT) is applied. GB_A_1_ACT (GB _A_1_VS-ACT) represents Average Goodput variation of A_1 when ACT (VS-ACT) is applied.

The result in Fig.12 (a) shows that, when ACT is employed, the *slowly-decreasing* method damages the benefit of A_1 when the network is changing from a heavy *implicit over-subscribed* situation [200th,400th]s to a light *implicit over-subscribed* situation [400,600]s. In the whole simulation, A_2-A_7 can achieve their CIRs; but the goodput of A_1 can reach its CIR only in [0, 200th]s and [500th,600th]s. The reason is that in [200th,400th]s excessive IN traffic in the network increases the probability of ECN-marking or dropping of the IN packets. Therefore, no matter how the CIR_{Thresh} of A_1 is increased, when the sending rate of A_1 (all are IN packets) increases past a certain point, some IN packets of A_1 are ECN-marked or dropped. As a consequence, the Average Goodput of A_1 can't reach its CIR. Fig.13 shows this. In [400th,600th]s, the sending rates of A_9 and A_{10} are small. Actually, A_1-A_7 can achieve their CIRs without using such a large CIR_{Thresh} as in the previous periods. But Fig.12 (a) shows that, in [400th, 500th]s, the Average Goodput of A_1 is less than its CIR. This is because, during this period, A_2-A_7 slowly decrease the large value of their CIR_{Thresh}, which is accumulated in [200th, 400th]s. Thus, there is still excessive IN traffic entering the network, preventing A_1 from increasing its goodput. A_1 in [600th, 800th]s behaves as in [200th,400th]s.



(a). ACT (b). VS-ACT (c). ARM

Fig.12. Simulation 5.3.1: Average Goodput variation of A_1---A_7

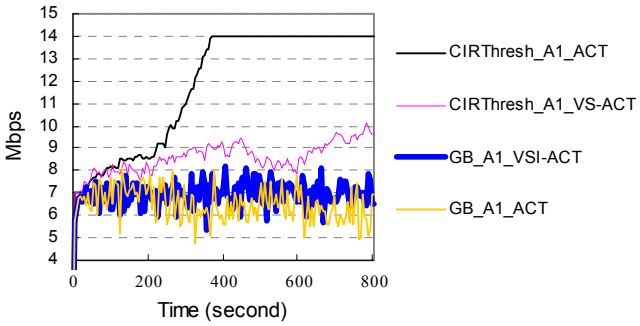


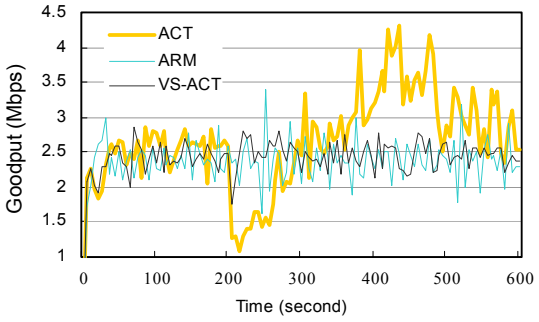
Fig.13. Simulation 5.3.1: Goodput and CIR_{Thresh} variations of A_1

Fig.12 (b) and (c) show that when VS-ACT and ARM are applied, the transient behavior is quite satisfactory in terms of the small settling time and the small overshoots. The Average Goodput of A_1-A_7 approximates their corresponding CIRs during the entire simulation.

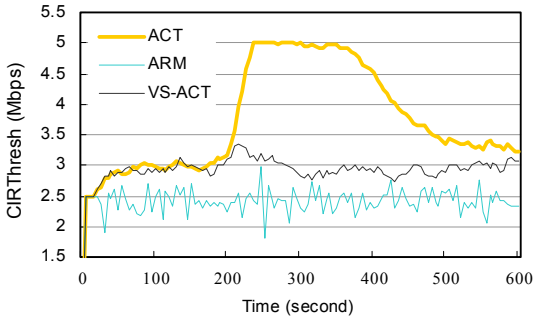
5.3.2. Varying number of micro-flows in aggregates

In this experiment, we examine the case of "larger r_a". We vary the number of micro-flows in the aggregates in order to give them different abilities in grabbing bandwidths. The number of micro-flows of A_1-A_8 is set to 5, 10, 15, 20, 25, 30, 35 and 15, respectively. All the aggregates are active. In order to avoid the impact of CIR, the CIRs of A_1-A_7 are all set to 2.5Mbps. The CIR of A_9 is 1.5Mbps. The sending rates of A_9 and A_{10} are both 5Mbps. Other settings are same as in TABLE II. The simulation lasts 600 seconds. In the first 200 seconds, only 5 micro-flows in each adaptive aggregate of A_1-A_7 are active. From the 200ths to 400ths, all micro-flows are active. In the last 200s, only 5 micro-flows in each adaptive aggregate are active. Fig.14 shows the Average Goodput and CIR_{Thresh} variations of A_1 and A_7 for ACT, VS-ACT and ARM.

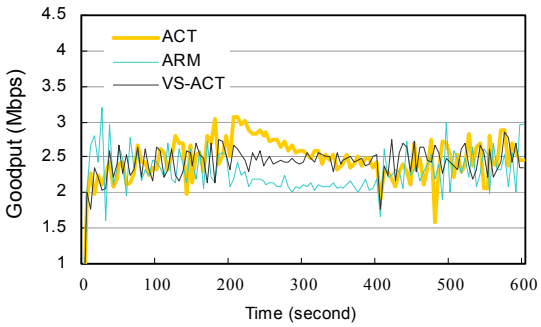
We can see that: (i) when ACT is applied, during the first 200 seconds, A_1 and A_7 have the same characteristics and the CIR_{Thresh} of both aggregates is approximately the same. At the 200th second, a number of micro-flows start. Although CIR_{Thresh} of A_1 is increased quickly (because γ =0.05), due to the slow decrease in CIR_{Thresh} of other aggressive *conditionally*-satisfied aggregates such as A_7, the goodput of A_1 is far below its CIR. This continues until the CIR_{Thresh} of A_7 is decreased sufficiently at about the 350th second. At the 400th second, most micro-flows stop and A_1---A_7 have the same traffic characteristics again. Thus, A_1 can achieve its CIR without using so large CIR_{Thresh} as in previous period. But the *slowly-decreasing* method in ACT makes CIR_{Thresh} of A_1 decrease very slowly, delaying other aggregates such as A_7 from improving bandwidth assurance. (ii) When VS-ACT is applied, it shows fast response to network changes and there is small variation in the Average Goodput of each adaptive AF aggregate in the whole simulations. (iii) Fig.14 (c) shows the weird performance of ARM during [200th, 400th]s. This is due to the fixed controller gains.



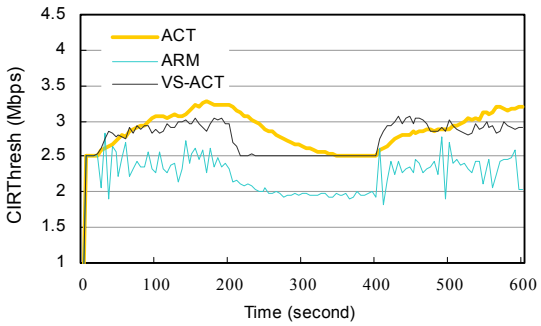
(a). A_1: Goodput



(b). A_1: CIR_{Thresh}



(e). A_7: Goodput

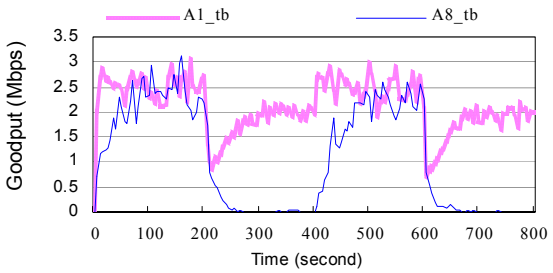


(f). A_7: CIR_{Thresh}

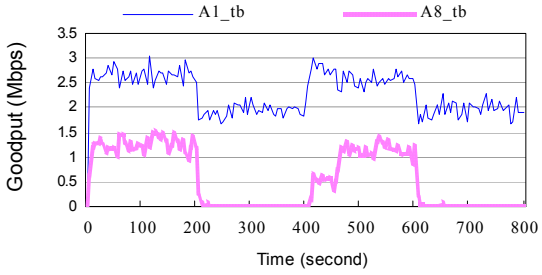
Fig.14. Simulation 5.3.2: Varying number of micro-flows in aggregates

5.4. The importance of upper bound and lower bound on CIR_{Thresh}

This experiment aims to investigate the importance of using upper bound and lower bound on CIR_{Thresh}, mentioned in Section 4.2.2. We do simulations only with VS-ACT. A_1-A_7 have the same traffic characteristics as in TABLE II. The simulation lasts 800s. The sending rates of A_9 and A_{10} are varying during the simulation, both 0.0 Mbps in [0, 200th]s, 9.0Mbps in [200th, 400th]s, 1.0Mbps in [400th,600th]s and 9.0Mbps in [600th,800th]s. We do two simulations and then compare their results. In the first simulation, during the first 400s CIR_{Thresh} varies in [0.0, 2CIR]; during the left 400s CIR_{Thresh} varies in [0.0, ∞]. In the second simulation CIR_{Thresh} is allowed to vary in [CIR, 2CIR] during the entire simulation. We use the results in the first 400s to illustrate the importance of the lower bound and the results in the left 400s to investigate the importance of the upper bound. We only show the results of A_1, A_8, A_9, and A_{10}. Fig.15 (a) and (b) plot the Average Goodput variation of A_1 and A_8 in the two simulations, respectively. Fig.16 (a) and (b) plot the Average Goodput variation of A_9 and A_{10} in the two simulations, respectively.



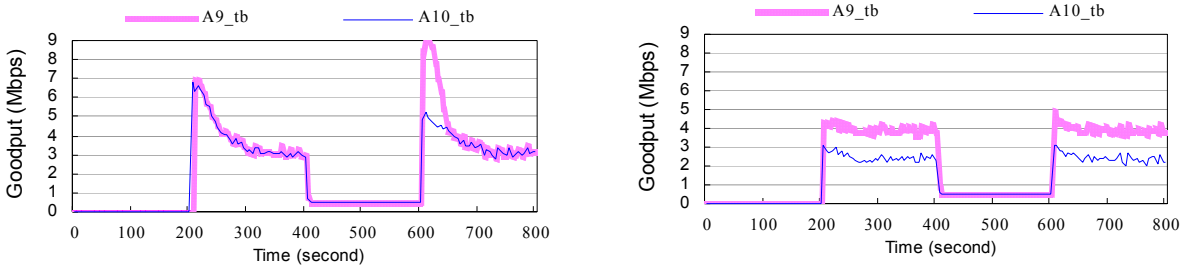
(a) First simulation: no bound



(b) Second simulation: bounded

Fig.15. Simulation 5.4: Performance of A_1 and A_8

Fig.15 (a) shows that in [80th,200th]s and [500th, 600th]s, the Average Goodput of A_1 approximates the Average Goodput of A_8. Fig.15 (b) shows that when CIR_{Thresh} is bounded, the fairness in sharing excess bandwidth between A_1 and A_8 is improved greatly. In addition, the results in [200th, 400th]s and [600th,800th]s show, when bounded, the bandwidth assurance of A_1 is achieved. Same conclusions can be made about non-adaptive aggregates A_9 and A_{10} from the results in Fig.16.



(a) First simulation: no bound

(b) Second simulation: bounded

Fig.16. Simulation 5.4: Performance of A_9 and A_{10}

6. Discussion and Conclusion

In this paper, we systematically explore the application of feedback control theory to design mechanisms to improve bandwidth assurance based only on the knowledge gathered at ingress routers. We use a control theoretic approach to analyze some existing adaptive mechanisms in the literature. Then, a Variable-Structure PI controller for adapting CIR Threshold is developed. The performance evaluation results support the conclusions derived from our control-theoretic analysis of the existing algorithms and demonstrate the superiority of VS-ACT over a wide range of network dynamics.

As in the case of other ingress-based mechanisms that use only *local knowledge* to improve bandwidth assurance, VS-ACT also faces the problem of low domain throughput [22] when there exist aggressive non-adaptive flows. This problem can be alleviated by combining it with the mechanism developed in [22].

Note that the ingress-based mechanisms run at the output queue at the ingress routers. All the above discussions in the *under-subscribed* networks assume that the arriving rate of an aggregate at the ingress input link card is equal to the departure rate of this aggregate from the ingress output link card. But this may not be true when switches, such as CIOQ switches, have multiple input and output queues. In such switches the existence of cross traffic between multiple input and output interfaces may cause the difference between the arriving rate and the departure rate of an aggregate and then affect the attainment of bandwidth assurance. When the failure of bandwidth assurance is caused by only cross-traffic, increasing CIR_{Thresh} contributes nothing to the attainment of bandwidth assurance. The authors in [34] propose a solution to prevent the failure of bandwidth assurance caused by cross-traffic. This solution and VS-ACT are complementary and can be used in conjunction with each other. Note that when the failure of bandwidth assurance of an aggregate is caused by cross-traffic inside the switch, increasing CIR_{Thresh} of this aggregate does not lead to serious performance degradation to those other flows that share other switches with this aggregate in the networks. By serious performance degradation, we mean that undesired increase of CIR_{Thresh} may cause transient performance degradation to other flows when there is no impact from cross-traffic, but the ingress-based mechanisms can quickly correct the undesired increase.

7. Acknowledgments

The authors thank anonymous referees for their valuable comments, which have been, in particular, helpful in better explaining the contributions of this work.

Appendix Local stability analysis of the *under-subscribed* AF-based DiffServ network

A.1 System stability analysis

This section presents the results of the local stability analysis of the *under-subscribed* AF-based Diffserv network where (i) there are n heterogeneous aggregates, each consisting of N_i identical long-lived TCP connections; (ii) the ingress router uses the TSW profiler to provide two-level edge coloring and uses a PI-type marker with fixed-gains to adjust the marking threshold CIR_i^{Th}; (iii) RIO is used as AQM at the core router with an infinite and non-emptying buffer. Without loss of generality, we assume that each aggregate is served by a separate ingress router. The traffic of all aggregates feed into a

core router with link capacity C and queue length denoted by $q(t)$. Our starting point is the linearized model for the standard AF-based DiffServ network and the method of analyzing stability proposed in [31]. For simplicity, we assume the dropping probability of IN traffic at congested routers is zero. Before continuing, we first introduce the notations that are used in the following. The subscript i refers to the i-th aggregate, from 1 to n.

- C: *link capacity (packets/sec).*
- N_i: *the number of micro-flows in the* i*-th aggregate.*
- R_i: *the round-trip delay of a micro-flow in the* i*-th aggregate (second).*
- p_r: *dropping/marking probability of red traffic.*
- q: *instantaneous queue length (packets).*
- W_i: *window size of a micro-flow in the* i*-th aggregate (packet).*
- T_{pi}: is the average propagation delay of the i-th aggregate.
- x_i : the sending rate of the i-th aggregate.
- CIR_i^{Th} : the marking threshold of the i-th aggregate.
- $|\bullet|$: the magnitude of $\bullet$.

A linearized model of the *under-subscribed* AF-based DiffServ network around the equilibrium point $\left(q^e, W_i^e, p_r^e, (CIR_i^{Th})^e\right)$ is described by

$$\begin{cases} \delta W_i(s) = \dfrac{\dfrac{\partial g_i}{\partial CIR_i^{Th}}}{s - \dfrac{\partial g_i}{\partial W_i}} \delta CIR_i^{Th}(s) + \dfrac{\dfrac{\partial g_i}{\partial p_r}}{s - \dfrac{\partial g_i}{\partial W_i}} e^{-sR_i} \delta p_r(s) \\ \delta q(s) = \displaystyle\sum_{i=1}^{n} \dfrac{\dfrac{\partial f}{\partial W_i}}{s - \dfrac{\partial f}{\partial q}} \delta W_i(s) \end{cases} \tag{A.1}$$

where

$$\begin{cases} \dot{W}_i(t) = g_i(q, W_i, p_r, CIR_i^{Th}) \\ \dot{q}(t) = f(q, W_i, p_r, CIR_i^{Th}) \\ \dfrac{\partial f}{\partial q} = -\displaystyle\sum_{i=1}^{n} \dfrac{x_i}{C \times R_i} \\ \dfrac{\partial f}{\partial W_i} = \dfrac{N_i}{R_i} \\ \dfrac{\partial g_i}{\partial W_i} = \left(\dfrac{CIR_i^{Th}}{2N_i} - \dfrac{CIR_i^{Th}}{N_i W_i^2} \right) p_r - \dfrac{W_i}{R_i} p_r \\ \dfrac{\partial g_i}{\partial p_r} = -\dfrac{1}{R_i} + \dfrac{CIR_i^{Th}}{N_i W_i} + \dfrac{W_i CIR_i^{Th}}{2N_i} - \dfrac{W_i^2}{2R_i} \\ \dfrac{\partial g_i}{\partial CIR_i^{Th}} = \dfrac{1}{N_i} \left(\dfrac{1}{W_i} + \dfrac{W_i}{2} \right) p_r \\ \delta W_i(t) = W_i(t) - W_i^e \\ \delta q(t) = q(t) - q^e \\ \delta p_r(t) = p_r(t) - p_r^e \\ \delta CIR_i^{Th}(t) = CIR_i^{Th}(t) - \left(CIR_i^{Th} \right)^e \end{cases}$$

The details of the model can be found in [33]. The equilibrium point $\left(q^e, W_i^e, p_r^e, (CIR_i^{Th})^e\right)$ satisfies the following equations

$$\begin{cases} 0 = 1 - \left(1 - \dfrac{CIR_{\mathrm{i}}^{\mathrm{Th}}}{x_{\mathrm{i}}}\right) p_{\mathrm{r}} - 0.5\left(1 - \dfrac{CIR_{\mathrm{i}}^{\mathrm{Th}}}{x_{\mathrm{i}}}\right) p_{\mathrm{r}} W_{\mathrm{i}}^2 \\ R_{\mathrm{i}}(t) \triangleq T_{\mathrm{pi}} + \dfrac{q}{C} \\ 0 = \sum_{i=1}^{n}\left(N_{\mathrm{i}} \dfrac{W_{\mathrm{i}}}{R_{\mathrm{i}}}\right) - C \end{cases} \tag{A.2}$$

PI-ACT$_{\mathrm{j}}$ defined in Eq.(A.3) is employed at the *j*-th ingress router to adjust the marking threshold $CIR_{\mathrm{j}}^{\mathrm{Th}}$ of the *j*-th unsatisfied AF adaptive aggregate, *j*=1..*m*. *m* is the number of unsatisfied AF adaptive aggregates.

$$\mathrm{PI\text{-}ACT}_{\mathrm{j}}(s) = \frac{k_{\mathrm{PI\text{-}ACT_j}}\left(\dfrac{s}{z_{\mathrm{PI\text{-}ACT\ j}}} + 1\right)}{s} \tag{A.3}$$

The aggregate arriving rate at the ingress router is computed by measuring the number of sent packets over a fixed time period T_{TSW} and further smoothed by a low-pass filter *F*. The transfer function representing this estimation is given by

$$F(s) = \frac{a}{s+a} e^{-sT_{TSW}} \tag{A.4}$$

The transfer function representing RED mechanism for OUT traffic is given by

$$AQM_{\mathrm{r}}(s) = \frac{L_{\mathrm{RED}}}{\dfrac{s}{k_{\mathrm{RED}}} + 1} \tag{A.5}$$

Combining the model in Eq.(A.1) with RIO and PI-ACTs leads to a closed-loop system. The details for stability analysis of this system are given in [33]. In the following we first give the ***Small Gain*** **Theorem** applied for stability analysis and then give the conditions for system stability.

Small Gain **Theorem [35]**: *Consider the feedback system shown in* Fig.17, where $\hat{P}$ and Δ are stable linear systems. *If* $\left|\hat{P}\Delta\right| < 1$ for all ω *then the feedback system is stable.*

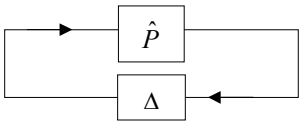


Fig.17. Simple feedback system

In the AF-based DiffServ network described by Eq.(A.1) and Eq.(A.3)-Eq.(A.5), $\hat{P}(s) \triangleq \dfrac{\delta p}{\delta \tilde{x}} = \dfrac{\dfrac{1}{s - \dfrac{\delta f}{\delta q}}\mathrm{AQM}}{1 - L_{\hat{P}}(s)}$ where

$$L_{\hat{P}}(s) = \frac{1}{s - \dfrac{\partial f}{\partial q}} \frac{L_{\mathrm{RED}}}{\dfrac{s}{k_{\mathrm{RED}}} + 1} \sum_{i=1}^{n}\left(e^{-sR_{\mathrm{i}}} \frac{\partial g}{\partial p} \frac{1}{s - \dfrac{\partial g_{\mathrm{i}}}{\partial W_{\mathrm{i}}}} \frac{N_{\mathrm{i}}}{R_{\mathrm{i}}}\right) = L_{\mathrm{RED}} \tilde{L}_{\hat{P}}(j\omega) \tag{A.6}$$

and $\Delta(s) = \sum_{j \in J} P_j \Delta_j(s)$ where

$$\begin{cases} \Delta_{\mathrm{j}}(s)=\dfrac{L_{\Delta_{\mathrm{j}}}(s)}{1+L_{\Delta_{\mathrm{j}}}(s)} \\ L_{\Delta_{\mathrm{j}}}(s)=\dfrac{1}{s-\dfrac{\partial g_{\mathrm{j}}}{\partial W_{\mathrm{j}}}}\dfrac{N_{\mathrm{j}}}{R_{\mathrm{j}}}\dfrac{\partial g_{\mathrm{j}}}{\partial CIR_{\mathrm{j}}^{\mathrm{Th}}}\dfrac{k_{\text{PI-ACT}_{\mathrm{j}}}\left(\dfrac{s}{z_{\text{PI-ACT}_{\mathrm{j}}}}+1\right)}{s}\dfrac{a_{\mathrm{j}}}{s+a_{\mathrm{j}}}e^{-sT_{TSW_{\mathrm{j}}}}=k_{\text{PI-ACT}_{\mathrm{j}}}\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega) \\ P_{\mathrm{j}}=e^{-sR_{\mathrm{j}}}\dfrac{\partial g_{\mathrm{j}}}{\partial p_r}\dfrac{1}{s-\dfrac{\partial g_{\mathrm{j}}}{\partial W_{\mathrm{j}}}}\dfrac{N_{\mathrm{j}}}{R_{\mathrm{j}}} \end{cases} \tag{A.7}$$

In Eq. (A.7), Δ_{j} denotes the perturbation induced by PI-ACT$_{\mathrm{j}}$ at the j-th aggregate and P_{j} denotes the plant of the j-th aggregate.

We can prove that the AF-based DiffServ network described by Eq.(A.1) and Eq.(A.3)-Eq.(A.5) is locally stable if L_{RED} and $k_{\text{PI-ACT}_{\mathrm{j}}}$ satisfies

$$\begin{cases} k_{\text{VS-ACT}_{\mathrm{j}}}<\min\left\{\dfrac{1}{2\left|\mathrm{Re}\left(\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega_2)\right)\right|},\dfrac{1}{\left|\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega_1)\right|}\right\},\ j=1,\ldots,m \\ L_{\mathrm{RED}}<\min\left\{\dfrac{\varepsilon_1}{M},\dfrac{1}{\tilde{M}},\dfrac{1}{\left|\tilde{L}_{\hat{P}}(j\omega_3)\right|}\right\} \end{cases} \tag{A.8}$$

More details about Eq.(A.8) are given in [33]. Some notations in Eq.(A.8) are explained next. ω_1 is the frequency such that $\left|\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega_1)\right|=\max_{\omega\in\Omega_1}\left|\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega)\right|$, where $\Omega_1=\left\{\omega:\angle\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega)=-180^\circ\right\}$. ω_2 is the frequency such that $\mathrm{Re}\left(\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega_2)\right)=\min_{\omega\in\Gamma}\mathrm{Re}\left(\tilde{L}_{\Delta_{\mathrm{j}}}(j\omega)\right)$ where Γ is Nyquist contour of $\tilde{L}_{\Delta_{\mathrm{j}}}(s)$. ω_3 is the frequency such that $\left|\tilde{L}_{\hat{P}}(j\omega_3)\right|=\max_{\omega\in\Omega_2}\left|\tilde{L}_{\hat{P}}(j\omega)\right|$. $M=\left|\dfrac{1}{\dfrac{\partial f}{\partial q}}\right|\sum_{i=1}^{n}\left|\dfrac{\partial g_{\mathrm{i}}}{\partial p}\right|\left|\dfrac{1}{\dfrac{\partial g_{\mathrm{i}}}{\partial W_{\mathrm{i}}}}\right|\left|\dfrac{N_{\mathrm{i}}}{R_{\mathrm{i}}}\right|$ and $\tilde{M}=\dfrac{m\max_{j\in J}\left\{\left|\dfrac{\partial g_{\mathrm{j}}}{\partial p}\dfrac{N_{\mathrm{j}}}{R_{\mathrm{j}}}\right|\right\}}{\min_{j\in J}\left\{\left|-\dfrac{\partial g_{\mathrm{j}}}{\partial W_{\mathrm{j}}}\right|\right\}\left|\dfrac{\partial f}{\partial q}(1-\varepsilon_1)\right|}$. Here $\varepsilon_1\in(0,1.0)$.

A.2 An illustrative example

In this subsection, we apply the above sufficient conditions to analyze stability of a simple *under-subscribed* AF-based DiffServ network. This network consists of three heterogeneous aggregates (A_1-A_3). A_1-A_3 consist of 20, 30 and 25micr-flows, respectively. All the micro-flows (FTP flows) in an aggregate have the same characteristics. The round-trip link delays of A_1, A_2 and A_3 are set to 0.23second, 0.1second and 0.05second, respectively. CIR_1=2000packets, CIR_2=500packets, CIR_3=1250packets. Core router buffer size is set to 1200packets. Link capacity is 4500packets. Thus only PI-ACT$_1$ is active.

We set [q_{min}, q_{max}, p_{max}, q_weight] for OUT packets to [600packets, 50packets, 0.25, 0.0000011111]. Thus the AQM controller used for OUT traffic at the core router is

$$AQM_{\mathrm{r}}(s)=\frac{4.5455\times10^{-4}}{\dfrac{s}{0.005}+1} \tag{A.9}$$

We choose

$$\text{PI-ACT}_1(s)=\frac{0.0006\left(\dfrac{s}{0.6}+1\right)}{s}\qquad\text{and}\qquad F(s)=\frac{1}{s+1}e^{-s} \tag{A.10}$$

That is, the arriving rate of an aggregate at the ingress node is estimated per one second. In PI-ACT, the proportional gain is 0.001 and the integral gain is 0.0006.

Queue length oscillates around 100packets. Hence, the round trip times are R_1 =0.2522s, R_2 =0.1222s, R_3=0.0722s. The nominal TCP/AQM system is described by

$$\left|\hat{P}(j\omega)\right| = \left| \frac{\dfrac{1}{j\omega - \dfrac{\partial f}{\partial q}} AQM}{1 - \dfrac{1}{j\omega - \dfrac{\partial f}{\partial q}} AQM \left(\sum_{i=1}^{3} P_i\right)} \right| \tag{A.11}$$

where $\dfrac{\partial f}{\partial q}$ = -8.1930 1/second; and where the transfer functions for A_1, A_2 and A_3 are described by

$$\begin{aligned}
P_1 &= e^{-sR_1} \frac{\partial g_1}{\partial p_r} \frac{1}{s - \dfrac{\partial g_1}{\partial W_1}} \frac{N_1}{R_1} = e^{-s(0.2522)} (-223.0209) \frac{1}{s-(-0.9733)} \frac{20}{0.2522} \\
P_2 &= e^{-sR_2} \frac{\partial g_2}{\partial p_r} \frac{1}{s - \dfrac{\partial g_2}{\partial W_2}} \frac{N_2}{R_2} = e^{-s(0.1222)} (-73.6498) \frac{1}{s-(-1.3460)} \frac{30}{0.1222} \\
P_3 &= e^{-sR_3} \frac{\partial g_3}{\partial p_r} \frac{1}{s - \dfrac{\partial g_3}{\partial W_3}} \frac{N_3}{R_3} = e^{-s(0.0722)} (-186.9806) \frac{1}{s-(-2.8477)} \frac{25}{0.0722}
\end{aligned} \tag{A.12}$$

The disturbance is described by $\Delta_1(s) = \dfrac{L_{\Delta_1}(s)}{1 + L_{\Delta_1}(s)}$, where

$$\tilde{L}_{\Delta_j}(s) = \frac{1}{s - \dfrac{\partial g_1}{\partial W_1}} \frac{N_1}{R_1} \frac{\partial g_1}{\partial CIR_1^{\text{Th}}} \left(\text{PI-ACT}_1\right) F_1(s) = \frac{1}{s-(-0.9733)} \frac{20}{0.2522} (0.0186) \frac{6\times 10^{-4}\left(\dfrac{s}{0.6}+1\right)}{s} \frac{1}{s+1} e^{-s} \tag{A.13}$$

We observe in Fig.18 that $\left|\left(\hat{P}\right)\left(P_1\Delta_1\right)\right| < 1$, in Fig.19 that $\left|\Delta_1\right| < 1$, and in Fig.20 that $\left|\hat{P}\right| < 1$, which establish local stability of the example network. In addition, we give the *ns-2* simulation results. Fig.21 shows the Average Goodput variation and CIR_{Thresh} Variations of A_1---A_3. Fig.22 gives the bottleneck link queue length variation versus time.

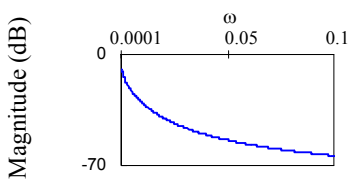


Fig.18. Magnitude Bode plot of ($\hat{P}\,\Delta$)

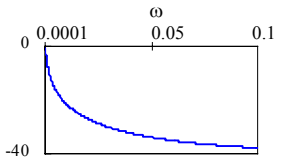


Fig.19. Magnitude Bode plot of Δ

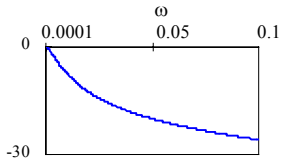


Fig.20. Magnitude Bode plot of $\hat{P}$

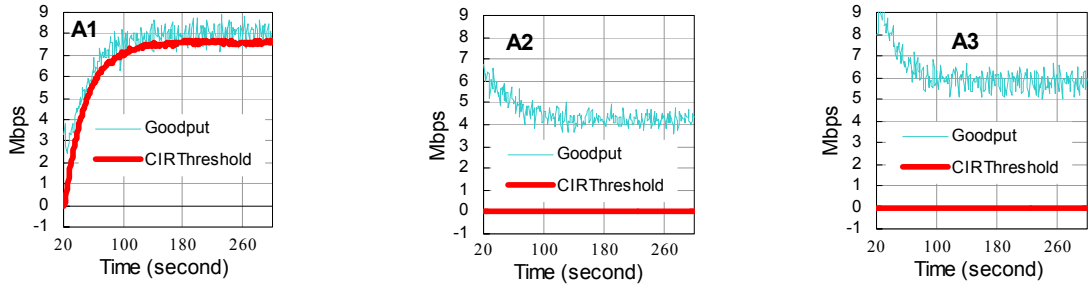


Fig.21. Goodput and CIR_{Thresh} variations of A_1-A_3

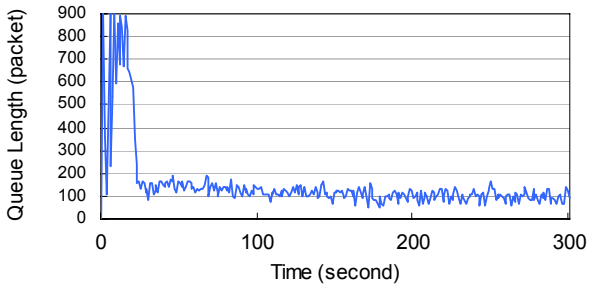


Fig.22. Bottleneck link queue length variation

A.3 Relations between network parameters and $k_{\text{VS-ACT}_j}$

We now analyze the relations between network parameters (N_j and R_j) and $\left|\tilde{L}_{\Delta_j}(s)\right|$, and the relations between network parameters (N_j and R_j) and $\left|\text{Re}\left(\tilde{L}_{\Delta_j}(s)\right)\right|$. When Trapezoidal rule is applied, $K_p=\frac{k_{\text{PI-ACT}_j}}{z_{\text{PI-ACT}_j}}+k_{\text{PI-ACT}_j}\frac{T}{2}$ and $K_i=\frac{k_{\text{PI-ACT}_j}}{z_{\text{PI-ACT}_j}}-k_{\text{PI-ACT}_j}\frac{T}{2}$. Here T is sampling interval. Letting $z_{\text{PI-ACT}_j}=-\frac{\partial g_j}{\partial W_j}$, we obtain

$$\frac{1}{\left|\tilde{L}_{\Delta_j}(j\omega_1)\right|}=\left|\frac{W_j-\left(\frac{R_j CIR_j^{\text{Th}}}{N_j}\left(\frac{1}{2}-\frac{1}{W_i^2}\right)\right)}{\left(\frac{1}{W_j}+\frac{W_j}{2}\right)}\right|\left|\frac{j\omega}{1}\frac{j\omega+a_j}{a_j}\frac{1}{e^{-sT_{\text{TSW}_j}}}\right| \tag{A.14}$$

and

$$\left|\text{Re}\left(\tilde{L}_{\Delta_j}(j\omega_2)\right)\right|=\left|\frac{1}{W_j-\frac{R_j CIR_j^{\text{Th}}}{N_j}\left(\frac{1}{2}-\frac{1}{W_j^2}\right)}\left(\frac{1}{W_j}+\frac{W_j}{2}\right)\frac{1}{\frac{1}{a_j^2}+\omega_2^2}(\cos(T_{\text{TSW}_j})\frac{1}{a_j}-\omega\sin(T_{\text{TSW}_j}))\right| \tag{A.15}$$

Thus, the upper bound of $\frac{1}{\left|\tilde{L}_{\Delta_j}(s)\right|}$ is decreasing function of CIR_i^{Th} and R_i, increasing function of N_i. Same relations are for $\left|\mathrm{Re}\left(\tilde{L}_{\Delta_j}(s)\right)\right|$.

8. References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC2475, December 1998.

[2] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," IETF RFC 2597, June 1999.

[3] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu, "On Achievable Service Differentiation with Token Bucket Marking for TCP," In ACM. Performance Evaluation Review, vol.28, no.1, pp.23-33, June 2000.

[4] N. Seddigh, B. Nandy, and P. Pieda, "Using TCP Models to Understand Bandwidth Assurance in a Differentiated Services Network," In Proc. IEEE Global Telecommunications Conference (GLOBECOM 2001), vol. 3, pp. 1800-1805, November 2001.

[5] N. Christin, J. Liebeherr, and Tarek Abdelzaher, "A Quantitative Assured Forwarding Service," In Proc. IEEE Conference on Computer Communications (INFOCOM 2002), vol. 2, pp. 864-873, June 2002.

[6] W. Fang, N. Seddigh, and B. Nandy, "A Time Sliding Window Three Color Marker (tswtcm)," IETF RFC 2859, June 2000.

[7] T. Abdelzaher, J. Stankovic, C. Lu, R. Zhang, and Y. Lu, "Feedback Performance Control in Software Systems," In Proc. IEEE Control Systems, vol. 23, no. 3, pp. 74-90, June 2003.

[8] D. Clark, and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," In IEEE/ACM Transactions on Networking, vol. 6, no. 4, pp. 362-373, August 1998.

[9] M. Goyal, A. Durresi, P. Misra, C. Liu, and Raj Jain, "Effect of Number of Drop Precedences in Assured Forwarding," In Proc. IEEE Global Telecommunications Conference (GLOBECOM 1999), vol. 1a, pp. 188-193, December 1999.

[10] B. Nandy, N. Seddigh, P. Pieda, and J. Ethridge, "Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks," In Proc. IFIP Conference on High Performance Networking (HPN 2000), May 2000.

[11] A. Habib, B. Bhargava, and S. Fahmy, "A Round Trip Time and Time-out Aware Traffic Conditioner for Differentiated Services Networks," In Proc. IEEE International Conference on Communications (ICC 2002), vol. 2, pp. 981-985, April 2002.

[12] M. A. El-Gendy, and K. Shin, "Equation-Based Packet Marking for Assured Forwarding Services," In Proc. IEEE Conference on Computer Communications (INFOCOM 2002), vol. 2, pp. 845-854, June 2002.

[13] W. Lin, R. Zheng, and J.C. Hou, "How to make assured service more assured," In Proc. IEEE International Conference on Network Protocols (ICNP 1999), pp. 182-191, November 1999

[14] K.R.R. Kumar, A.L. Ananda, and L. Jacob, "A Memory-Based Approach for a TCP-Friendly Traffic Conditioner in DiffServ Networks," In Proc. IEEE International Conference on Network Protocols (ICNP 2001), pp. 138-145, November 2001.

[15] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated Services Internet," In IEEE/ACM Transactions on Networking, vol. 7, no. 5, pp. 685-697, October 1999.

[16] X. Chang, and J. K. Muppala, "Adaptive Marking Threshold for Improving Performance of Assured Forwarding Services," In Proc. IEEE Global Telecommunications Conference (GLOBECOM 2003), vol. 6, pp. 3073-3077, December 2003.

[17] Y. Chait, C.V. Hollot, V. Misra, D. Towsley, H. Zhang, and J.C.S. Lui, "Providing Throughput Differentiation for TCP Flows Using Adaptive Two-color Marking and Two-level AQM," In Proc. IEEE Conference on Computer Communications (INFOCOM 2002), vol. 2, pp. 837-844, June 2002.

[18] P. Siripongwutikorn, S. Banerjee, and D. Tipper, "A Survey of Adaptive Bandwidth Control Algorithms," In IEEE Communications Surveys, vol. 5, no. 1, Third Quarter 2003.

[19] K.R.R Kumar, A.L.Ananda, and L. Jacob, "Using Edge-To-Edge Feedback Control to make Assured Service More Assured in DiffServ Networks," In Proc. IEEE Local Computer Networks (LCN 2001), pp. 160-167, November 2001.

[20] B. Nandy, J. Ethridge, A. Lakas, and A. Chapman, "Aggregate Flow Control: Improving Assurances for Differentiated Services Network," In Proc. IEEE Conference on Computer Communications (INFOCOM 2001, vol. 3, pp. 1340-1349, April 2001.

[21] D. Harrison, Y. Xia, S. Kalyanaraman, and A. Venkatesan, "An Accumulation-based, Closed-loop Scheme for Expected Minimum Rate and Weighted Rate Services," In Journal of Computer Networks, vol.45, no.6, pp. 801-818, 2004.

[22] X. Chang, and J. K. Muppala, "Adaptive Marking Threshold for Assured Forwarding Services," In Proc. IEEE International Conference on Computer Communications and Networks (ICCCN 2003), pp. 325-330, October 2003.

[23] E. C. Park, and C. H. Choi, "Adaptive Token Bucket Algorithm for Fair Bandwidth Allocation in DiffServ Networks", In Proc. IEEE Global Telecommunications Conference (GLOBECOM 2003), vol. 6, pp. 3176-3180, December 2003.

[24] C.V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," In Proc. IEEE Conference on Computer Communications (INFOCOM 2001), vol. 3, pp. 1726-1734, April 2001.

[25] B. Armstrong, and B.A. Wade, "Nonlinear PID control with partial state knowledge: design by quadratic programming," In Proc. IEEE American Control Conference (ACC 2000), vol. 2, pp. 774-778, June 2000.

[26] C. L. Phillips, and H. T. Nagle, "Digital control system analysis and design," Englewood Cliffs, N.J.: Prentice Hall, 1995, 3rd edition.

[27] K.J. Åström, and T. Hgglund, "PID Controllers: Theory, Design, and Tuning," 2nd, Instrument Society of America, Research Triangle Park, NC, 1995.

[28] B.C. Li, and K. Nahrstedt. "A Control-based Middleware Framework for Quality of Service Adaptations," In IEEE Journal on Selected Areas in Communications, Special Issue on Service Enabling Platforms, Vol. 17, No. 9, pp. 1632-1650, September 1999.

[29] Y. Lu, C.Y. Lu, Tarek Abdelzaher, and G. Tao, "An Adaptive Control Framework for QoS Guarantees and its Application to Differentiated Caching Services," In Proc. the Eleventh International Workshop on Quality of Service (IWQoS 2002), pp. 23-32, May 2002.

[30] H.G. Zhang, C.V.Hollot, Don Towsley, and Vishal Misra, "A Self-Tuning Structure for Adaptation in TCP/AQM Networks," In Proc. IEEE Global Telecommunications Conference (GLOBECOM 2003), vol. 7, pp. 3641-3646, December 2003.

[31] Y. Cui, Y. Chait, and C.V. Hollot, "Stability Analysis of a DiffServ Network Having Two-Level Coloring at the Network Edge and Preferential Dropping at the Core," In Proc. IEEE American Control Conference (ACC 2004), vol. 1, pp. 343-348, July 2004.

[32] UCB/LBNL/VINT Network Simulator – NS (version 2), http://www-mash.cs.berkeley.edu/ns/.

[33] X. Chang, and J. K. Muppala, "On Improving Bandwidth Assurance in AF-based DiffServ Networks Using a Control Theoretic Approach," Technical Report HKUST-CS04-09.

[34] H. Balakrishnan, S. Devadas, D. Ehlert, and Arvind, "Rate Guarantees and Overload Protection in Input-Queued Switches," In Proc. IEEE Conference on Computer Communications (INFOCOM 2004), vol.7, pp. 2185-2195, March 2004.

[35] H. Özbay, "Introduction to feedback control theory," Boca Raton : CRC Press, c2000.