Fuzzy-Based Adaptive Bandwidth Control for Loss Guarantees

Peerapon Siripongwutikorn, *Member, IEEE*, Sujata Banerjee, *Senior Member, IEEE*, and David Tipper, *Senior Member, IEEE*

***Abstract*—This paper presents the use of adaptive bandwidth control (ABC) for a quantitative packet loss rate guarantee to aggregate traffic in packet switched networks. ABC starts with some initial amount of bandwidth allocated to a queue and adjusts it over time based on online measurements of system states to ensure that the allocated bandwidth is just enough to attain the specified loss requirement. Consequently, no *a priori* detailed traffic information is required, making ABC more suitable for efficient aggregate quality of service (QoS) provisioning. We propose an ABC algorithm called augmented Fuzzy (A-Fuzzy) control, whereby fuzzy logic control is used to keep an average queue length at an appropriate target value, and the measured packet loss rate is used to augment the standard control to achieve better performance. An extensive simulation study based on both theoretical traffic models and real traffic traces under a wide range of system configurations demonstrates that the A-Fuzzy control itself is highly robust, yields high bandwidth utilization, and is indeed a viable alternative and improvement to static bandwidth allocation (SBA) and existing adaptive bandwidth allocation schemes. Additionally, we develop a simple and efficient measurement-based admission control procedure which limits the amount of input traffic in order to maintain the performance of the A-Fuzzy control at an acceptable level.**

***Index Terms*—Adaptive bandwidth allocation, fuzzy control, quality of service (QoS).**

I. Introduction

A. Background and Motivation

NETWORK users are demanding better quality of service (QoS) guarantees for their traffic, which cannot be achieved by means of per-flow QoS mechanisms due to their lack of scalability. An answer to this challenge is an aggregate class-based traffic management framework, such as differentiated service (DiffServ) [1] proposed by IETF, which helps mitigate the scalability problem. In this framework, the issue of guaranteeing quantitative QoS at the aggregate traffic level still remains. Specifically, traffic at an aggregate level can be highly variable and unpredictable, which makes dimensioning of network resources to provide the required QoS a very difficult problem. Often, wasteful gross over-provisioning of network resources is the only course of action, though this is clearly not economically sustainable. A common approach for a quantitative QoS guarantee is to use static bandwidth allocation (SBA) [2]–[4]. Under SBA, the input traffic is first parameterized to fit some specific stochastic model and a fixed amount of bandwidth that satisfies the required QoS under that particular traffic model is then calculated. Upon each aggregate flow arrival the calculated SBA is reserved at each node along the path using some protocol such as RSVP. As such, any SBA method requires that its relevant traffic parameters must be known *a priori*. In addition, the calculated bandwidth can be either overallocated or underallocated due to inaccuracies in the assumed traffic model, parameter estimation and analytical method used.

The question we address here is how to achieve a loss guarantee to aggregate traffic while achieving high bandwidth utilization without knowing *a priori* detailed traffic information. In order to achieve this objective we propose here an alternative approach to SBA, namely, adaptive bandwidth control (ABC). In contrast to SBA, ABC starts with some initial amount of bandwidth allocation and adjusts it over time to ensure that the allocated bandwidth is just enough to attain a specified loss requirement. As a result, ABC has major advantages when compared to SBA. Particularly:

- no assumption of a specific stochastic input traffic model is required. Thus, there is no need for *a priori* traffic parameterization;
- since the control adapts over time to actual traffic conditions, less bandwidth will be wasted due to overallocation, thereby improving the network utilization.

ABC is the way we envision how quantitative QoS should be guaranteed; users just simply supply their QoS requirements (and possibly very minimal traffic information such as the aggregate average transmission rate) without having to know what traffic model to use, or without having to specify values of complicated traffic parameters. Next, we describe the system model under consideration and how exactly ABC is applied.

B. Model

Consider the output port of an output-queued switch, where the port has an outgoing link capacity of C bps. The port is assumed to support K QoS traffic classes and one best effort (BE) traffic class, each of which has its own queue as shown in Fig. 1. The objective is to guarantee a packet loss rate to traffic in each QoS queue by properly allocating the amount of bandwidth $C_i(t),\ i = 1, 2, \ldots, K$, with the constraint that $\sum_{i=1}^{K} C_i(t) \leq C$. Any unallocated bandwidth $C - \sum_{i=1}^{K} C_i(t)$ is consumed by the BE queue. We will first focus on ABC for a single queue with an unlimited link capacity to allocate, and then describe extensions to the limited link capacity case and the multiple queue and node case. A model of ABC for a single

Manuscript received January 19, 2004. This work was supported in part by the NSF CARRIER award ANI-9702389.

P. Siripongwutikorn is currently with Department of Computer Engineering, King Mongkut's University of Technology Thonburi, Thailand. This work was done while he was at the University of Pittsburgh.

S. Banerjee is with Hewlett Packard Laboratories, Palo Alto, CA 94304 USA.

D. Tipper is with Department of Information Science and Telecommunications, University of Pittsburgh, PA 15260 USA.

Digital Object Identifier 10.1109/TNN.2005.853429

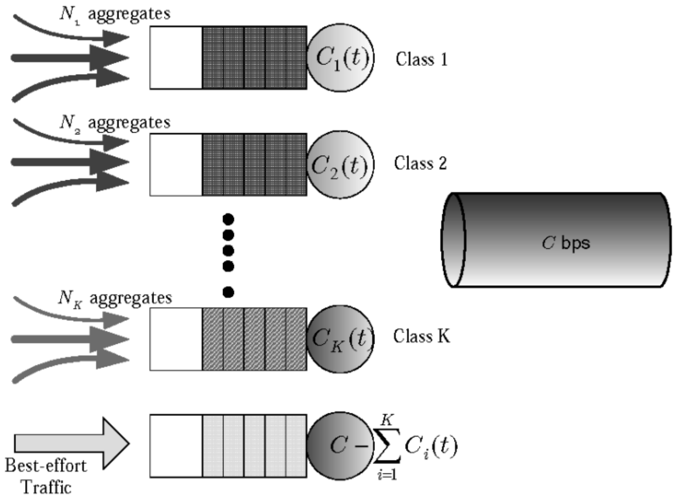


Fig. 1. Transmission link system model.

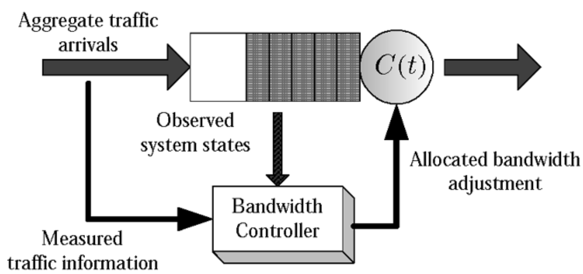


Fig. 2. Model of ABC for a single queue.

queue is depicted in Fig. 2. The bandwidth controller adjusts the allocated bandwidth (service rate) at a time scale between the packet level and the connection level (e.g., one tenth of second to seconds) based on measurement of system states and measured traffic information (and possibly traffic prediction), such that the loss requirement is achieved while maintaining a high bandwidth utilization. In practice, the bandwidth is allocated in discrete time instants rather than in continuous time.

C. Related Work

ABC algorithms have been developed to guarantee different QoS metrics including the average queue length, packet delay, and packet loss rate. However, the existing work on ABC for a loss guarantee thus far has many shortcomings as detailed in our survey paper [5]. Those based on simple linear feedback control directly measure the packet loss to adjust the service rate [6], [7] by means of integral control, and are slow to adapt in case of nonstationary or highly dynamic traffic conditions, leading to poor performance. Furthermore, the packet loss rate can be a poor feedback variable because of its bursty nature and difficulty in accurate measurement. In [8], we modified the integral control in [6], which will be referred in here as I-control, and developed a heuristic for the feedback gain tuning. Simulation results presented in [8] have shown that the modified control has significantly better performance than its original version. The I-control will be briefly described in Section III and will be considered in our comparative performance evaluation (Section IV) with the fuzzy based control proposed in this work.

A loss rate guarantee could also be indirectly accomplished by means of the average queue length control [9]. If a given target loss rate can be mapped to an appropriate target queue length, any ABC of the average queue length could be applied. However, none of the existing work for queue length control so far, appears to be effective, and the issue of the mapping between the target loss rate and the target queue length has to be resolved. Another approach for ABC is based on traffic prediction which allocates bandwidth to match the predicted arrival rate and, hence, results in negligible loss [10]–[14]. Except in [13], this approach is primarily applied to compressed video traffic. In this case, the bit rate in each frame typically varies from one frame to another, and the bandwidth adjustment is made in a per frame basis to match the varying frame rates. However, as shown in [5], the prediction error can be large under aggregate traffic. Furthermore, the resulting performance is difficult to quantify and control, and one may still end up overallocating the bandwidth.

D. Contributions

In this paper, we propose an ABC algorithm called augmented Fuzzy (A-Fuzzy) control to guarantee a packet loss rate. The control consists of two components, namely: A fuzzy-based component and a augmented component. *The fuzzy-based component* maintains the average queue length at an appropriate target value to achieve a desired packet loss rate by means of a fuzzy logic based controller. As will be seen later, tracking the average queue length allows the allocated bandwidth to adapt to the input traffic, resulting in high bandwidth utilization. However, controlling the average queue length is not always sufficient to maintain the loss rate at a desired value because changes in the average queue length can be too fast to track under heavily bursty and correlated traffic, which can result in undesirable bursts of packets loss. *The augmented component* deals with undesirable burst loss by using the measured packet loss rate to adjust the allocated bandwidth as needed between bandwidth adjustment instants of the fuzzy-based component. As a result, the A-Fuzzy control essentially operates on two time scales. We note that the fuzzy-based component can exist without the augmented component, and we simply refer to the resulting control as the *Fuzzy control*. However, its performance is inferior to the A-Fuzzy control, which will be shown later in our performance evaluation.

From an extensive simulation study, the A-Fuzzy control is shown to be robust against a wide range of system parameters (e.g., buffer size, target loss rate), input traffic types, and traffic dynamic conditions, in the sense that a single set of control parameters provides relatively good performance for a variety of cases. Therefore, no further parameter tuning is required, making it easily deployed in real networks. Additionally, we develop a simple and efficient measurement-based admission control procedure for bandwidth allocation under the A-Fuzzy control, whereby only the mean input traffic rate is required from users.

E. Outline

The remainder of this paper is organized as follows. In Section II, we describe the A-Fuzzy control. The selection of the control time interval as well as the mapping from a given target loss rate to the target average queue length (TQL) are also discussed. In Section III, we present the I-control, and briefly describe its properties as well as the gain tuning. In Section IV, we present comparative performance evaluation results between the proposed ABC algorithms and existing SBA methods. A variety of input traffic scenarios are studied including self-similar traffic, real traffic traces, and step input traffic. Additionally, the sensitivity of the control performance to variations in the input traffic characteristics is studied. In Section V, a simple and efficient measurement-based admission control procedure for the A-Fuzzy control is developed for a single queue by applying the performance results under a limited link capacity and the normal approximation to the distribution of the allocated bandwidth. Then, the extension to the multiple queue case is described in Section VI. Lastly, in Section VII we summarize our results and future research directions are given.

II. A-Fuzzy Control

This section describes the design of the A-Fuzzy control. The control consists of two components—the fuzzy-based component and the augmented component. The fuzzy-based component uses a fuzzy logic controller to maintain the average queue length between a lower threshold lth and an upper threshold uth, or equivalently at some TQL, by dynamically adjusting the allocated bandwidth (service rate). Methods to obtain the TQL and the control time interval are discussed after the description of the fuzzy-based component. In the augmented component, the measured packet loss is used to correct the allocated bandwidth calculated by the fuzzy-based component. This is done in a smaller time scale than the control time interval to deal with the impact of traffic burstiness and correlation that cause undesirable packet loss. The control with only the fuzzy-based component will be refered to as the Fuzzy control.

A. *Fuzzy-Based Component*

Under the fuzzy-based component, the allocated bandwidth C_k is adjusted at time instant $t_k, k = 0, 1, 2, 3, \ldots$, where C_k is held constant over $(t_k, t_{k+1}]$. The length of $(t_k, t_{k+1}]$ is called the *control time interval*, which needs not be a fixed constant. Let Q_k be an average of the queue length in packets (including the packet in service) seen by packet arrivals over the kth interval. Two state feedback variables are used. The first one is the exponential weighted moving average (EWMA) of Q_k, given by

$$\hat{Q}_k = \alpha Q_k + (1-\alpha)\hat{Q}_{k-1}. \qquad (1)$$

The value of α is normally small (e.g., $\alpha = 0.1$) to smooth out sudden changes. The other feedback is the normalized change in $\hat{Q}_k$, given by

$$\Delta\hat{Q}_k = \frac{\hat{Q}_k - \hat{Q}_{k-1}}{u\text{th} - l\text{th}}. \qquad (2)$$

Based on $\hat{Q}_k$ and $\Delta\hat{Q}_k$, the change in the bandwidth ΔC_k is determined, and C_{k+1} is updated to $C_k + \Delta C_k$ at the end of the interval.

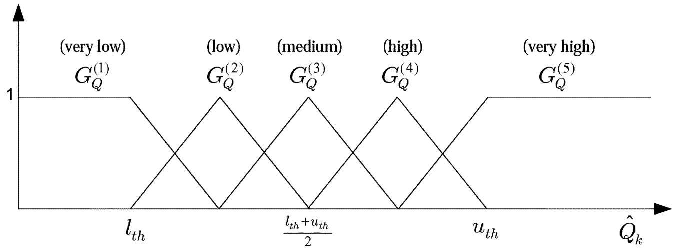


Fig. 3. Membership functions for $\hat{Q}_k$.

The main reason for considering fuzzy control is that it has been shown to work well in systems for which no sufficient detail is available to describe them mathematically, or the model is so complex that it is not amenable to analysis. In our case, the time-averaged queue length (Q_k) of a queue with ABC and the amount of packet loss (L_k) can be described by $Q_k = (1/T_c)\int_{t_k}^{t_{k+1}} \min(B, Q(s))\,ds$ and $L_k = \int_{t_k}^{t_{k+1}} \max(0, Q(s) - B)\,ds$, where $Q(s) = \sup_{s'>0}(q_k + A(s', s) - C_k(s - s'))$, q_k is the queue length at the beginning of time interval k and $A(s', s)$ is the number of traffic arrivals over $(s', s]$. We see that such mathematical model is intractable to techniques in conventional control theory.

Fuzzy control consists of three main steps—fuzzification, inference, and defuzzification. In the fuzzification step, a real number representing a feedback value is converted to linguistic values, each of which characterized by its own membership function. In the inference step, a set of rules called the *rule-base*, which emulates the decision-making process of a human expert, are applied to the linguistic values of the inputs so as to infer the output (fuzzy) sets. These outputs are then defuzzified to the actual control signal for the process. These steps will be described in the following.

Fuzzification is the process of translating real number inputs of each feedback to linguistic values. Five linguistic values are defined for each feedback variable. For $\hat{Q}_k$, its linguistic values $A_Q^{(m)}, 1 \le m \le 5$, are 1) very low, 2) low, 3) medium, 4) high, and 5) very high, with the corresponding membership functions $G_Q^{(m)}(\hat{Q}_k)$ shown in Fig. 3. The membership functions are defined such that $\sum_{m=1}^{5} G_Q^{(m)}(\hat{Q}_k) = 1, \forall \hat{Q}_k \in [0, B]$ where B is the buffer size.

Similarly for $\Delta\hat{Q}_k$, its linguistic values $A_\Delta^{(m)}, 1 \le m \le 5$, are defined as 1) −high, 2) −low, 3) zero, 4) +low, and 5) +high, with their membership functions $G_\Delta^{(m)}(x)$ shown in Fig. 4. As before, the membership functions are defined such that $\sum_{m=1}^{5} G_\Delta^{(m)}(\Delta\hat{Q}_k) = 1, \forall \Delta\hat{Q}_k \in \mathbb{R}$. Here, the membership functions of $\Delta\hat{Q}_k$ have one tunable parameter d, which in our numerical results is set to 0.05. That is, we consider $\Delta\hat{Q}_k$ as being low if it is about $\pm 5\%$ of the threshold gap $(u\text{th} - l\text{th})$, and as being high if it is about $\pm 10\%$ of the threshold gap. We show later that fairly good control results are obtained without any further tuning of the membership functions.

After the real number inputs are mapped to the linguistic values through the membership functions in the fuzzification step, the inference step decides which output control values should be applied to the system by using a rule base. The rule base is a set of rules that emulates the decision-making process of the human expert controlling the system, which is where

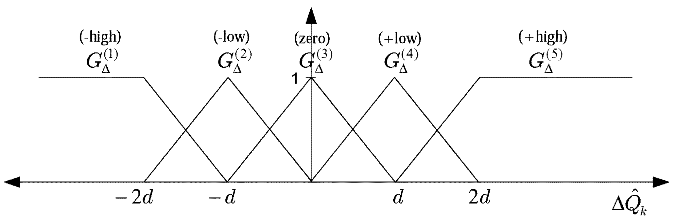


Fig. 4. Membership functions for $\Delta\hat{Q}_k$.

TABLE I
VALUES OF x_i FOR RULE-BASE

$\hat{Q}_k$	$\Delta\hat{Q}_k$ -high	-low	zero	+low	+high
very low	-2%	-1.5%	-1%	-0.5%	0%
low	-1.5%	-1%	-0.5%	0%	1%
medium	-1%	-0.5%	0%	1%	1.5%
high	-0.5%	0%	1%	1.5%	2%
very high	0%	1%	1.5%	2%	2.5%

heuristic information of how to achieve good control is systematically incorporated. In our case, we adopt a Sugeno-type fuzzy control inference and the rule base is expressed in a form:

Rule i : IF $\hat{Q}_k$ is $A_Q^{(l)}$ and $\Delta\hat{Q}_k$ is $A_\Delta^{(m)}$
THEN Change C_k by x_i% of λ.

where λ is the average input traffic rate. From the rule base, x_i is the amount of change in the service rate due to rule i. The values of x_i's are established based on insights of the queue behavior. For example, if $\hat{Q}_k$ is low and $\Delta\hat{Q}_k$ is zero, then x_i should be a small negative value in order to decrease the service rate and, hence, would likely increase $\hat{Q}_{k+1}$. The values of x_i's selected for use are tabulated in Table I. Since λ may not be known or may dynamically change over time, we use an online measurement version of λ, denoted by $\hat{\lambda}$. Therefore, the absolute amount of bandwidth adjustment in the rule-base $(\hat{\lambda}x_i)$ actually changes with the average input traffic rate, enabling the control to adapt to traffic dynamics. Here, the measurement time period for $\hat{\lambda}$ is 10 000 packet arrivals, and the EWMA with the weighting factor of 0.3 is applied to track dynamic changes.

In the inference step, we determine the change in the service rate contributed by rule i, denoted by b_i, by weighting $x_i\hat{\lambda}$ with the term $G_{Ql}(\cdot) * G_{\Delta m}(\cdot)$. In other words

$$b_i = G_Q^{(l)}(\hat{Q}_k) * G_\Delta^{(m)}(\Delta\hat{Q}_k) * x_i * \hat{\lambda}$$

If rule i is not active, then b_i will simply be zero because either $G_Q(\hat{Q}_k)$ or $G_\Delta(\Delta\hat{Q}_k)$, or both are zero. Due to the complementary nature of the membership functions, the weights from all the active rules sum to unity. Once the output control values b_i's have been determined from the inference step, they are combined in the defuzzification step to obtain ΔC as $\Delta C = \sum_{i=1}^{25} b_i$, and C_{k+1} is set to $C_k + \Delta C$.

B. Control Time Interval Selection

Because the feedback used is an average of the queue lengths seen by packet arrivals, the control time interval is the time taken to obtain a required number of packet arrivals to calculate the feedback. We next discuss a heuristic modified from the concept of dominant time scale to select the control time interval. The underlying idea is that the control time interval must be less than the time scale that the input traffic correlation affects the queueing performance. This time scale is known as the *dominant time scale* (T_d). The study of the dominant time scale in [15] under a single queue with a constant service rate and stationary input traffic essentially reveals that such time scale is an increasing function of the buffer size B and depends on the input traffic types. Our control time interval should be less than T_d to ensure effective control performance. Otherwise, the control may be too slow to react to the input traffic. For a queue having a constant service rate C and fed by fractional Gaussian noise (fGn) input traffic with mean rate λ, variance coefficient a, and Hurst parameter H, the dominant time scale is $T_d = (H/(1-H))(\rho/(1-\rho))(B/\lambda)$, where $\rho \equiv \lambda/C$. However, the formula is valid only for fGn traffic. Given that we do not want to assume any *a priori* information on the input traffic, we come up with an approximation. Assuming that $H = 0.9$, we have the approximation

$$T_d \approx 9\frac{\rho}{1-\rho}\frac{B}{\lambda}. \tag{3}$$

In our case, since both C and λ can change from one interval to another, the number of required packets in each control time interval becomes $N_k = \hat{\lambda}T_d = 9B\rho_k/(1-\rho_k)$, where ρ_k is the server utilization in the kth interval (exponentially weighted with the weight of 0.3) and $\hat{\lambda}$ is the measured average arrival rate.

The concept of dominant time scale used to derive N_k above is valid only in a queue with fixed capacity in a steady state. In a queue with ABC, the system often stays in a transient mode and may never reach a steady state due to varying C_k. In other words, the time scale of traffic correlation that impacts the queueing performance under ABC can be much less than T_d, and using the above N_k can lead to poor performance since it can be too large. As a result, we bound N_k to 10 000 samples to limit the impact the approximation error in T_d as well as to obtain a small enough control granularity. Additionally, we cannot make the control time interval arbitrarily small because the processing overhead will become too high if the arrival rate is large, and the average queue length widely fluctuates such that the control does not effectively maintain the average queue length at the target. In addition, bandwidth thrashing may occur under highly fluctuating feedback. Therefore, we set the lower bound on N_k to 1 000 samples. As such, the required number samples in each control time interval becomes

$$N_k = \max(1000, \min(10000, 9\frac{\rho_k}{1-\rho_k}B)). \tag{4}$$

C. TQL Tuning

The remaining question is how to determine the TQL, denoted by Q_r, for the control objective from a given packet loss requirement ε. Given the TQL, the threshold pair uth and lth discussed in Section II is given by

$$l\text{th} = \max(0, Q_r - (\delta/2) * B) \quad (5)$$
$$u\text{th} = \min(B, 2Q_r - l\text{th}) \quad (6)$$

where $0 < \delta < Q_r/B$ is the buffer space between the two thresholds in terms of the fraction of the buffer size. Here, we set δ to 7% (or 0.07).

To eliminate the need for *a priori* input traffic characteristics, the TQL is dynamically adjusted from some initial value to the value that yields the target loss rate by using the loss feedback. The TQL tuning is done repeatedly to keep the loss rate at a desired value. Unlike the bandwidth adjustment process, a long measurement interval for the loss feedback has no significant effect on the TQL tuning process because the TQL adjustment is performed over a much longer time scale. Note that, the use of a measured loss rate will limit the range of achievable target loss requirement because the long measurement time in cases of small target loss rates will render the TQL tuning process slow to converge. However, for medium to high target loss rates, i.e., those greater than 10^{-4}, the tuning process is fast enough for our purpose. Besides, an 10^{-4} order of packet loss rate should be sufficient for many QoS applications on the Internet (e.g., voice over IP and video streaming), which can tolerate or is adaptive to congestion conditions.

We choose the initial TQL based on a $G/M/1$ queueing model, which requires no traffic parameters. From basic queueing theory, the average queue length $\overline{Q}$ and ε in the $G/M/1$ queue are related by $\overline{Q} = \varepsilon^{(1/(B+1))}/(1-\varepsilon^{(1/(B+1))})$. This initial TQL value obtained from the $G/M/1$ approximation tends to be high because it is obtained from the tail probability approximation and the packet interarrival time is expected to be highly correlated instead of being IID as assumed in the $G/M/1$ model. Because the actual input traffic is likely to be more correlated, an initial TQL equal to $(3/4)\overline{Q}$ is used instead to provide a conservative approximation.

Next, we present an algorithm that adapts the initial TQL to an appropriate TQL. The packet loss rate due to a currently used TQL is measured over some number of packet arrivals N_p calculated from a standard statistical method as follows. If the packet loss is in the order of ε, to get an accuracy of r% of ε at $100(1-\alpha)$% confidence interval, one must observe the number of packets of $N_p = z^2_{1-\alpha/2}(\varepsilon(1-\varepsilon)/((r/100)\varepsilon)^2)$, where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of a unit normal variate. Here we use $r = 10$ and a 95% confidence interval. The TQL adjustment algorithm uses a simple multiplicative gain $Q_r(1+g) \rightarrow Q_r$ as shown in Algorithm 1. The algorithm starts by observing N_p packet arrivals to get a rough estimate of the packet loss rate $\hat{\varepsilon}$. The algorithm adjusts the TQL whenever $\hat{\varepsilon}/\varepsilon > 1.12$ or $\hat{\varepsilon}/\varepsilon < 0.89$ (line 5), and the average queue length has actually been kept between the target thresholds (line 11). The latter condition prevents the TQL from being decreased to zero under a sustaining high packet loss rate because of the failure to control the average queue length. The multiplicative factor of 0.2 in line 6 governs how aggressive the adjustment is. Lastly, the gain g is bounded between -0.5 and 0.5 (line 10) to prevent the TQL from being changed too radically. After a new TQL is obtained, the threshold pair for the control is set according to (5) and (6). Another N_p packet arrivals are then observed and the algorithm repeats indefinitely.

```
1: Q_r: Target queue length, ε: Target loss rate, ε̂: Measured loss rate
2:
3: Wait until N_p packets have arrived to obtain ε̂.
4: g ← log_10(ε/max(ε̂, 10^-9));
5: if   abs(g) > 0.05 then
6:    g ← 0.2 * g;
7: else
8:    g ← 0;
9: end if
10: g ← max(−0.5, min(0.5, g));
11: if u_th ≥ Q̄ ≥ l_th then
12:    Q_r ← Q_r(1 + g);
13: end if
```

Algorithm 1: TQL adjustment.

D. Augmented Component

Extensive simulation experiments have been conducted to obtain results on the performance of the control with only the fuzzy-based component as detailed in [8]. We report sample results with input traffic that is fGn with $\lambda = 10000, a = 1$, and $H = 0.85$, with the target loss rate $\varepsilon = 10^{-4}$. Fig. 5 shows the sample paths of the EWMA queue length under the fuzzy control for $B = 200$ packets and $B = 900$ packets. It has been found that, under a large buffer size $(B = 900)$, the packet loss within some interval can become large due to too low C_k, and the average queue length will grow far beyond uth during that particular time interval. The reason is that average queue length can fluctuate more widely when the buffer size is large. This fluctuation in turn causes sudden increases in the average queue length, which cannot be effectively handled by the fuzzy control. According to the rule-base described earlier, the queue length feedback at uth and ten times of uth would induce the same amount of ΔC, which is 2.5% of the average arrival rate (given that $\Delta\hat{Q}_k$ is +high). Intuitively, the latter case should require a larger value of ΔC.

Evidently, controlling the average queue length by using fuzzy control alone may not provide satisfactory loss performance. We have to deal with the effect of bursty traffic on the packet loss which would also reduce the sudden increases in the average queue length. Our approach is to introduce the augmented component, which performs asynchronous increases in the allocated bandwidth when packet losses occur, in addition to the bandwidth adjustment due to the fuzzy-based component. In each control time interval k, suppose that we expect N_k packets to arrive according to our control time selection described previously. A series of asynchronous bandwidth increases of y%, i.e., $(1 + y\%)C_k \rightarrow C_k$, are invoked if the number of lost packets in the time interval reaches

$$l_k(i) = 10^{(i-1)} N_k \varepsilon, \quad i = 1, 2, \ldots \quad (7)$$

which is an exponential type of threshold setting. That is, the 1st bandwidth increase is called for when $N_k\varepsilon$ packets losses occur; the 2nd at $10N_k\varepsilon$; and so on. For example, at $\varepsilon = 10^{-4}$ and $N_k = 10000, l_k(1) = 1$. If a single packet loss occurs, the bandwidth is increased by y% of the current value. Next, we have $l_k(2) = 10$. If another 9 packets are lost during that same time interval (which results in a total loss of 10 packets), another

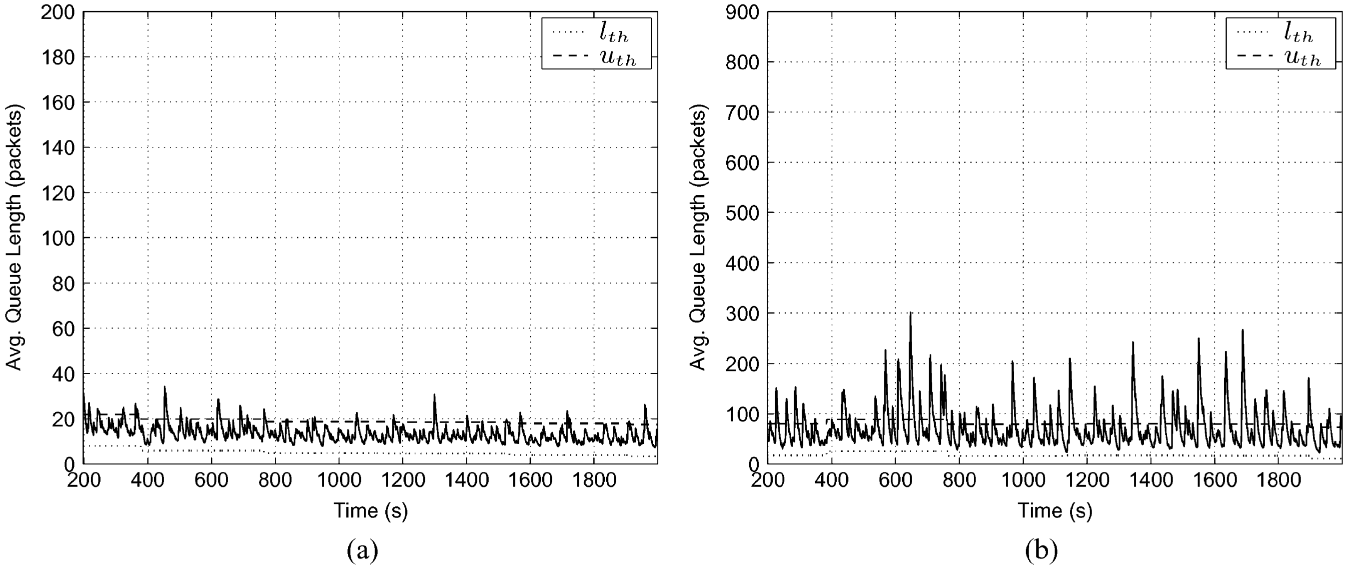


Fig. 5. Sample paths of the EWMA queue length under fuzzy control with fGn traffic ($\lambda = 10000, a = 1, H = 0.85, \varepsilon = 10^{-4}$).

$y\%$ increase on the allocated bandwidth is made, and so on. If an asynchronous bandwidth adjustment by the previous algorithm has been made during the kth interval, the current value of C_k will continue to be used in the $(k+1)$th interval, and the amount of ΔC_k calculated by the fuzzy-based component is ignored. We note that the approach described previously is similar to the one in [16], where the bandwidth adjustment can be made at every fixed-length subinterval of length I within each control time interval. However, such periodic bandwidth adjustment is ineffective because the adjustment instant may be triggered after a large burst loss has already occurred.

For the sake of simplicity, we only consider the amount of increase y to be dependent on the buffer size. For small buffers, the queue sees less bursty traffic due to the dropping effect of a small buffer and, hence, a small increase in the allocated bandwidth should be sufficient. On the other hand, as the buffer size increases, the queue sees more bursty traffic and, hence, we require a greater amount of bandwidth increase. After some rough trail-and-error tuning, we found that a linear interpolation of $y = 2\%$ and 5% from the buffer size of 30 to 900 packets provides reasonable performance, and will be used throughout.

E. Remarks on Control Parameters and Complexity

We have not attempted to tune the A-Fuzzy controller to provide optimal performance because this appears to be very difficult due to the many degrees of freedom associated with the membership functions, rule-base, and the parameters thereof. However, unlike other types of control, fuzzy control has been known to be robust against its parameter selection because of its rule base that smoothes out errors in the control signal. Therefore, we believe that intuitive tuning of the fuzzy control parameters is sufficient, as confirmed by our extensive simulation results presented later. For example, from Table I, $-2.5\% \leq x_i \leq 2\%$. We found that doubling the range so that $-5\% \leq x_i \leq 4\%$ and scaling the value of x_i's in the table accordingly does not provide a significant performance difference.

The amount of computation in the controller depends on the number of rules, which in turn depends on the number of membership functions. Suppose there are N_1 membership functions for Q_k and N_2 for ΔQ_k. The fuzzification step requires N_1+N_2 function calls to obtain the membership value of each function. Each rule requires two multiplications in the inference step, and the defuzzification requires $N_1 N_2 - 1$ additions. As such, the computational complexity is on the order of $N_1 N_2$.

III. Integral Control Using Loss Feedback

This section explores a simple ABC based on integral control. We have found that this type of control can perform surprisingly well in some cases, but has not been investigated or reported elsewhere. Let ε be a target loss rate that we want to achieve. ABC based on integral control (I-control) of loss feedback is in a form

$$C_{k+1} = C_k + G(L_k - \varepsilon A_k) \tag{8}$$

where G is the feedback gain, L_k is the number of lost packets over $(t_{k-1}, t_k]$, and A_k is the number of packet arrivals over $(t_{k-1}, t_k]$. This form of control is adapted from the one developed in [6] by modifying the decreasing gain G/k to a constant gain G. With the decreasing gain, C_k will converge to some constant value, and has been shown in [5] to perform poorly in achieving a given target loss rate because C_k no longer adapts after some period of time. On the other hand, using the constant feedback gain allows the control to regularly adapt to dynamic traffic conditions. The two parameters of the I-control are the feedback gain G and the control time interval T_c, which is the time duration of $(t_{k-1}, t_k]$.

Although this I-control appears to be simple to use, optimal values of G and T_c are difficult to determine because they may change with the input traffic types, the target loss rate, and the buffer size. In [8], we partially address the selection of G based on insights obtained from extensive simulation results on fGn traffic. We have found that the optimal gain does not depend on the buffer size and the average input traffic rate. Only major results will be summarized here due to space limitations. By fixing the second order statistics of the input traffic and the control time interval, we are able to numerically express the optimal gain only in terms of the target loss rate using extensive simulation results to fit a regression model. Specifically, the optimal gain is given by

$$G = \frac{0.335 \log(\varepsilon) + 3.23}{\varepsilon}. \tag{9}$$

Since this formula is valid only for a limited range of ε and for a specific set of traffic parameter values, it will provide suboptimal performance in other cases.

IV. Performance Evaluation

This section presents a comparative simulation study of three ABC algorithms in a single buffer queue, including the Fuzzy control, the A-Fuzzy control, and the I-control. SBA methods will also be compared where possible. The simulation is implemented using CSIM,[1] a process-oriented simulation language based on C. The performance metrics are the bandwidth utilization and the deviation of the measured cumulative packet loss rate ($\hat{\varepsilon}$) from the target loss rate (ε), measured by $\log_{10}(\hat{\varepsilon}/\varepsilon)$.

A. Traffic Descriptions

Extensive simulations have been conducted in [8] and [17] to argue that ABC can be a viable alternative to SBA, for various traffic types and scenarios. Here due to space limitations, we summarize the results for the cases of fGn traffic and real traffic traces only. Each traffic type is described in the following.

1) fGn Traffic: An fGn process exhibits long-range dependent (LRD) behavior, a key feature of network traffic that strongly impacts the network performance. An fGn traffic source can be characterized by three parameters—the mean rate (λ), variance coefficient (a), and Hurst parameter (H). The parameters a and H, respectively, represent the degrees of burstiness and correlation in the traffic. The fGn traffic generator based on fast Fourier transform described in [17] is used.[2]

The main reason for using fGn traffic is that important statistical traffic characteristics including burstiness and correlation can easily be controlled, which is useful in examining the robustness of ABC algorithms. Two SBA methods for comparison with ABC here are the Effective Bandwidth (EB) formula for LRD traffic by Norros [18], and the maximum variance asymptotic (MVA) technique for a finite buffer [19]. The MVA technique is so far the most nearly accurate SBA method for aggregate traffic. *However, it is important to keep in mind that these SBA methods require the knowledge of detailed traffic parameters while ABC does not.*

2) Traces Input: We also consider a traffic trace of IP-level packets collected from measurement over a single ATM virtual circuit connection at an access point.[3] Since the entire trace itself lasts 24 hours, we excerpted from the two portions from the entire trace which appear to have the highest traffic rate variation. We will refer to these two traces as Trace-1, which lasted 4 hours, and Trace-2, which lasted 8 hours. Both traces have a relatively low packet arrival rate ($\approx$1.2 Mb/s in average rate and an average packet size of 596 B for Trace-1) but have a high variance, as indicated by the plot of $\mathrm{var}[R_t]/\mathbb{E}[R_t]$ against t in Fig. 6. The value at which $\mathrm{var}[R_t]/\mathbb{E}[R_t]$ converges is equivalent to the variance coefficient a of the fGn traffic. As such, we have $a \approx 20$ for Trace-1 and $a \approx 40$ for Trace-2, which is much higher than the value used in the fGn traffic considered. However, aggregate traffic in a core network typically has a much larger average rate than the trace considered here [20]. Therefore, for each of the two traces, we construct an aggregate with larger arrival rate by multiplexing many of its instances together, each of which starts at a uniformly distributed time in the trace and wraps around to the starting time.

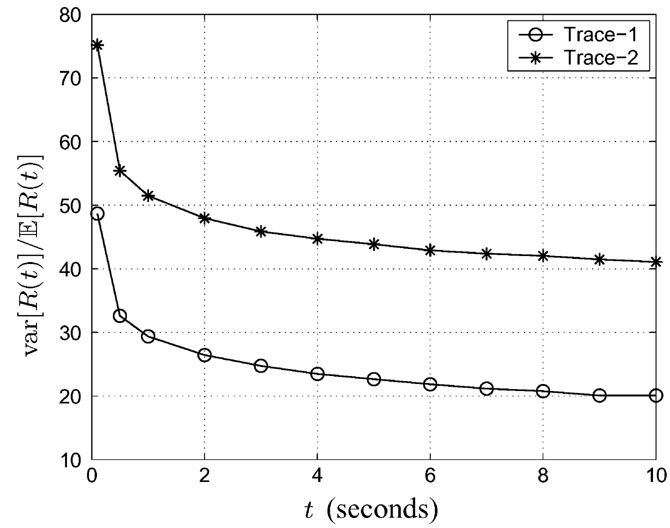


Fig. 6. Variance of arrival rate over mean rate at different time scale t.

3) Step Input: The step input is an abrupt shift in the mean traffic rate either when a new aggregate joins the queue or an existing aggregate departs the queue. It provides the worst-case scenario and, hence, the worst-case response time and loss performance. In our context, the response time is the time taken for the allocated bandwidth to reach its new level, which is determined by visual assessment. We make no assumption that the controller is aware of the step input due to new aggregate arrivals or departures.

B. Simulation Results

Each simulation run lasts 2000 s unless explicitly mentioned. The packet size distribution used in the fGn traffic is obtained from real measurement in IP backbone networks, with the average value of 427 B.[4] More specifically, 55% of packets is 40-B long, 15% is 576-B long, 12% is 1500-B long, and 20% is between 40–1500 B long. The performance metrics are the bandwidth utilization and the deviation of the measured cumulative packet loss rate ($\hat{\varepsilon}$) from the target loss rate (ε), measured by $\log_{10}(\hat{\varepsilon}/\varepsilon)$. Positive values of $\log_{10}(\hat{\varepsilon}/\varepsilon)$ mean that the resulting loss rate is higher than the target while negative values of $\log_{10}(\hat{\varepsilon}/\varepsilon)$ mean that the resulting loss rate is lower than the target. Ideally, the best scheme would yield the highest bandwidth utilization with $\log_{10}(\hat{\varepsilon}/\varepsilon)$ equal to zero.

[1] http://www.mesquite.com.

[2] The slot size or time step of 5 ms and uniform arrivals within the bin are used.

[3] The trace was obtained from University of Auckland Internet uplink in Dec. 1999. The measurement work was supported by the NSF Grant no. ANI-9 807 479 and the NLANR Measurement and Network Analysis group under vBNS and Internet2 Abilene networks.

[4] http://advanced.comms.agilent.com/RouterTester/member/journal/JTC_003.html.

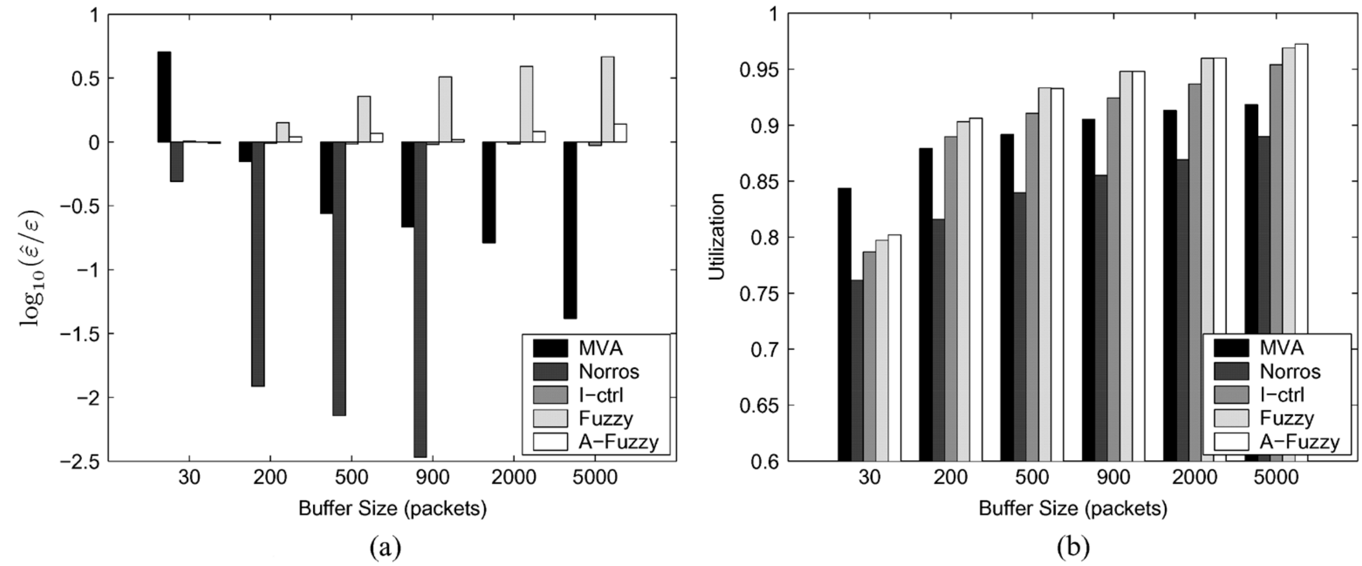


Fig. 7. Performance comparisons for fGn traffic ($\lambda = 10000, a = 1, H = 0.85$), ($\varepsilon = 10^{-3}$).

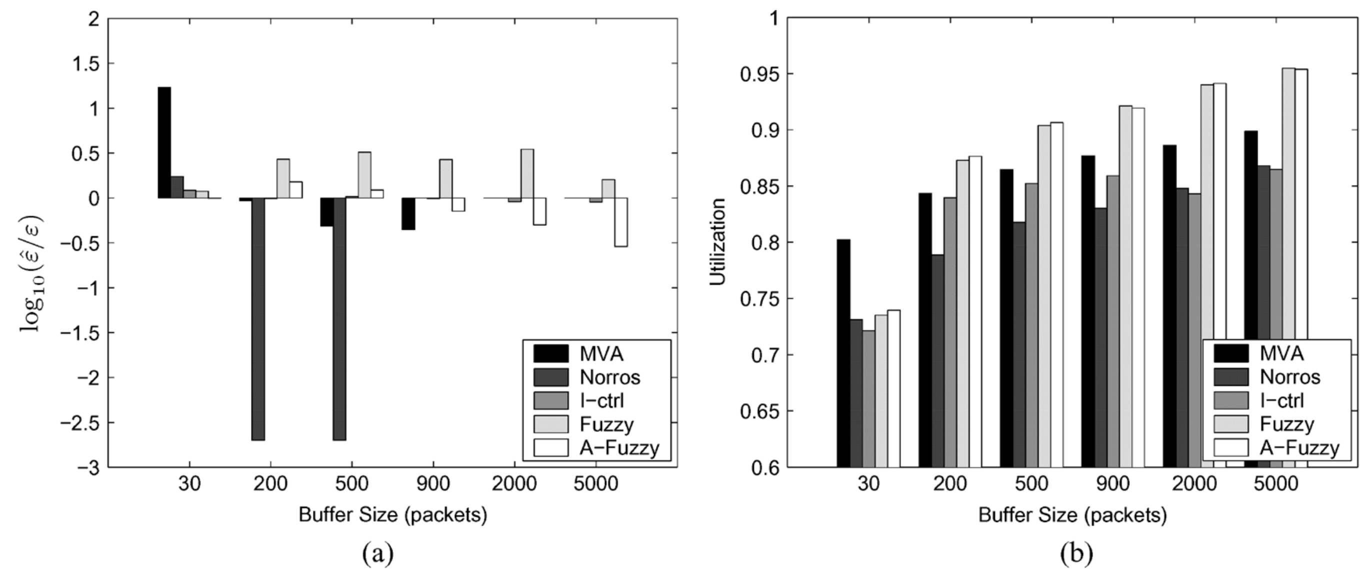


Fig. 8. Performance comparisons for fGn traffic ($\lambda = 10000, a = 1, H = 0.85$), $\varepsilon = 10^{-4}$.

fGn Traffic: We set a baseline traffic configuration to $\lambda = 10\,000$ packets/s, $a = 1$, and $H = 0.85$. Sensitivity of the control performance to the traffic parameters will be discussed later. Figs. 7 and 8 show the utilization and packet loss performance over a range of buffer sizes (B) under the baseline case, respectively, for $\varepsilon = 10^{-3}$ and 10^{-4}. Except for a small buffer size (30 packets), the MVA method yields good utilization and a loss rate that is lower than the target. These results are not surprising because the MVA method is derived under the Gaussian approximation. Norros EB formula is also somewhat accurate when the buffer is small but significantly overestimates the loss when the buffer is large, leading to underutilization. Specifically, the resulting packet loss rates are about two orders of magnitude lower than the target at large buffers, and no loss at $B > 900$ packets, $\varepsilon = 10^{-4}$. Both A-Fuzzy control and I-control are able to achieve higher utilization than the MVA approximation and have good loss rates. However, the A-Fuzzy control has consistently better performance than the Fuzzy control (and the I-control control), especially at a low target rate and large buffer size.

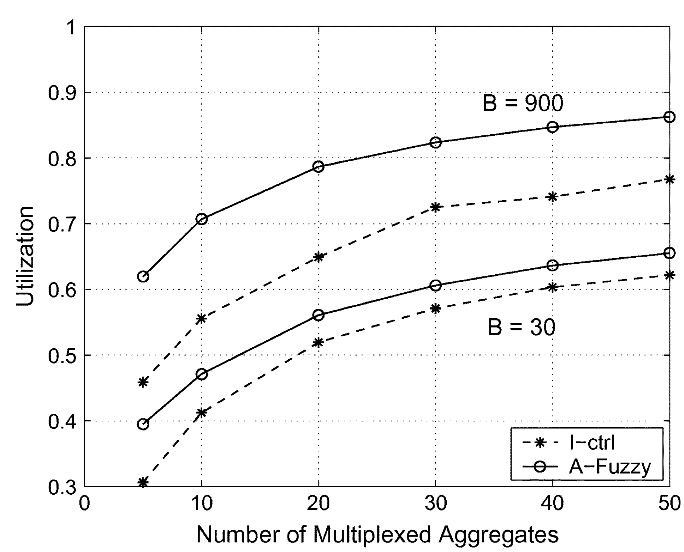


Fig. 9. Utilization comparison for Trace-2 ($\varepsilon = 10^{-4}$).

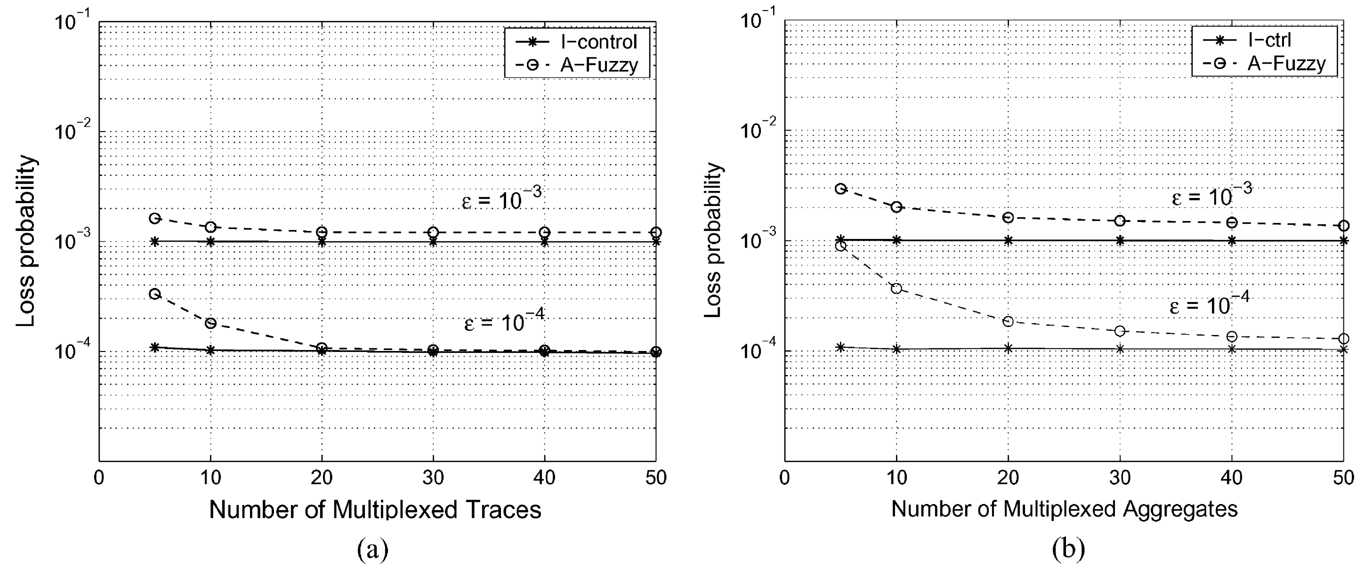


Fig. 10. Loss comparison for Trace-1 and Trace-2 (Buffer $=$ 900 packets).

1) Trace Input: Fig. 9 shows the utilization of the I-control and the A-Fuzzy control versus the number of multiplexed traces, respectively, for Trace-2. The A-Fuzzy control achieves better utilization in every case but its achievable loss rate deviates from the target when the number of multiplexed traces is less than 10 for Trace-1 and less than 30 for Trace-2 as shown in Fig. 10. A higher number of Trace-2 aggregates are needed because the arrival rate of Trace-2 has a higher variance. However, this is not a serious problem because we expect the A-Fuzzy control to operate in a core network environment, whereby traffic has a large volume and is relatively smooth as indicated by recent measurement of the Internet core backbones reported in [20].

2) Step Inputs: Two types of step input are considered—step increase and step decrease. The step increase traffic models the effect of admitting a new aggregate in the queue, while the step decrease traffic models the effect of aggregates departing the queue. For the step increase traffic, we use the baseline case of fGn traffic ($\lambda = 10000, a = 1, H = 0.85$), with λ being increased by 50% right after the simulation time has reached 2500 s. For the step decrease traffic, we use fGn traffic with $\lambda = 15000, a = 1, H = 0.85$, with λ being decreased by 33% at time 2500 s. The performance comparison will be made only between the A-Fuzzy control and the I-control by looking at the response time and the ability to satisfactorily maintain a packet loss rate.

Fig. 11 shows the sample paths of the cumulative loss rate for step increase and step decrease traffic with $\varepsilon = 10^{-4}$ and $B = 500$ packets. In the case of the step increase traffic, a sudden increase of the loss rate at the traffic shift instant occurs in both controls, but the amount of increase in the I-control is smaller. However, the A-Fuzzy control also has comparable loss increase to the I-control while attaining better utilization (0.912 versus 0.875). In case of the step decrease traffic, both controls can maintain the packet loss rate at the target. However, the response time of the I-control is much longer as seen from the sample paths of the allocated bandwidth in Fig. 12. In this case, the I-control cannot keep up with the rapid decrease in the input traffic due to its long response time caused by a linear decreasing allocated bandwidth. Under the step decrease traffic, the A-Fuzzy control is still able to keep up with the traffic, and achieve the target loss rate while maintaining good utilization.

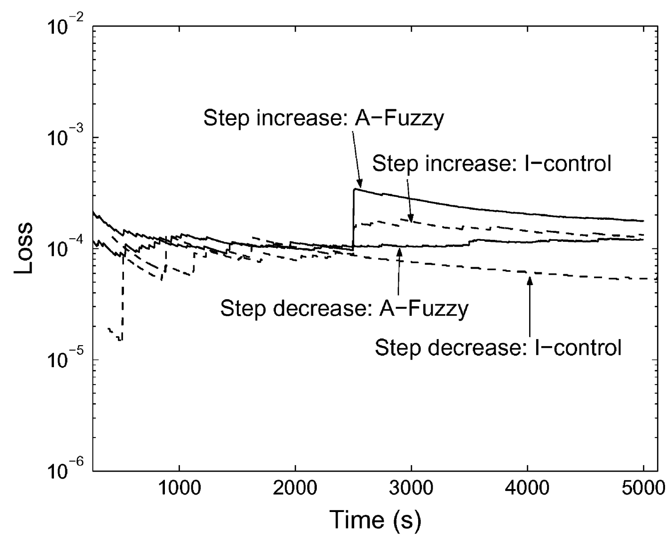


Fig. 11. Loss performance under step-input fGn traffic ($\varepsilon = 10^{-4}$, Buffer $=$ 500 packets).

C. Discussion

According to the simulation results, ABC can attain a given target loss rate with high utilization even without prior knowledge of the input traffic. Among the three ABC algorithms investigated, the A-Fuzzy control consistently provides the best utilization compared to the other two, which can be explained as follows. With large buffers, bursty traffic causes a highly fluctuating buffer content that cannot be effectively controlled by the Fuzzy control (i.e., the fuzzy-based component), resulting in a number of rapid increases in the average queue length and, hence, a loss rate somewhat above the target. With the A-Fuzzy control, the effect of bursty traffic is dealt with through the augmented component (by rapidly increasing the allocated bandwidth) and, thus, the average queue length under the A-Fuzzy

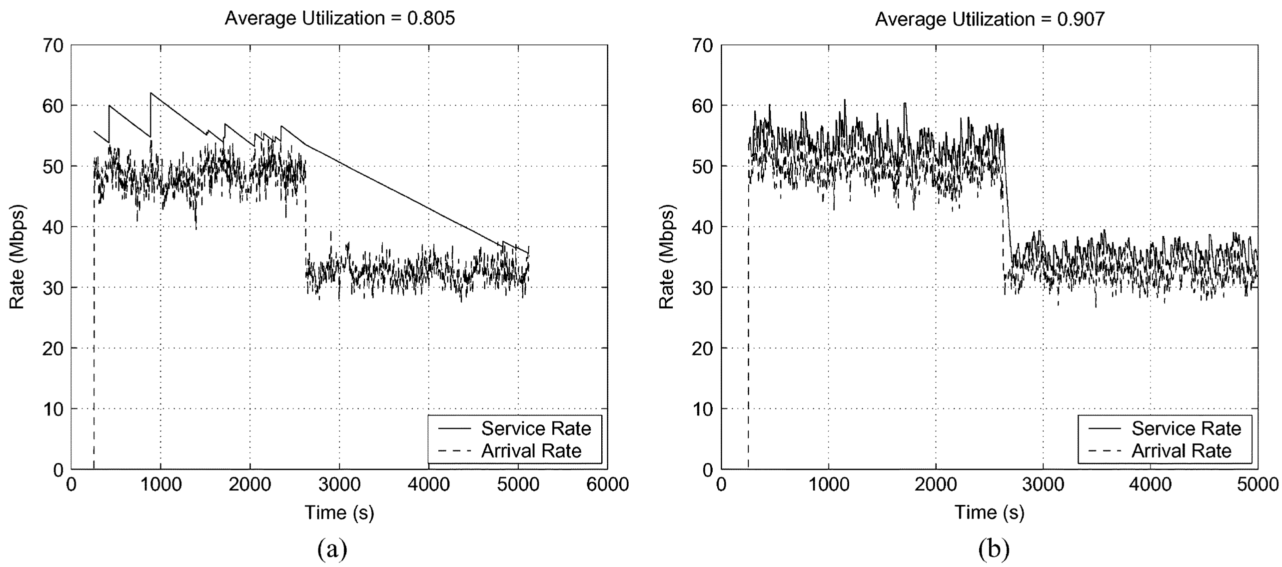


Fig. 12. Allocated bandwidth under step decrease fGn traffic (Initial $\lambda = 15000$ packets/s, $a = 1, H = 0.85$), $\varepsilon = 10^{-4}$, Buffer = 500 packets.

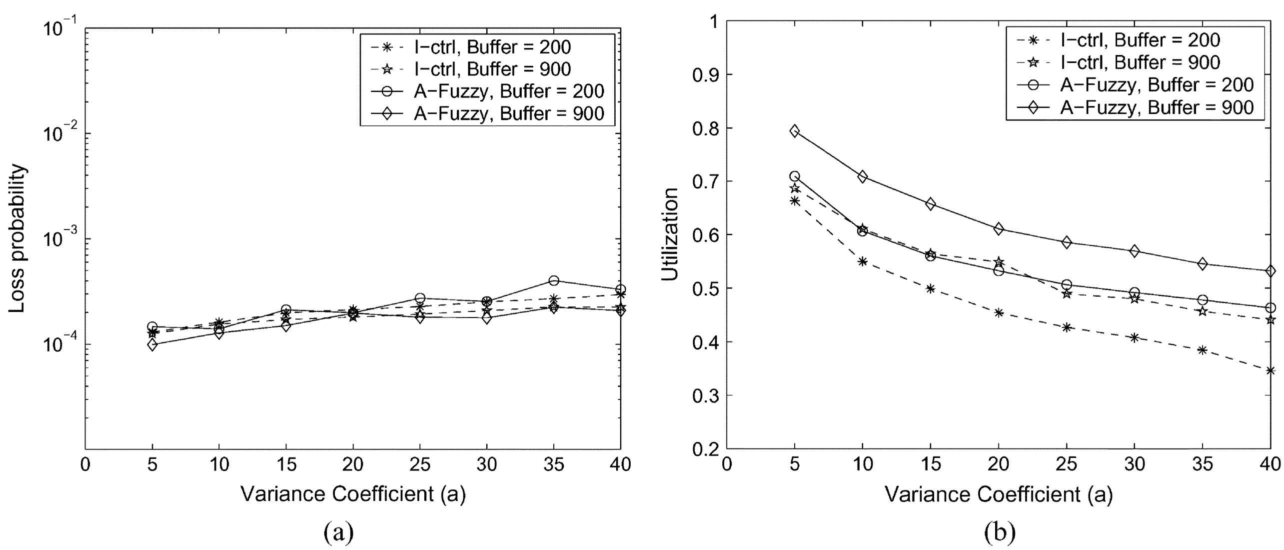


Fig. 13. Performance comparison between I-control and A-Fuzzy control over a range of variance coefficient $a(\varepsilon = 10^{-4})$.

control is better controlled. As such, the A-Fuzzy control can achieve utilization as good as that of the Fuzzy control while achieving a better loss rate. Under less bursty traffic arrivals or small buffer size, both Fuzzy and A-Fuzzy controls are practically the same because the augmented component does not play a significant role.

Compared to the I-control, the A-Fuzzy control can achieve better utilization because it allocates the bandwidth that closely tracks the input traffic (with the fuzzy-based component) while the I-control has an approximately linear bandwidth decrease, as shown in Fig. 12. As the target loss rate reduces, ΔC_k in the I-control also reduces (from $G(L_k - \varepsilon A_k)$), leading to an even slower response time. Also, the gain G is not optimal because it was optimized for a specific input traffic type as previously discussed in Section III. On the other hand, the A-Fuzzy control has a control structure that can incorporate heuristic information to the control and smooth out any control error, making it robust over a wide range of system configurations.

In most of the traffic scenarios, the A-Fuzzy control yields a loss rate lower than the target at small buffer sizes, higher than the target at medium buffer sizes, and right at the target at the largest buffer size. We believe that this behavior is attributed to the parameter settings in the augmented component, namely, the amount of bandwidth increase and the loss threshold to adjust the bandwidth. *Nevertheless, we have shown that the parameter settings based on intuition can still give good performance on different input traffic types and system configurations.*

D. Sensitivity Analysis

This section will illustrate the robustness of the A-Fuzzy control and the I-control with respect to changes in the burstiness (a) and correlation (H) of the fGn traffic. We conduct experiments to see how parameters a and H affect the performance of the ABC algorithms. The mean arrival rate λ will be fixed to 10 000 packets/s. In Experiment I, we fix $H = 0.85$ and vary a from 1 to 21. In Experiment II, we fixed $a = 1$ and vary H from 0.6 to 0.95. In these cases, we allow the control to operate in a relatively long period of time so that the control performance reaches its steady-state behavior. Fig. 13 plots the loss rate and utilization performance for Experiment I at the buffer size of 200

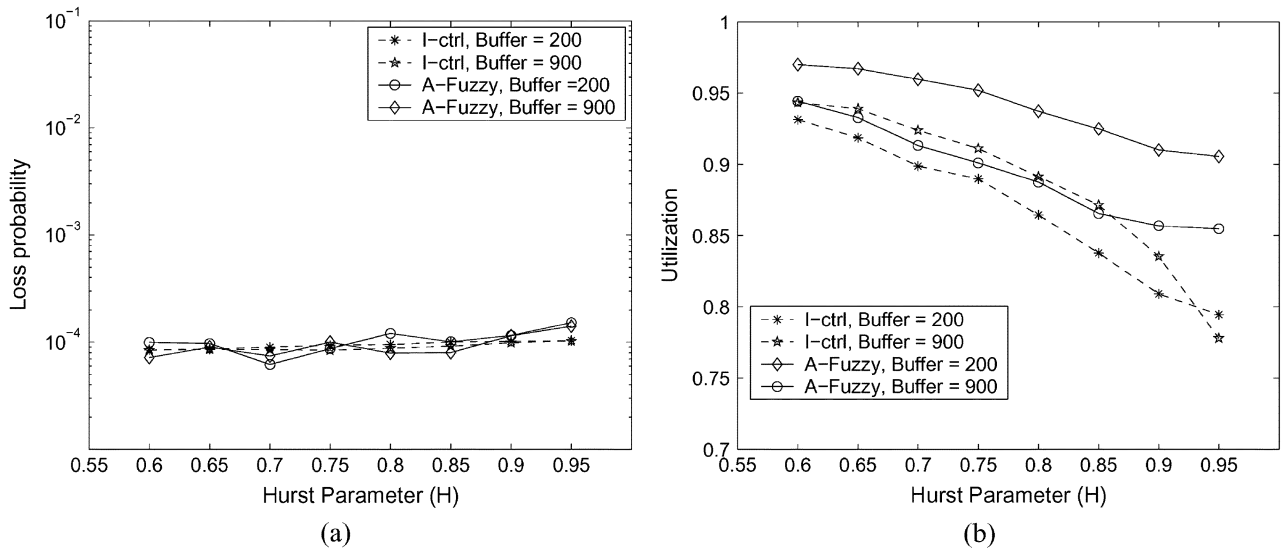


Fig. 14. Performance comparison between the I-control and the A-Fuzzy control over a range of Hurst parameter $H(\varepsilon = 10^{-4})$.

and 900 packets. We can see that the A-Fuzzy control provides loss performance as good as the I-control does, but with higher utilization. Note that both the A-Fuzzy control and the I-control still yield a loss rate that somewhat deviates from the target at high values of a. In those cases, it has been found that the TQL becomes very close to zero, and the average queue length cannot be effectively maintained at such a small target queue length. Therefore, the achievable performance of both the A-Fuzzy control and the I-control are limited when the input traffic is heavily bursty. Fig. 14 plots the loss rate and utilization performance for Experiment II at the buffer size of 200 and 900 packets. Similar to Experiment I, the A-Fuzzy control can achieve the target loss rate with better utilization than the I-control. Note that in Experiment II, both controls effectively achieve the target loss rates over a wide range of H, implying that traffic correlation has less impact on the control performance than burstiness does.

V. Admission Control in Single Queue

Thus far, we have considered only ABC in a single queue with no constraint on the capacity to allocate. In real networks, there exists a limit on the bandwidth resources imposed by the maximum link capacity or some administrative policy. This capacity limitation necessitates the use of admission control on the input traffic to ensure that the bandwidth is sufficient to maintain or achieve the required loss performance. In case of SBA, the amount of required bandwidth is calculated at the instant of connection arrivals. A newly arriving aggregate flow is admitted only if the sum of the required bandwidth of the arriving flow and those of existing flows does not exceed the available bandwidth. Such an admission control procedure is not directly applicable to ABC because the allocated bandwidth under ABC is time-varying by its very nature. We propose that the admission control problem under ABC could be solved by decomposing the problem into two steps.

- The first step is to investigate the relationship between the degree of QoS degradation and the bandwidth violation due to a limited link capacity. With such knowledge available, we can quantify how much bandwidth violation ABC can tolerate while retaining an acceptable level of QoS.
- The second step is to discover the impact of admitting new aggregates with respect to the bandwidth violation.

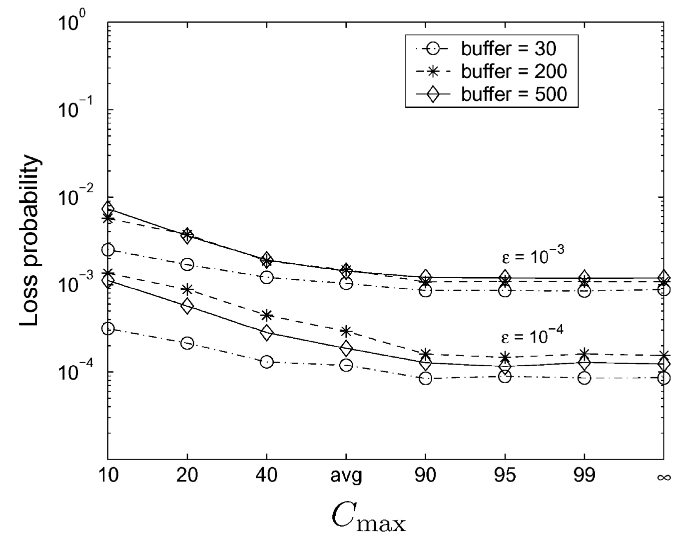


Fig. 15. Loss rate under limited link capacity for fGn traffic ($\lambda = 10000, a = 1, H = 0.85$).

By combining these two steps, the admission control in ABC can be exercised successfully.

A. Impacts of Limited Link Capacity

Denote C_u as a random variable representing the allocated bandwidth under no capacity constraint, i.e., $C_{\max} = \infty$. We start by determining the empirical distribution of C_u from 10 runs. Then, different percentiles of C_u as well as its average value are used as $C_{\max}$. Fig. 15 plots the packet loss rates versus $C_{\max}$ for the A-Fuzzy control under fGn traffic. We can see that minor bandwidth violation has no severe impact on the loss performance. Setting $C_{\max}$ to the average of C_u does not significantly increase the packet loss rate. In most cases, we note that

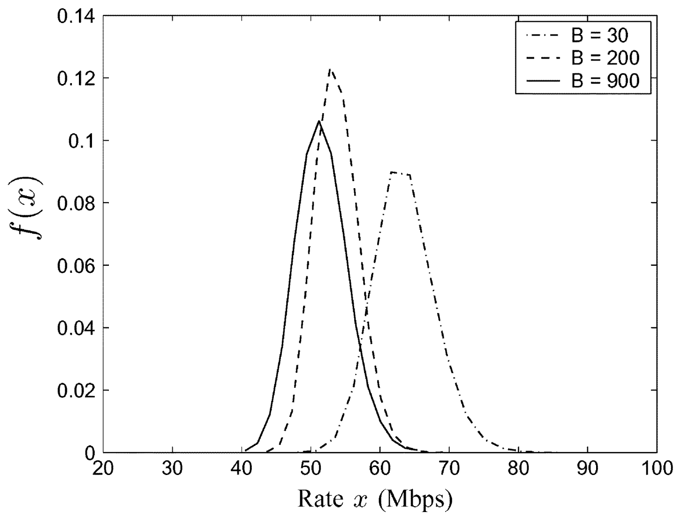


Fig. 16. A-Fuzzy Control: Empirical distributions of allocated bandwidth under no capacity constraints for a multiplex of 30 Trace-1 traces ($\varepsilon = 10^{-4}$).

the loss performance is acceptable even if $C_{\max}$ decreases to the 90th percentile of C_u.

B. Relating Input Traffic to Bandwidth Violation

We have seen previously that ABC can tolerate a small degree of bandwidth violation without noticeable QoS degradation, e.g., the 95th percentile of C_u should be less than $C_{\max}$. Therefore, we only have to admit traffic such that the allocated bandwidth by ABC will stay below $C_{\max}$ with a high probability. Given the required information on the input traffic (obtained from either user declaration or from online measurement), we relate them to the statistical behavior of the allocated bandwidth and then determine the bandwidth violation probability.

Fig. 16 shows the empirical statistical distribution of C_u (obtained from a sample path of C_u from 5 runs) for a multiplex of 30 Trace-1 traces (described in Section IV). The similar behavior is also obtained for the case of fGn traffic and an aggregate of Parato ON-OFF sources (not shown here due to space limitation). Therefore, for all the traffic types considered, the assumption that C_u is approximately normally distributed is justified, even though the underlying traffic generation process of each source is non-Gaussian. This result can be explained intuitively by the fact that the input traffic rate process on the aggregate level is approximately normal [20], [21], and the A-Fuzzy control allocates a bandwidth that closely tracks the input traffic. According to the previous observation, we model C_u due to the A-Fuzzy control with a normal distribution having a mean μ and a variance σ^2. Note that even if C_u may not be exactly normal, we argue that some degree of model accuracy can be traded off for simplicity.

Next, we relate the mean rate λ' of a new aggregate to C_u. Let ρ be the estimated bandwidth utilization of the queue at the time of the aggregate arrival. We estimate that the new aggregate would require a bandwidth of

$$\mu' = \lambda'/\rho. \tag{10}$$

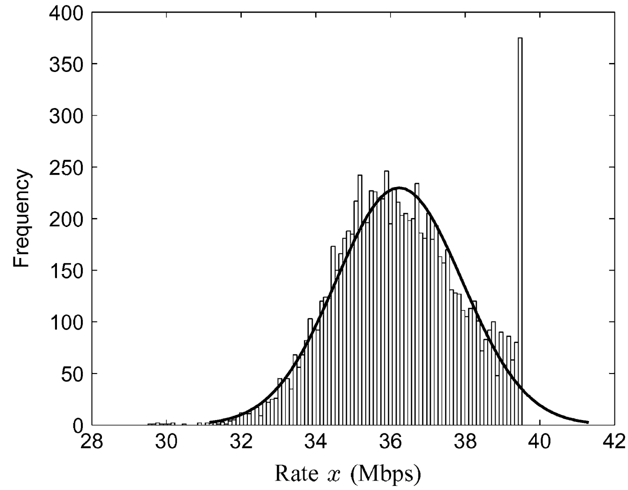


Fig. 17. Plot of a normal distribution superimposed on the histogram of C_k for fGn traffic (Buffer = 500 packets, $\varepsilon = 10^{-4}$).

Note that (10) underestimates μ' if the new aggregate causes the overall traffic to the queue to have higher burstiness, and overestimates μ' otherwise. Assuming that σ^2 does not change significantly after the new aggregate has been admitted, the estimated allocated bandwidth due to the addition of the new aggregate is normally distributed with mean $\mu + \mu'$ and variance σ^2.

C. Admission Control Algorithm

We define the risk of bandwidth violation (ζ) as the probability that C_u due to the A-Fuzzy control exceeds the limit $C_{\max}$. Then

$$\begin{aligned}\zeta &= \mathbb{P}\{C_u > C_{\max}\}\\ &= \mathbb{P}\left\{Z > \frac{C_{\max} - (\mu + \mu')}{\sigma}\right\}\end{aligned}$$

where Z is a standard normal variate. The value of ζ to be used as a requirement must be selected carefully. The results for the limited link capacity from the previous chapter show that ABC can easily tolerate a bandwidth limit $C_{\max}$ equal to the 95th percentile of C_u. Consequently, we suggest that $\zeta = 0.05$ should be sufficient. At $\zeta = 0.05, \Phi^{-1}(0.05) = 1.645$, the admission control test to admit the new traffic aggregate becomes

$$C_{\max} > (\mu + \mu') + 1.645\sigma. \tag{11}$$

The last question is whether μ and σ^2 that are estimated from online measurement are accurate enough to be used when the controller operates in a limited link capacity. Fig. 17 plots parameterized normal distributions superimposed on the histogram of the allocated bandwidth under a limited link capacity for fGn traffic ($C_{\max}$ =95th percentile of C_u). The figure reveals that even though the mean and variance of the fitted distribution are parameterized from the allocated bandwidth under the limited capacity case, the histogram appears as if they are a truncated version of the fitted distribution. Table II shows the mean and variance of the allocated bandwidth under the limited link capacity for the previous case, together with the percentiles of the fitted distribution as well as the actual bandwidth limits $C_{\max}$. We see that the percentiles of the fitted distribution closely match the actual $C_{\max}$. However, we found

TABLE II
PARAMETERS OF THE FITTED NORMAL DISTRIBUTION AND ITS PERCENTILE (IN MB/S) UNDER A LIMITED LINK CAPACITY

Traffic	Mean	Variance	Percentile from fitted distribution	Actual $C_{\max}$
fGn ($C_{\max} = 90^{th}$ percentile on C_u)	35.98	1.64	38.08	38.77
fGn ($C_{\max} = 95^{th}$ percentile on C_u)	36.24	1.69	39.02	39.52

that this is true only if $C_{\max}$ is set at high percentiles (90th or 95th) of C_u, which fortunately is within the range that we are concerned with. Consequently, we can estimate μ and σ of C_u from the allocated bandwidth under a limited link capacity.

The admission control procedure for a single queue case with the capacity limit $C_{\max}$ is summarized as follows. The values of $\mu = \mathbb{E}(C_k), \sigma = \sqrt{\text{var}(C_k)}$, and the bandwidth utilization ρ are obtained through online measurement and reset after a new aggregate is admitted or an existing aggregate departs the queue. At each aggregate arrival instant where the aggregate declares its mean rate of λ', we determine the average bandwidth μ' required by the new aggregate from (10). Using μ, σ, and μ' obtained above, we check to see if (11) holds. If not, the new aggregate is rejected. Otherwise, the new aggregate is admitted to the queue.

D. Simulation Results

In order to investigate the admission control algorithm performance, a preliminary simulation study under various heterogeneous traffic conditions is conducted. In the simulation model, the link capacity $C_{\max}$ is set to 155 Mb/s. The aggregate connection arrival process follows a Poisson distribution with mean rate λ_c connections per second and exponential holding time of $1/\mu_c = 900$ seconds. The offered load to the queue is controlled by varying λ_c. We set λ_c/μ_c to 15, the buffer size to 900 packets and the target loss rate is 10^{-4}. The offered load λ_c/μ_c is set to a high value so that we can obtain a relatively accurate blocking rate.[5] The following experiments are conducted.

- **Exp. I: Heterogeneous fGn** Each aggregate is fGn with different parameters. To model traffic heterogeneity, the parameters of each aggregate were randomly selected from uniform distributions:$\lambda = U(1, 16)$ Mb/s, $a = U(1, 10)$, and $H = U(0.8, 0.95)$. The performance is measured by the connection blocking rate and the packet loss rate performance. The simulation time lasts 60 000 s and $\zeta = 0.05$ is used.
- **Exp. II: Heterogeneous fGn+Traces** In this case, there are two types of aggregates. The first type is of heterogeneous fGn traffic described in Experiment I. The second type is constructed from Trace-1 (from Section IV), where each aggregate is a group of Trace-1 traces where the number of traces in the group is randomly drawn from $U(1, 10)$. Each trace in the aggregate starts at a uniformly distributed time in the trace and wraps around until the connection holding time is reached. We set the connection arrival rate of each aggregate type to 8 so that both traffic types contribute an equal amount of load.
- **Exp. III: Traces** In this case, only Trace-1 is used. Each aggregate is constructed from Trace-1 the same way as described in Experiment II.

Fig. 18 plots the sample paths of the allocated bandwidth and the cumulative loss rate for Exp. II (The plots for those of Exp. I and Exp. II are similar, so we show only the results of Exp. II here.) We can see that the packet loss rate can be satisfactorily maintained at the target under the proposed admission control independent of whether the input traffic is fGn or comes from a mix among different heterogeneous traffic types. For Exp. I, the connection blocking rate is 20.5%. Under the same simulation setting, using the MVA approximation (which is the most accurate static allocation method developed in the literature so far) yields a connection blocking rate of 42.7%. The link utilization (an average of the allocated bandwidth over the link capacity) is 0.84 for the A-Fuzzy control and 0.92 for the MVA approximation. This means that the A-Fuzzy control not only accepts more aggregates but also uses less link bandwidth than the MVA approximation. For Exp. II, the connection blocking rate is 21.8%. For Exp. III, the connection blocking rate is 12.6%. The lowest connection blocking rate in Exp. III suggests that the allocated bandwidth under the input traffic constructed from Trace-1 has the lowest variance. This result implies that although an individual Trace-1 trace has a very high rate variance as shown in Fig. 6, multiplexing a large number of Trace-1 traces results in traffic that is smoother than fGn traffic whose parameters are selected according to Exp. I.

VI. ADMISSION CONTROL IN MULTIPLE QUEUES

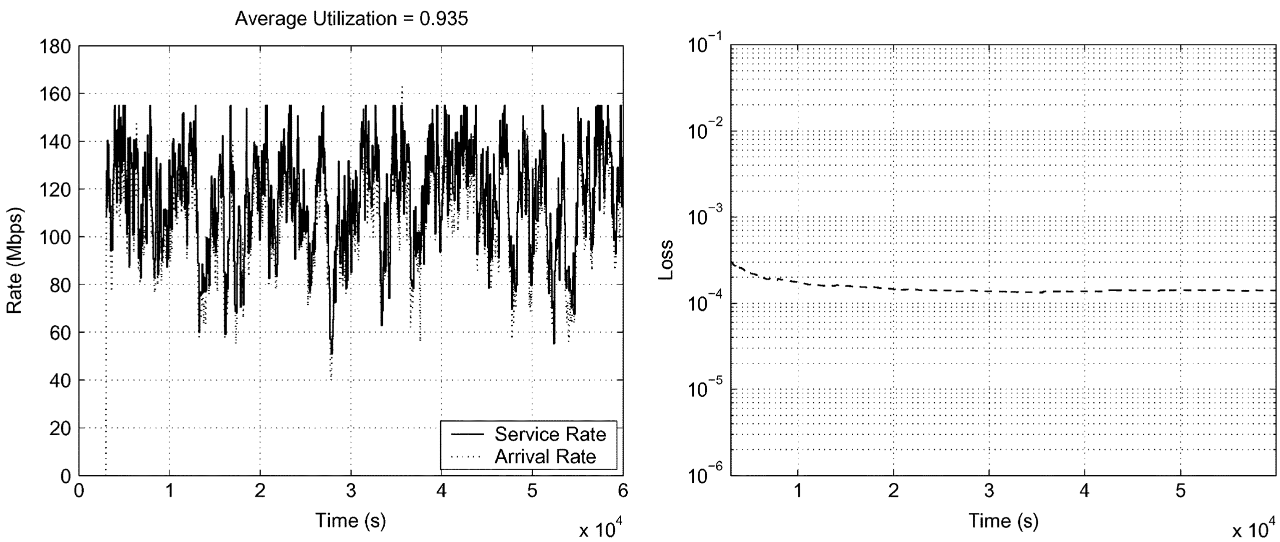
In general, we have a multiple queue scenario as shown previously in Fig. 1. In each queue, aggregate flows arrive and depart over time. The dynamic of aggregate arrival process can represent the aggregate merging in the sink-tree model [13], whereby many aggregates can be merged at some node in the network and are carried in the same path to the destination. The merging is equivalent to admitting a new aggregate into the queue. The number of queues can be fixed or varied.

The constraint for bandwidth allocation in the multiple queue case is $\sum_{i=1}^{K} C_i(t) < C$. To handle this constraint, we define a global variable Σ and assign its initial value as $\sum_{i=1}^{K} C_i(0)$. Then, at a bandwidth reallocation instant t' for queue i where the bandwidth controller has calculated the amount of change in the allocated bandwidth $\Delta C_i(t')$ (which can be either positive or negative), the controller first checks if $C - \Sigma > \Delta C_i(t')$. If so, the bandwidth adjustment of $\Delta C_i(t')$ is granted to queue i and Σ is updated to $\Sigma + \Delta C_i(t)$. Otherwise, the bandwidth adjustment of $C - \Sigma$ is granted to queue i and Σ is updated to C. Each queue can perform the previous process independently. In cases of more than one queue trying to update Σ simultaneously, the order of updates is chosen randomly.

A. Fixed Number of Queues

Each queue i will have an independently allocated capacity limit $C_{\max}^i$, which is assigned according to the blocking probability constraints on the connection level. Also, each queue has its own bandwidth controller and admission control module that operate independently. Therefore, the admission control procedure for a single queue developed in the previous section is directly applicable. Particularly, each queue i is assigned a hard

[5]90% confidence intervals at the relative precision of less than 10% are obtained in each experiment.



(a) (b)

Fig. 18. Experiment II: Allocated bandwidth and loss rate of the A-Fuzzy control under a mixed of heterogeneous fGn traffic and real traces with a limited link capacity and admission control.

capacity limit $C^i_{\max}$ such that $\sum \overline{C}^i \max \leq C$, which allows us to apply the admission control in a single queue case previously developed to each queue independently. However, one minor drawback to this approach is that it does not consider the correlation of the allocated bandwidth among the queues. For example, there can be a time when the allocated bandwidth of queue i reaches its capacity limit while the available capacity for queue j is large, but cannot be used by queue i. By allowing the available bandwidth for queue j to be used by queue i, we expect that a better loss rate performance could be achieved because there would be less frequent bandwidth violations. However, the performance improvement may be only marginal because the probability of allowable bandwidth violation (ζ) according to the admission control for a single queue is already small.

B. Varying Number of Queues

In this case, queues varied by being dynamically created and destroyed respective to each aggregate arrival and departure, and each queue will be relatively long-lived compared to the time scale of bandwidth adjustment. This scenario can represent a VPN application, whereby a new queue is set up in response to the request of some particular VPN user. There may not be prespecified hard capacity limits for each queue. As such, the connection blocking probability cannot be guaranteed to each queue. According to the queue dynamics, there are two levels of admission control. The first level is the admission control to admit new aggregates to the existing queues, referred to as the *aggregate-level* admission control. The second level is the admission control for a request to set up a new queue, referred to as the *queue-level* admission control. We describe here only the aggregate-level admission control. The queue-level case is outlined in [8].

Let $\mathcal{N}$ be a set of queues currently set up on the link with capacity C and $N = |\mathcal{N}|$. We can assume that the bandwidth $C^{(i)}$ allocated to queue i is approximately normally distributed with a mean μ_i and a variance σ_i^2. Next, we relate the mean rate

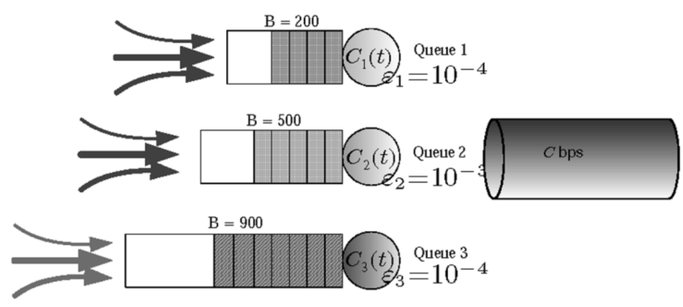
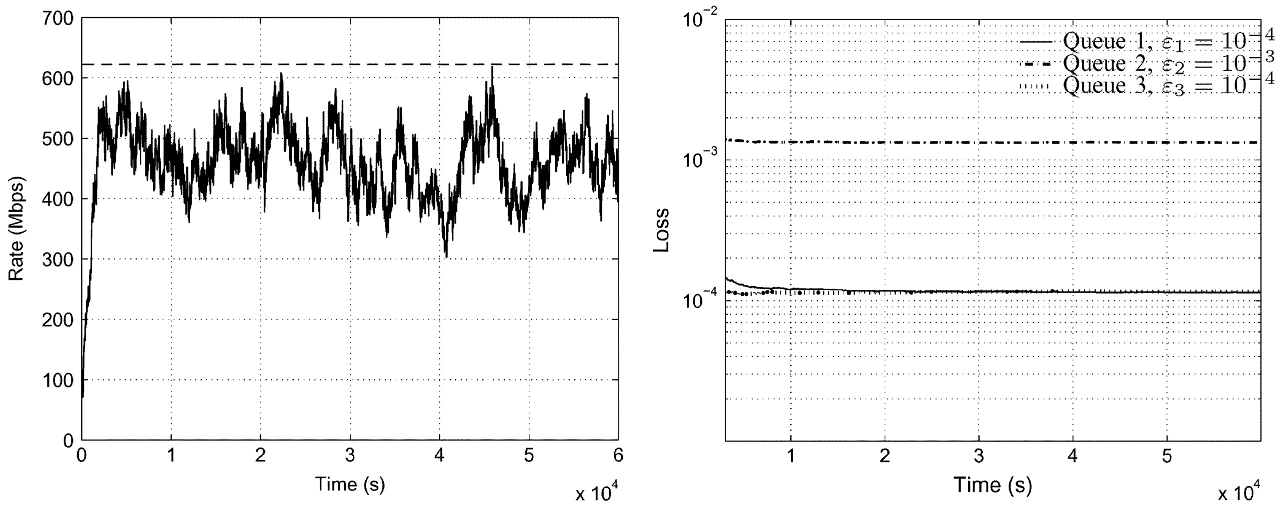


Fig. 19. Simulation setting for the multiple queue case.

λ_i' of a new aggregate to $C^{(i)}$ by using an approach similar to that described in Section V. Let ρ_i be the estimated bandwidth utilization of queue i at the time of the aggregate arrival. We estimate that the new aggregate would require the amount of bandwidth of $\mu_i' = \lambda_i'/\rho_i$ and, thus, the minimum capacity limit required by queue i is $(\mu_i + \mu_i') + 1.645\sigma_i$. The idea behind the admission control here is that the sum of the minimum capacity limits required by each queue must not exceed the link capacity after the new aggregate is admitted. The minimum total capacity limit required by all the other queues $j \neq i$ is $\sum_{j \in \mathcal{N}, j \neq i} \mu_j + 1.645\sigma_j$. Therefore, the admission control condition is

$$C > \mu_i' + \sum_{i \in \mathcal{N}} (\mu_i + 1.645\sigma_i). \tag{12}$$

Under the admission control (12), the statistical multiplexing gain is lost because the minimum capacity limit is computed independently. Observe that the total allocated bandwidth at any time instant is the sum of $C^{(i)}$ and, hence, normally distributed with mean $\mu = \sum_{i=1}^{N} \mu_i$ and variance $\sigma^2 = \sum_{i=1}^{N} \sigma_i^2$. We thus estimate that a new aggregate arrival to queue i would require the amount of bandwidth of $\mu_i' = \lambda_i'/\rho_i$ as described earlier. Then, the total allocated bandwidth is approximately normally distributed with mean $\mu + \mu_i'$ and variance σ^2. This leads to the



(a) (b)

Fig. 20. Allocated bandwidth and loss rates in a three-queue scenario with admission control ($C = 622$ Mb/s).

admission control condition that takes into account a statistical multiplexing gain

$$C > \mu_i' + \sum_{i\in\mathcal{N}} \mu_i + 1.645\sqrt{\sum_{i\in\mathcal{N}} \sigma_i^2}. \tag{13}$$

To investigate the performance of this admission control scheme, a simulation study of three queues sharing the same link is conducted, as depicted in Fig. 19. The loss rate guaranteed by Queue 1 and Queue 3 is 10^{-4}, and 10^{-3} for Queue 2. Each queue also has different buffer sizes (200, 500, and 900 packets) so that the variance of the allocated bandwidth would be different, i.e., the queue with the higher buffer size will have the higher variance. The link capacity C is set to 622 Mb/s. The aggregate connection arrival to each queue follows a Poisson process with a mean rate λ_c connections per second and a holding time $1/\mu_c$ of 900 s. The offered load to each queue is controlled by varying λ_c. Each aggregate is fGn traffic. Traffic heterogeneity is modeled by randomly selecting the parameters of each aggregate from a uniform distribution $-\lambda = U(1, 16)$ Mb/s, $a = U(1, 10)$, and $H = U(0.8, 0.95)$. The performance is measured by the connection blocking rate and the packet loss rate performance of each queue. The simulation time lasts 60 000 s and $\zeta = 0.05$ is used.

Fig. 20 plots the sample path of the total allocated bandwidth for $\lambda_c/\mu_c = 15$ and the cumulative loss rate of each queue. We can see that the target loss rates are met in every queue. Furthermore, the result indicates that the admission control is efficient because the total allocated bandwidth stays sufficiently close to the link capacity (622 Mb/s) and, hence, has a high link utilization. The connection blocking rates in this case are 1.4%, 1.2%, and 0.8%, respectively. Under the MVA approximation for admission control, the connection blocking rate of each queue is approximately 10%.

VII. Conclusion

Existing bandwidth allocation techniques to guarantee a packet loss rate to aggregate traffic require detailed statistical information of the traffic, which is inconvenient and frequently infeasible for users to come by. We considered the use of ABC, which dynamically adjusts the allocated bandwidth based on state information obtained from online measurement to achieve a given loss rate requirement. Therefore, no prior detailed traffic information is required. An ABC algorithm based on fuzzy control, called A-Fuzzy control is proposed. The algorithm uses fuzzy control to keep the average queue length at some appropriate target value such that the desired loss rate would be achieved, and also incorporates the measured packet loss information to correct the allocated bandwidth at a smaller time scale to cope with undesirable packet losses. Our extensive performance evaluation on different traffic types, buffer sizes, and dynamic conditions, has shown that the A-Fuzzy control provides better utilization than existing static allocation methods in most cases. In addition, the A-Fuzzy control is highly robust in the sense that the same set of control parameters from intuitive settings works well in most cases considered, including both theoretical traffic models and traffic traces from operational IP networks. Also, based on the normal approximation of the allcated bandwidth, we developed a simple and efficient measurement-based admission control procedure for both single queue case and multiple queue case. Simulation results have shown that high link utilization under such admission control procedures is attained.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, Dec. 1998. rFC 2475.

[2] D. Anick, D. Mitra, and M. Sondhi, "Stochastic theory of a data-handling system with multiple sourcees," *Bell Syst. Tech. J.*, vol. 61, pp. 1261–1281, 1982.

[3] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 968–981, Sep. 1991.

[4] C. Courcoubetis, V. Siris, and G. Stamoulis, "Application and evaluation of large deviation techniques for traffic engineering in broad-band networks," *Proc. ACM SIGMETRICS'98*, pp. 212–221, 1998.

[5] P. Siripongwutikorn, S. Banerjee, and D. Tipper, "A survey of adaptive bandwidth control algorithms," *Commun. Surveys Tuts.*, vol. 5, no. 1, 2003.

[6] I. Hsu and J. Walrand, "Dynamic bandwidth allocation for atm switches," *J. Appl. Prob.*, vol. 33, no. 3, pp. 758–771, 1996.

[7] S. Rampal, D. Reeves, Y. Viniotis, and D. Argrawal, "Dynamic resource allocation based on measured qos," Center Adv. Comput. Commun., North Carolina State Univ., Raleigh, Tech. Rep. TR 96-2, Jan. 1996.

[8] P. Siripongwutikorn, "Adaptive bandwidth control for efficient aggregate qos provisioning," Ph.D. dissertation, Dept. Inf. Sci. Telecommun., Univ. Pittsburgh, Pittsburgh, PA, Sep. 2003.

[9] R. R.-F. Liao and A. Campbell, "Dynamic edge provisioning for IP networks," *Proc. IEEE/IFIP IWQoS'00*, pp. 148–157, Jun. 2000.

[10] S. Chong, S.-Q. Li, and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time vbr video over atm," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 1, pp. 12–23, Jan. 1995.

[11] M. Grossglauser, S. Keshav, and D. Tse, "Rcbr: A simple and efficient service for multiple time-scale traffic," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 741–755, Dec. 1997.

[12] A. Adas, "Using adaptive linear prediction to support real-time vbr video under rcbr network service model," *IEEE/ACM Trans. Netw.*, vol. 6, no. 5, pp. 635–645, Oct. 1998.

[13] N. Duffield, P. Goyal, A. Greenberg, K. Ramakrishnan, and J. van der Merwe, "A flexible model for resource management in virtual private networks," in *Proc. ACM SIGCOMM'99*, Cambridge, MA, Oct. 1999, pp. 95–108.

[14] Z. Sahinoglu and S. Tekinay, "A novel approach bandwidth allocation: Wavelet-decomposed signal energy approach," in *Proc. IEEE GLOBECOM'01*, San Antonio, TX, Dec. 2001, pp. 2253–2257.

[15] M. Grossglauser and J.-C. Bolot, "On the relevance of long-range dependence in network traffic," in *Proc. ACM SIGCOMM'96*, Palo Alto, CA, 1996, pp. 1–10.

[16] E. Fulp and D. Reeves, "Online dynamic bandwidth allocation," in *Proc. Int. Conf. Network Protocols*, Atlanta, GA, Oct. 1997, pp. 134–141.

[17] V. Paxson, "Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic," *Comput. Commun. Rev.*, vol. 27, no. 5, pp. 5–18, Oct. 1997.

[18] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 953–962, Aug. 1995.

[19] H. Kim and N. Shroff, "Loss probability calculations and asymptotic analysis for finite buffer multiplexers," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 755–768, Dec. 2001.

[20] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning ip backbone networks to support latency sensitive traffic," in *Proc. IEEE INFOCOM'03*, San Francisco, CA, Mar. 2003, pp. 375–385.

[21] N. Duffield, W. Massey, and W. Whitt, "A nonstationary offered-load model for packet networks," *Telecomm. Syst.*, vol. 16, no. 3–4, pp. 271–296, 2001.

Peerapon Siripongwutikorn (M'03) received the M.S. and Ph.D. degrees in telecommunications from the University of Pittsburgh, Pittsburgh, PA, in 1998 and 2003, respectively.

He is currently with the Department of Computer Engineering, King Mongkut's University of Technology Thonburi, Thailand. His current research interests include ad hoc networks, dynamic resource allocation and control in communication networks, and network performance analysis.

Sujata Banerjee (S'89–M'93–SM'99) received the B.S. and M.S. degrees in electrical engineering from the Indian Institute of Technology, Bombay, India and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles.

She is a Senior Scientist and R&D Project Manager of a networking research group in Hewlett-Packard Laboratories, Palo Alto, CA. Prior to joining HP, she was an Associate Professor of Telecommunications at the University of Pittsburgh. Her research interests are in quality of service issues in networked systems.

Dr. Banerjee is a recepient of the National Science Foundation CAREER award in Networking Research. She has served as an Associate Editor for the IEEE TRANSACTIONS ON RELIABILITY as well as on the technical program committee of several conferences and workshops.

David Tipper (S'78–M'88–SM'95) received the B.S.E.E. degree from the Virginia Polytechnic Institute and State University, Blacksburg and the M.S.S.I.E and Ph.D.E.E. degrees from the University of Arizona, Tuscon.

He is an Associate Professor in the Department of Information Science and Telecommunications, University of Pittsburgh. Prior to joining the University of Pittsburgh in 1994, he was an Associate Professor of Electrical and Computer Engineering at Clemson University, Clemson, SC. His current research interests are network design, traffic restoration procedures for survivable networks, network control (i.e., routing, flow control, etc.), and performance analysis. He is currently a member of the editorial board of the *Journal of Network and Systems Management*.

Dr. Tipper has been on numerous conference technical commitees including serving as the Technical Program Chair of the Fourth IEEE International Workshop on the Design of Reliable Communication Networks (DRCN 2003).