THIRD QUARTER 2003, VOLUME 5, NO. 1

IEEE COMMUNICATIONS SURVEYS

The Electronic Magazine of
Original Peer-Reviewed Survey Articles

www.comsoc.org/pubs/surveys

A SURVEY OF ADAPTIVE BANDWIDTH CONTROL ALGORITHMS

PEERAPON SIRIPONGWUTIKORN, UNIVERSITY OF PITTSBURGH
SUJATA BANERJEE, HEWLETT-PACKARD LABORATORIES
DAVID TIPPER, UNIVERSITY OF PITTSBURGH

ABSTRACT

In packet-switched network traffic management and control, efficiently allocating bandwidth to provide quantitative packet-level QoS to aggregate traffic has been difficult due to unpredictable, unknown statistical characteristics of the aggregate traffic. With inaccurate traffic information, using static bandwidth allocation results in the network being underutilized, or the QoS requirement not being satisfied. An alternative is to use Adaptive Bandwidth Control (ABC), whereby the allocated bandwidth is regularly adjusted over the packet-level time scale to attain a given QoS requirement. This paper provides a literature review of ABC algorithms that guarantee aggregate traffic packet-level QoS metrics, such as the average queue length, packet loss, and packet delay. We describe different ABC algorithms, identify their advantages and shortcomings, and provide some comparative performance evaluation results. Open issues in ABC for future research directions are also discussed.

In the past decade, we have witnessed the evolution of IP networks from providing a single best-effort service to providing multiple types of services with Quality-of-Service (QoS) guarantee, namely bounded packet delay and loss. Although the majority of current network traffic is still best-effort, emerging networked applications with different QoS constraints (e.g., multimedia applications) are becoming more popular, and may soon contribute an appreciable share of total network traffic. In addition, users are demanding better QoS guarantees for their traffic, or QoS differentiation for their virtual private networks (VPNs). The Integrated Service (IntServ) framework is an attempt to achieve *end-to-end per-flow* QoS is, whereby bandwidth along the path is reserved for individual flows through RSVP. IntServ was unsuccessful in large-scale networks mainly due to scalability and heterogeneity concerns. In particular, the amount of per-flow states is prohibitively large, and all nodes along the end-to-end path must implement the same reservation protocol. An alternate attempt to solve the challenges of providing QoS is the Differentiated Service (DiffServ) framework, proposed by the IETF, which delivers a coarse level of QoS on a *per-node*, *per-aggregate* basis such that scalability is preserved. DiffServ only provides relative or qualitative QoS differentiation such as high bandwidth, low delay, or low loss by allocating more bandwidth to one aggregate than another, or by implementing dropping preferences among aggregates. DiffServ together with Multiprotocol Label Switching (MPLS) provides a powerful and highly scalable framework for QoS provisioning in IP networks: MPLS controls the data path while DiffServ controls the QoS differentiation.

A component missing from current approaches to aggregate-based traffic management and control such as DiffServ is the capability to guarantee quantitative aggregate QoS, which provides bounded delay and packet loss. In the flow level, quantitative QoS guarantees have typically been accomplished by static allocation methods, which assume some stochastic properties of the traffic arrivals (see, e.g., [1–3]). However, such static allocation is inefficient for traffic in the aggregate level for the following reasons:

- The underlying traffic model may not completely or accurately capture the statistical behavior of the actual traffic. For example, a heterogeneous mix of aggregates with different statistical properties can result in a traffic stream whose characteristics are unknown. Furthermore, even if the traffic characterization is known at the network ingress, it can be altered after passing through multiplexers and buffers in the network.
- The user-declared traffic parameters may not accurately represent the actual traffic.
- Traffic parameterization is not possible when the traffic is generated on-the-fly, e.g., from video conferencing or sporting events.

In per-flow traffic management and control, the latter two issues have been partly addressed by Measurement-Based Admission Control (MBAC), whereby the measured values of traffic parameters, e.g., the mean arrival rate and the rate variance, and QoS of the ongoing traffic are used respectively in conjunction with the user-declared parameters and analytical models to improve the accuracy in estimating the required bandwidth (see, e.g., [4]). For example, in [5, 6] the measured statistics of the ongoing aggregate process, such as the mean aggregate rate and variance, are adjusted when a flow arrives or leaves the aggregate to take into account the effect of flow arrival or departure. These adjusted statistics are then substituted into a specific equivalent bandwidth (EB) formula to estimate the bandwidth needed to guarantee the required QoS. Even if there is no flow arrival or departure, such input traffic statistics can also be measured regularly to update the required bandwidth. Using the measurement information significantly reduces the error in bandwidth estimation due to inaccurate user-declared traffic parameters and analytical models. However, the per-flow traffic information within each aggregate required by these MBAC frameworks is not available in the framework of aggregate traffic management and control. In addition, the accuracy of MBAC is questionable given the strong assumptions of some EB formula and traffic models, which may be further intensified by measurement error. An MBAC that requires no per-flow arrival information is developed in [7], but the QoS metric is the probability that the instantaneous bandwidth demand exceeds the link capacity in a bufferless multiplexer, not the packet loss probability in a buffered system.

Due to the aforementioned difficulties in providing quantitative QoS guarantees, many researchers have instead focused on improving the quality of best-effort (BE) service by providing a fair share of bandwidth and preventing congestion (see [8] for a comprehensive survey on the subject). The basic idea is to use scheduling mechanisms and/or queue management schemes for the following functions:

- To prevent misbehaved or aggressive UDP flows from receiving more bandwidth than their fair share.
- To implicitly or explicitly notify the sources (end-hosts) of TCP flows to reduce their sending rates at the onset of congestion rather than waiting for packet loss to occur due to queue overflows.

For example, in the Active Queue Management (AQM) framework, the router intentionally drops packets upon arrival with a probability that increases with the average queue length (see, e.g., [9, 10]). The packet loss will act as an implicit feedback signal of congestion events to senders (end-hosts). Improving the quality of the BE service is appealing because the vast majority of traffic is BE and some QoS applications, e.g., streaming multimedia and VoIP, can tolerate mild congestion in the network by adapting their transmission rates. However, during the period of transient congestion or network overload, fairness obviously does not imply QoS, which becomes worse under sustained congestion. Therefore, we believe that there is still the need to provide quantitative QoS because of inconsistency and unpredictability of QoS experienced in the BE service.

An alternative to static bandwidth allocation for attaining (per-node) aggregate QoS with simplicity and efficiency is Adaptive Bandwidth Control (ABC). Under static bandwidth allocation, the required bandwidth calculated upon each aggregate traffic flow arrival is reserved at each node (using some protocol such as RSVP). In contrast, ABC starts with some initial amount of bandwidth allocation to the queue and adjusts it over time in the packet-level time scale, e.g., on the order of seconds, to ensure that the allocated bandwidth is just enough to attain the specified QoS requirements. As a result, less bandwidth will be wasted due to over-allocation, thus improving network utilization. Because ABC algorithms obtain information needed for the control through online measurement of system states, they do not have the deficiencies found in static bandwidth allocation, particularly:

- No assumption on the input traffic model is required.
- The impact on performance of any inaccuracy in the user-declared traffic parameters will be reduced because the control will adapt over time to actual traffic conditions.
- Since the traffic model is not required, there is no need for *a priori* traffic parameterization.

ABC is the way we envision how quantitative QoS should be guaranteed: users simply supply their QoS requirements (and possibly very minimal traffic information such as the aggregate average rate) without having to know which traffic model to use.

The objective of this survey paper is twofold. First, we provide a literature review of existing ABC algorithms that guarantee packet-level QoS of aggregate traffic, including the average queue length, the packet loss, and the packet delay. Although there is extensive literature on ABC in ATM networks that adapts the bandwidth of virtual paths at a connection-level time scale to satisfy the connection-blocking probability (see, e.g., [11]), our focus here is on the packet-level ABC, where the bandwidth allocated to a traffic aggregate is adapted at a packet-level time scale to satisfy packet delay or loss requirements. Therefore, ABC to guarantee the connection blocking probability QoS metric, mostly developed in the ATM related literature, is not in the scope of this paper.

The second objective is to discuss the performance tradeoff of ABC algorithms as well as open issues related to the practical deployment of ABC in networks. The next section describes the system model considered in the operating context of ABC, and then classifies ABC algorithms. Then various ABC algorithms proposed in the literature are presented in order according to the type of QoS metrics guaranteed. Their advantages and drawbacks are identified. Issues that are unaddressed by the previous research in ABC will be discussed later, including the control time scale, admission control, and control in multiple queues. The paper concludes with a summary and perspectives on different ABC algorithms.

PRELIMINARIES

MODEL

Consider the output port of an output-queued switch, where the port has an outgoing link capacity of C b/s. The port is assumed to support K QoS traffic classes and one BE traffic class, each of which has its own logical queue, as shown in Fig. 1. We consider the problem of QoS guarantee for traffic in each QoS queue by properly allocating the amount of bandwidth $C_i(t)$, $i = 1, 2, \ldots, K$, with the constraint that

$$\sum_{i=1}^{K} C_i(t) \le C.$$

Any unallocated bandwidth

$$C - \sum_{i=1}^{K} C_i(t)$$

is consumed by the BE traffic. This model describes the DiffServ/MPLS network framework, in which QoS is differentiated on a per-node, per-class basis. It can also be used to describe a Virtual Private Network (VPN) environment, in which each queue is dedicated to aggregate traffic belonging to a particular VPN user. For a more complicated scenario, each VPN user may be assigned a fixed bandwidth pool which supports multiple traffic classes to provide service differentiation to traffic within the same VPN user. According to the model, the QoS guarantee by means of ABC can be achieved independently in each queue, with the constraint on the out-

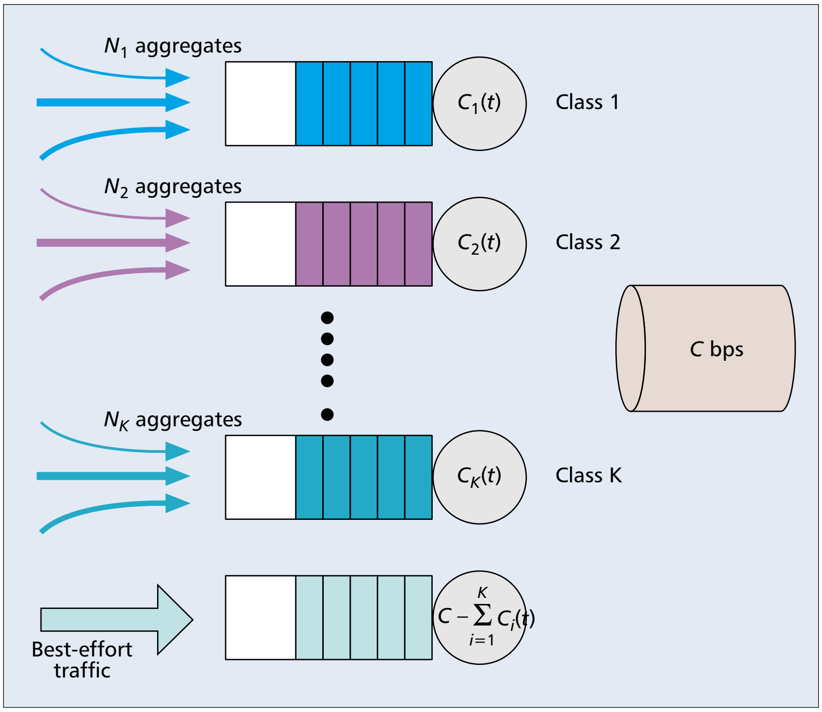


FIGURE 1. *The transmission link system model.*

put link capacity as mentioned before. Therefore, we focus our discussion of ABC on the case of a single queue, and then later describe issues that arise due to the limited link capacity and multiple queues.

CLASSIFICATION

A model of ABC for a single queue is depicted in Fig. 2. The bandwidth controller adjusts the allocated bandwidth (service rate) on a time scale between a packet level and a connection level (e.g., one tenth of a second to seconds) based on online measurement of the system state and measured traffic information and possibly traffic prediction, such that the QoS requirement is achieved while maintaining a high bandwidth utilization. Existing ABC algorithms can be classified according to the underlying control technique used, or according to the QoS metrics guaranteed, as shown in Fig. 3. Since the classification based on the QoS metric is straightforward, only the discussion on the underlying control technique will be provided.

Regarding the underlying control technique, ABC schemes can roughly be classified as being either *closed-loop* or *open-loop*. The closed-loop, or feedback control, approach arises naturally in this context where the packet loss, average queue length, or other system states, are regularly observed to provide the feedback to adjust the allocated bandwidth. The closed-loop control approach can be further categorized based on the guaranteed QoS metrics, including the average queue length [12, 13], loss [14–17], and delay [18, 19]. The open-loop control approach involves predicting the input traffic rate using the past history. The service rate is then adjusted to match the predicted rate to attain zero packet loss or low queuing delay. Although effective, using open-loop control to achieve a given target QoS is difficult due to the lack of explicit relationship between the predicted traffic rate and the target QoS. Consequently, most of the existing work for open-loop ABC only attempts to deliver very low or zero packet loss rather than guarantee it quantitatively [20–22]. The hybrid of feedback control and open-loop control is also possible to eliminate the drawbacks found in both approaches [23]. Since it is the QoS metrics that are of interest, in the following five sections we describe ABC algorithms in terms of their guaranteed QoS metrics. A summary and comparison of the ABC algorithms discussed are provided in Table 1 toward the end of the article.

QUEUE LENGTH CONTROL

A great deal of work has focused on using ABC to maintain the average queue length at some desired target value so that the packet loss and the average queuing delay are kept low. Many researchers have investigated the control based on a fluid-flow model of a queuing [12, 13, 24, 25]. In the fluid-flow model, the ensemble average queue length $Q(t)$ in a single server queue is described by a non-linear differential equation

$$\frac{dQ(t)}{dt} = -C(t)G(Q(t)) + \lambda(t) \tag{1}$$

where $C(t)$ is the service rate, $G(\cdot)$ is a function obtained from matching the average utilization as a function of the average queue length at steady-state, and $\lambda(t)$ is the ensemble average packet arrival rate. A closed form of $G(\cdot)$ is generally not easy to determine unless for a simple Markovian model, or using a function fitting from measurement data. In [12], the desired trajectory of $Q(t)$ is first specified as a function of $\lambda(t)$, and the control objective is to determine $C(t)$ that minimizes the tracking error. $C(t)$ is derived from a Lyapunov-based control and is shown to be directly proportional to the tracking error. A series of work by Pitsillides *et al*. [13, 24, 25] considers bandwidth control of an $M/M/1$ queue modeled by Eq. 1 in various contexts. A multilevel optimal control approach is developed to coordinate the decentralized nodal bandwidth control to improve the global network performance and robustness [24]. In [25], the proportional control, $C(t) = \alpha(1 + Q(t))$, in the context of an ATM Available Bit Rate (ABR) service is derived to handle a finite buffer and finite capacity case, where α is chosen based on the buffer size and capacity limit, assuming the bound on $\lambda(t)$ is known. Also based on Eq. 1, feedback linearization and robust adaptive control is applied to derive the bandwidth control law to maintain a low average queue length (and hence low

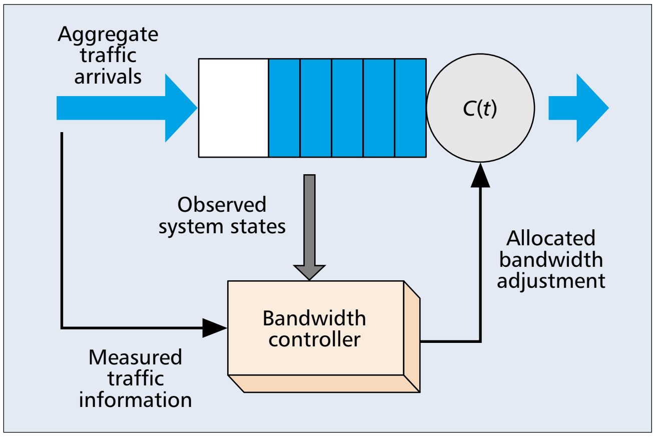

Aggregate traffic arrivals
$C(t)$
Observed system states
Allocated bandwidth adjustment
Bandwidth controller
Measured traffic information

FIGURE 2. *A model of adaptive bandwidth control for a single queue.*

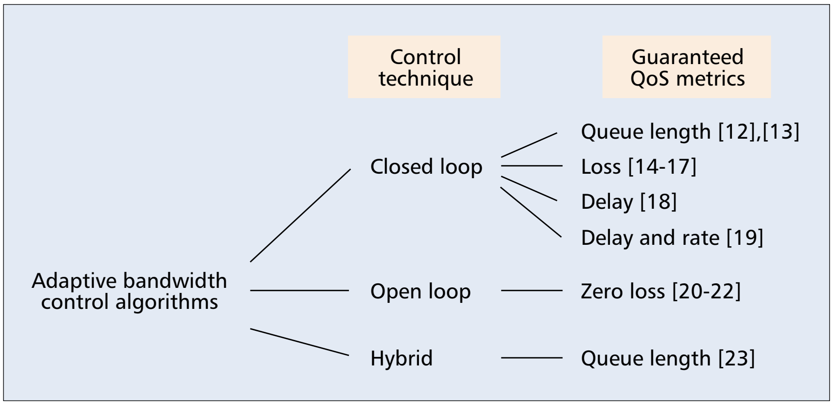


FIGURE 3. *Classification of ABC algorithms that guarantee packet-level QoS metrics*

delay), in the context of a DiffServ Premium service class [13]. There are disadvantages to using the control based on Eq. 1:

- The control is performed in continuous time.
- $\lambda(t)$ may not be measurable.
- The form of $G(\cdot)$ changes from one queuing system and configuration to another.

Hybrid control that combines the input traffic information and the measured average queue size has recently been developed in [23]. Specifically, the wavelet-energy approach is used to separate the short-term and long-term fluctuations in the arrival rate time series data into different frequency bands. The energy content in each sub-band frequency as well as the measured average queue length are then used to calculate the bandwidth to be allocated in the next time interval. The simulation results in [23] show that the queue length can be maintained at some constant level with the above wavelet-energy approach, but one has no control over the value at which the queue length is to be maintained.

NON-ZERO LOSS GUARANTEE

DIRECT FEEDBACK

Guaranteeing non-zero packet loss with ABC is to ensure that the cumulative loss converges to the desired (non-zero) target packet loss rate and stays at that value. ABC algorithms based on integral control are considered in [14, 15]. In those algorithms, the allocated bandwidth is adjusted in proportion to the difference between the measured loss and the target loss, and finally reaches some constant value. In [14], the discrete-time bandwidth adjustment at time instant t_k follows the equation

$$C_{k+1} = C_k + G_k \ln(P_k/\varepsilon) \tag{2}$$

where C_k is the service rate, G_k is the feedback gain, P_k is the measured packet loss rate over $[t_{k-1}, t_k)$, and ε is a target packet loss rate. The error feedback term $\ln(P_k/\varepsilon)$ governs how much C_k should be adjusted; the amount of adjustment becomes smaller as P_k gets closer to ε.[1] The parameter G_k amplifies the feedback and controls how quickly the algorithm converges. A heuristic procedure to dynamically adjust G_k is also proposed in [14], which basically makes G_k smaller as P_k gets closer to ε. The performance of a slightly modified version of this algorithm for MPEG video traces and a multihop network setting are investigated in [26]. Hsu and Walrand [15] propose a similar bandwidth adaptation algorithm:

$$C_{k+1} = C_k + (G/k)(L_k - \varepsilon A_k) \tag{3}$$

where G is a positive constant, and L_k and εA_k are, respectively, the number of lost packets and the expected number of losses during the time interval $[t_{k-1}, t_k]$. The algorithm is proved to converge to the minimum bandwidth almost surely for a Markov modulated fluid source. However, no performance evaluation is presented therein. Observe that in both Eq. 2 and Eq. 3 the feedback gain G_k keeps decreasing over time so that smaller bandwidth adjustment is made as time increases and the allocated bandwidth will eventually reach some constant value. While Eq. 2 and Eq. 3 intend to make the allocated bandwidth converge to some minimum steady-state value, in a typical dynamic network load environment, one needs regular bandwidth adjustment.

To acquire some key insights on the performance of ABC with integral control of loss feedback, we show the performance of Eq. 3 using a simulation model coded in CSIM,[2] a process-oriented simulation language based on C. The model is a single queue with a finite buffer as shown in Fig. 2. The controller has two main parameters including the initial gain (G) and the control time scale T_c, the length of time interval at which the bandwidth adjustment is made. The input traffic used in the simulation is fractional Gaussian noise (fGn), which has three parameters including the mean rate (λ), variance coefficient (a), and Hurst parameter (H). The values λ = 5,000 packets/s, $a = 1$, and $H = 0.85$ are used. Unless stated otherwise, this input traffic will be used in the simulation study through the rest of the paper. The buffer size is 200 packets. As shown in Fig. 4, the cumulative loss under the integral control may or may not reach the target loss rate $\varepsilon = 10^{-3}$ depending on G and T_c. Furthermore, we found that when either G is increased to 10000, or T_c to 2 seconds, there is no loss and the utilization significantly reduces (0.394 and 0.706, respectively). This is because the bandwidth assigned becomes very large as a result of packet loss at early steps but decreases very slowly afterward due to the decreasing feedback gain.

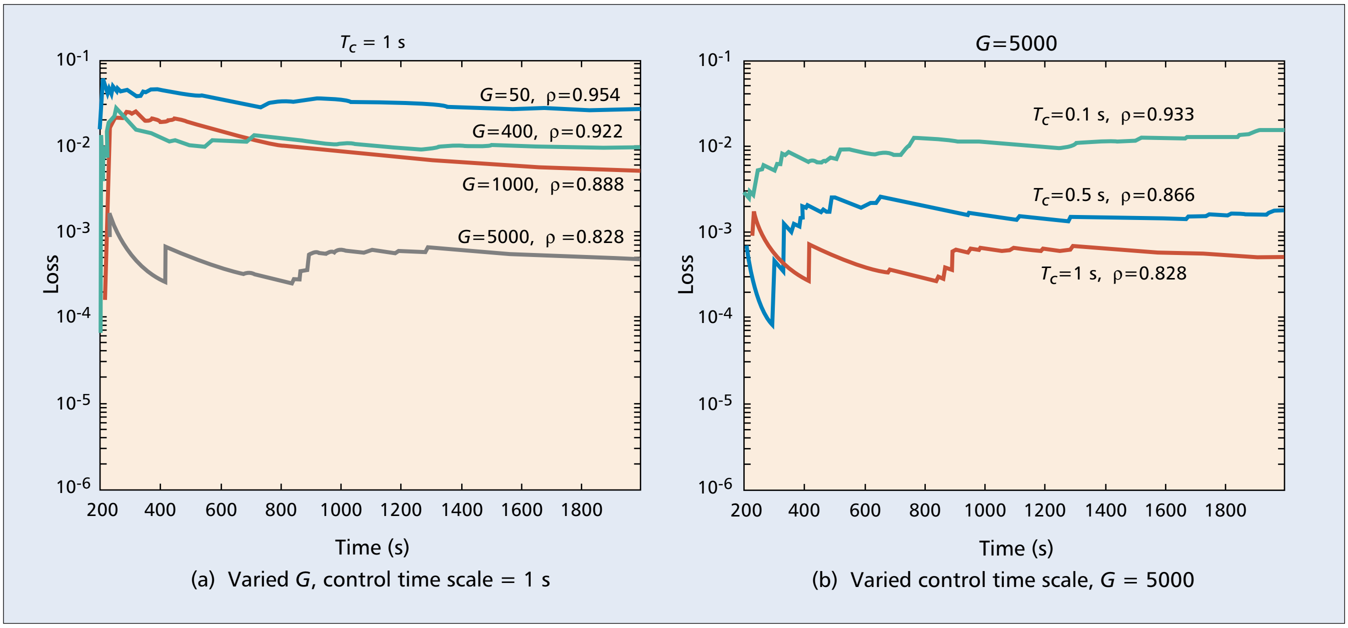
INDIRECT FEEDBACK

Another approach to guarantee packet loss is to translate a loss requirement to some other performance measure; the control then attempts to maintain that performance measure to indirectly achieve the target loss requirement. This approach is adopted in [16, 17], where the control does not necessarily make the allocated bandwidth reach a constant value as in [14, 15]. We define the utilization as the fraction of time the server is busy. In [16], a target loss rate (ε) is translated to the target average queue length ($\overline{Q}$) as well as the target utilization ($\overline{\rho}$) through an $M/M/1/K$ queue assumption, and the control attempts to maintain the utilization at $\overline{\rho}$ by observing changes in the average queue length. The measured average queue length ($\hat{Q}$), packet loss rate ($\hat{\varepsilon}$), packet arrival rate ($\hat{\lambda}$), and packet departure rate ($\hat{r}$) are used to estimate the utilization ($\hat{\rho}$) according to the expression derived from an $M/M/1/K$ queue. In particular,

$$\hat{\rho} = \begin{cases} \dfrac{\hat{\lambda}}{\hat{r}} & \dfrac{\hat{\lambda}}{\hat{r}} > 1 \\ 0.5 \left[\sqrt{(\hat{Q} - (K+1)\hat{\varepsilon})^2 + 4\hat{Q}} - \hat{Q} + (K+1)\hat{\varepsilon} \right] & \text{Otherwise} \end{cases} \tag{4}$$

[1] *The behavior of the algorithm when* $\mathrm{P_k} = 0$ *is not discussed in [14]. From Eq. 2, the control goes into an invalid state when no packet loss is observed during the measurement period.*

[2] *http:\\www.mesquite.com*



(a) Varied *G*, control time scale = 1 s

(b) Varied control time scale, *G* = 5000

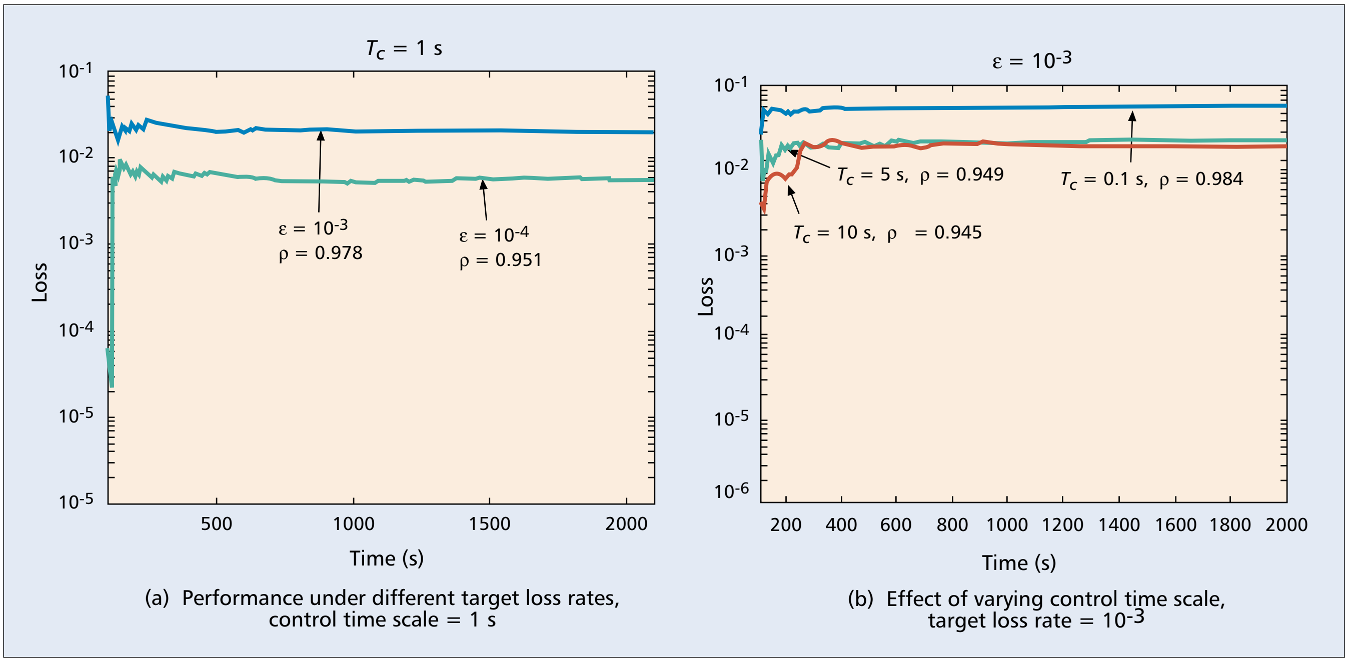
■ **FIGURE 4.** *Loss performance of direct loss feedback [15] (target loss rate = 10^{-3}).*

The control action, invoked when $\hat{Q}$ deviates from $\overline{Q}$ beyond some thresholds, is used to adjust the service rate in such a way that the difference between $\hat{\rho}$ and $\overline{\rho}$ is minimized by multiplying the current service rate with $\overline{\rho}/\hat{\rho}$. In real networks, however, the input traffic usually has complex (e.g., non-stationary, long-range dependent) and unknown characteristics, and thus the queuing process can have radically different characteristics from that of the $M/M/1/K$ queue. As shown in Fig. 5, the resulting loss is far from the desired target loss with the same fGn traffic input used earlier.

In [17], we take a different approach by applying fuzzy control to maintain the average queue length between a queue threshold pair u_{th} and l_{th} that depends on the target loss rate. For a system that cannot be adequately described by detailed mathematical equations, fuzzy control is a convenient approach to synthesize a non-linear controller whose control laws are heuristically derived through some key insights of the process under control. Basically, two feedbacks — the average queue length (Q_k) determined over $[t_{k-1}, t_k)$ and its changes (ΔQ_k) — are translated through the membership functions to linguistic values associated with the degree of certainty ranging between 0 and 1. Fig. 6 shows the membership function for Q_k. We then define a set of rules, called a *rule-base*, which represents heuristic descriptions of how to achieve good control. In our case, each rule is associated with some amount of bandwidth adjustment which depends on the behavior of Q_k and ΔQ_k. For instance,

IF $\hat{Q}_k$ is low and $\Delta\hat{Q}_k$ is high THEN decrease the bandwidth by 2.5 percent of the average input arrival rate.

From the rule-base, we determine the amount of band-



(a) Performance under different target loss rates, control time scale = 1 s

(b) Effect of varying control time scale, target loss rate = 10^{-3}

■ **FIGURE 5.** *Loss performance using control based on the M/M/1/K model [16].*

width adjustment ΔC_i due to rule i by weighting the amount of bandwidth adjustment specified in rule i by the product of the membership function values of Q_k and ΔQ_k. The total amount of bandwidth adjustment is then the sum of ΔC_i.

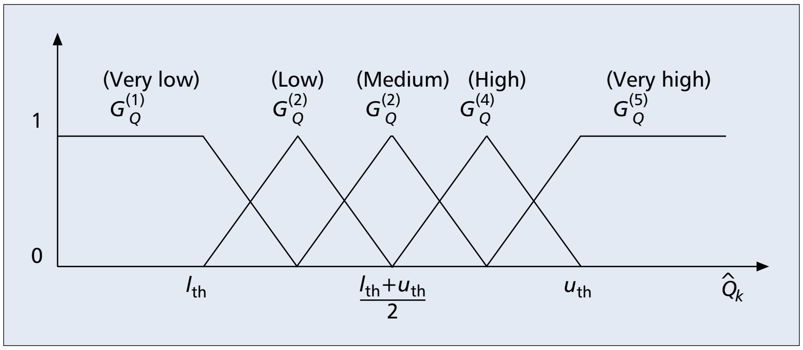


FIGURE 6. *Membership functions for the average queue length.*

Figure 7 shows the packet loss performance of the fuzzy controller, with the control time scale varying from 1 second to 20 seconds, under the same queue configuration and input traffic used earlier. Currently, the mapping between the target loss rate and the target queue length is based on the $G/M/1$ queuing model with trial and error tuning. The results indicate that with the right target queue length, the target loss rate can be achieved with high utilization and the performance appears to be robust against the control time scale, although the utilization decreases as the control time scale increases. Compared to Fig. 4b, at $\varepsilon = 10^{-3}$, the fuzzy control can achieve the target loss with higher utilization (0.855 versus 0.828). Also note that although the utilization due to the control shown in Fig. 5 is higher, it yields the loss rate that is more than an order of magnitude higher than the target value.

Two issues that need to be resolved for the fuzzy controller are the choice of the control time scale and the mapping between the target queue length and the target loss rate. We expect that an appropriate choice of the control time scale would change with the network parameters such as the buffer size and the input average rate. We also found that the packet loss and queue length translation based on the $G/M/1$ queuing model performs poorly at a large buffer size. These two issues are being investigated in our ongoing research.

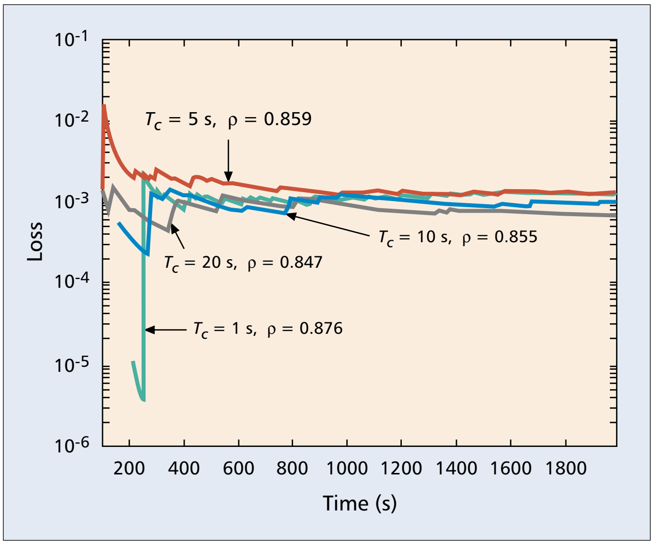


FIGURE 7. *Loss performance of fuzzy control [17] at different control time scales (target loss rate = 10^{-3}).*

ZERO LOSS GUARANTEE

Specifying quantitative loss guarantee with ABC is appealing if the target loss rate is not too low, e.g., greater than 10^{-4}. The control techniques discussed in the previous section tend to perform poorly when dealing with low packet loss rates. With direct loss feedback, we found that the feedback term $L_k - \varepsilon A_k$ in Eq. 3 becomes so small that the resulting cumulative loss is slow to converge due to very small bandwidth adjustment steps. With the indirect feedback based on the average queue length, the target queue length used in [17] can also become so small at low target loss rates that the control to keep the average queue length at the target is ineffective. At low target loss rates, we therefore argue that it is better to provide zero loss guarantee.

Zero loss guarantee is attained by allocating the bandwidth that *matches* the arrival rate, which is best implemented with the open loop control. That is, if the service rate is sufficiently greater than the arrival rate in every time interval, zero loss can be expected. Zero loss guarantee with open-loop ABC is particularly appealing to video traffic, the characteristic of which is that the data rate can vary significantly from one frame to another. Effective open-loop ABC requires prediction of the input traffic to determine the amount of allocated bandwidth in subsequent time intervals. The prediction can be done either directly in a time domain, or by incorporating the knowledge of traffic in the frequency domain. More precisely, the time-domain prediction directly calculates the required bandwidth only from traffic rates measured in the past. The frequency-domain prediction first uses the knowledge of traffic in a frequency domain to properly filter the traffic; then a time-domain prediction is applied to the filtered traffic. These two approaches are described below.

TIME DOMAIN

Let x_k be the input traffic rate measured over time interval $[t_{k-1}, t_k]$. In simple forms of time-domain prediction, the bandwidth C_{k+1} allocated during the next interval $[t_k, t_{k+1})$ can be set to one of the following:

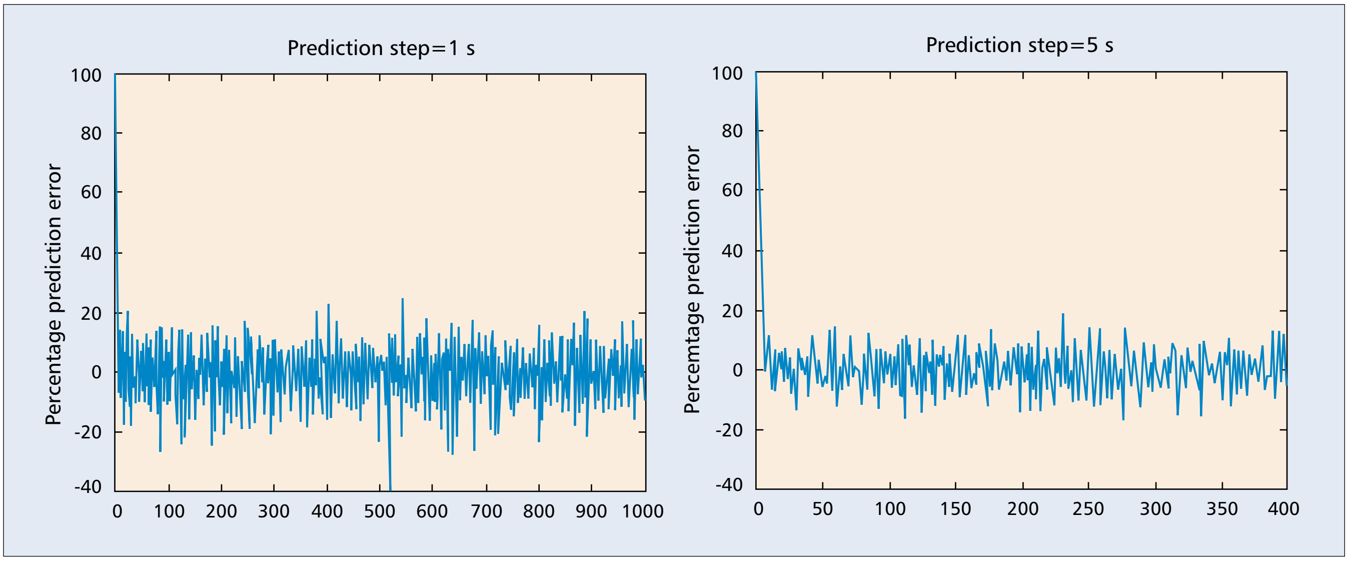
- The measured traffic rate over the kth interval, $C_{k+1} = x_k$.
- The average of the measured rate of past M intervals;

$$C_{k+1} = \bar{x}_k(M) \equiv (1/M)\sum_{i=0}^{M-1} x_{k-i} \quad .$$

- The weighted sum between $\bar{x}_k(M)$ and x_k; $C_{k+1} = (1-\alpha)\bar{x}_k(M) + \alpha x_k$.

It is apparent that these kinds of allocation are unlikely to yield good performance for bursty traffic due to poor traffic prediction, as confirmed by results in prior work [23, 27].

Better time-domain prediction methods have been developed in [21, 28]. In [28], the D-BIND deterministic traffic model proposed in [29] is fitted to M successive frames of the video sequence in real-time, and the bandwidth is adjusted accordingly to match the worst case traffic arrivals to guarantee zero loss. However, the fitting process is computationally intensive, and the deterministic traffic model can result in underutilization. Adas [21] uses the Least Mean Square (LMS) error linear predictor to predict the bandwidth requirement of future frames in MPEG video sequences. In particular, a pth-order linear predictor has the form



■ **FIGURE 8.** *Performance of LMS prediction on fGn traffic.*

$$\hat{x}_{k+n} = \sum_{l=0}^{p-1} w_l x_{k-l} \tag{5}$$

where x_k represents the frame rate at time step k, and w_l are the prediction filter coefficients. Note that previous p values are used in the prediction. In an LMS approach, w_l are time-varying and determined from the prediction error $x_{k+n} - \hat{x}_{k+n}$ and previous values of x_k. In this case, the prior knowledge of traffic statistics and stationarity assumption are not needed. At $n = 1$ and $p = 12$, it is shown that the bandwidth allocated according to the predicted values results in less delay and fewer buffers needed. The number of reallocations is reduced by introducing an allocation threshold, whereby the bandwidth is adjusted according to the predicted rate only if the predicted rate differs from the currently allocated rate by more than some threshold value. This same prediction technique is adopted in [30] for the dynamic bandwidth allocation of self-similar video conferencing traffic. The performance of the LMS prediction on LRD traffic is shown in Fig. 8, where we apply the prediction with $p = 20$ to the fGn traffic used in the simulation study in the previous section. We can see that prediction error as high as 10 percent is typical, which can lead to excessive loss in some time intervals. Gallardo *et al*. [31] propose the α-stable self-similar process to model aggregate traffic. They go on to develop a prediction method for the above traffic process based on an autoregressive model. The predicted traffic information is used to control the token rate of a leaky bucket regulator at the edge node to reduce packet loss.

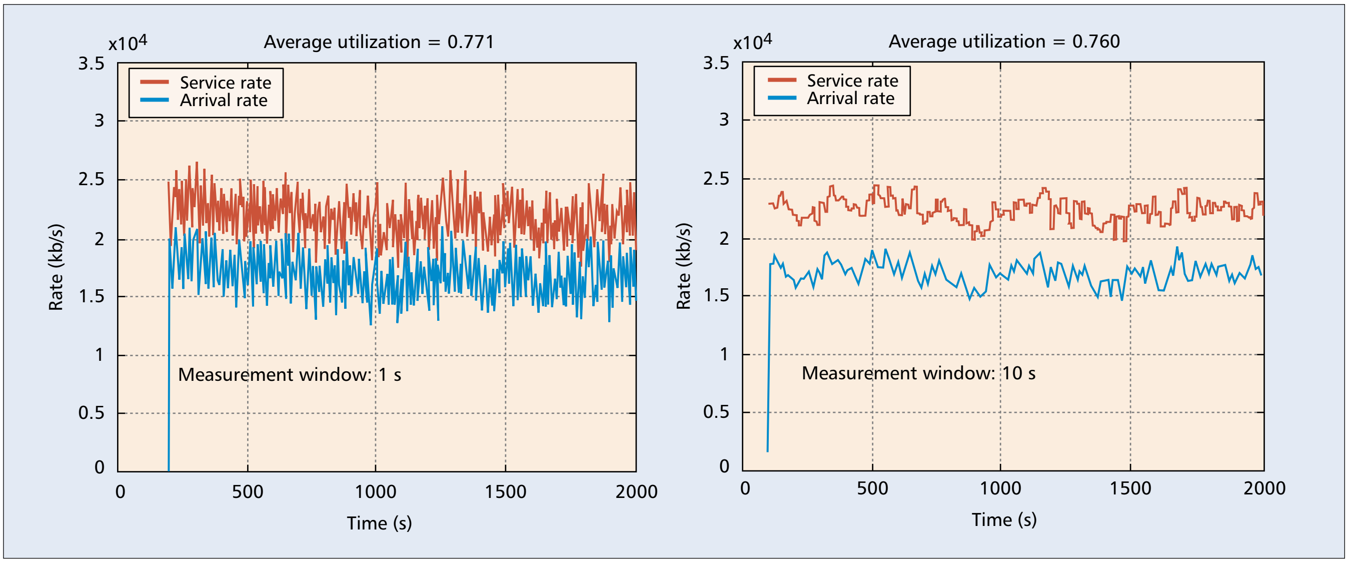
In [22] Duffield *et al*. consider ABC in the context of VPN bandwidth allocation. They suggest dynamically resizing a VPN link capacity in their *hose* VPN model to attain better bandwidth sharing among traffic aggregates in a sink tree VPN. In the hose model, a VPN customer specifies a set of end points to be connected to, as well as a rough estimate of incoming and outgoing aggregate traffic from and to those end points, comprising the required bandwidth on the access link to the network. From initial bandwidth assignments, the bandwidth on both the access link (at hose interface) and internal links are dynamically resized. The bandwidth resizing is done from the ingress node, which in turn signals the reallocation along the path to the egress node.

Two prediction methods to obtain the required bandwidth from the sampled traffic rate measured over a fixed measurement window are proposed in [22] to achieve the bandwidth resizing. Although their work does not explicitly state the objective of bandwidth resizing to guarantee zero loss, these two methods are discussed here because the required bandwidth, which is calculated from the time-domain prediction of past traffic arrivals, is intended to match the input traffic. With the measurement window of size T_c and the sampling interval T_s, $n \equiv \lfloor T_c/T_s \rfloor$ sampled rates are available for the estimation. The first method is called the *local maximum predictor*, whereby the maximum of the sampled rates over the measurement window is used. The other one is called the *local Gaussian predictor*. It sets the allocated rate in the next measurement window to $m + \alpha\sqrt{\upsilon}$, where m and υ are, respectively, the mean and variance of the sampled rates over the measurement window. The multiplicative factor α is selected such that the arrival rate will exceed the allocated bandwidth $m + \alpha\sqrt{\upsilon}$ with probability $\zeta = 1 - \Phi(\alpha)$, where $\Phi(\alpha)$ is a standard normal CDF. That is, $\alpha = \Phi^{-1}(1 - \zeta)$. The paper also discusses the robustness of the predictors with respect to traffic non-stationarity, measurement errors, and short time-scale burstiness. It is argued that the predictors are robust with respect to non-stationarity provided that the measurement time scale is less than the non-stationary time scale. Regarding the measurement errors, they prove that for the local Gaussian predictor, the error from n samples can be avoided by modifying α to

$$\sqrt{(n+1)(\exp((\Phi^{-1}(1-\zeta))^2/n)-1)}\,.$$

To deal with burstiness in multiple time scales, which results in underestimating the required bandwidth, it is suggested that *a priori* knowledge of the rate variance at different time scales be used to correct α.

Figure 9 shows sample paths of the allocated bandwidth due to the local Gaussian predictor at a different measurement window T_c under the same queue configuration and the fGn input traffic used previously with $\zeta = 0.95$. Zero loss is achieved in both cases. In these cases, we expect the prediction to perform well because the underlying traffic rate is Gaussian. However, as the input traffic has been changed to an aggregate of Pareto on-off sources (with the same Hurst parameter) as shown in Fig. 10, the performance of the local Gaussian predictor degrades, resulting in some packet loss. This suggests the modification of ζ to increase the allocated bandwidth. Therefore, getting the right α can be difficult because it requires manual tuning. Note also that the bandwidth allocation totally ignores the presence of buffers. The



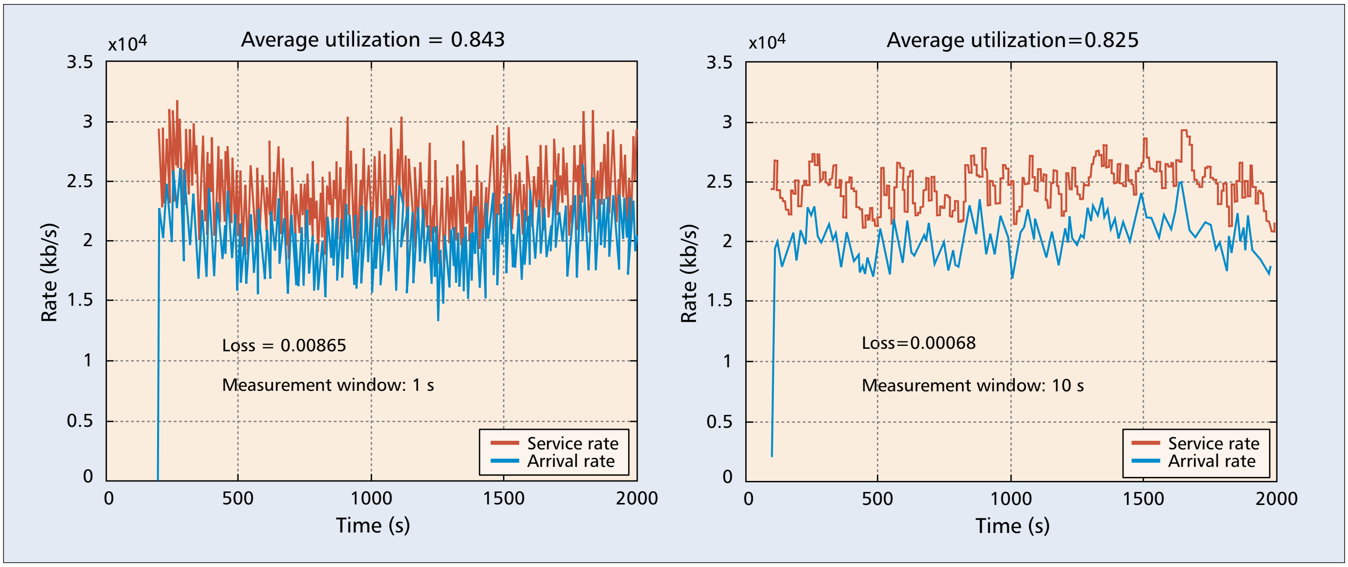
■ **FIGURE 9.** *Allocated bandwidth with local Gaussian predictor [22] under fGn traffic (sampling interval = 5 ms).*

utilization can be improved if the buffer is taken into account, especially when a large buffer exists.

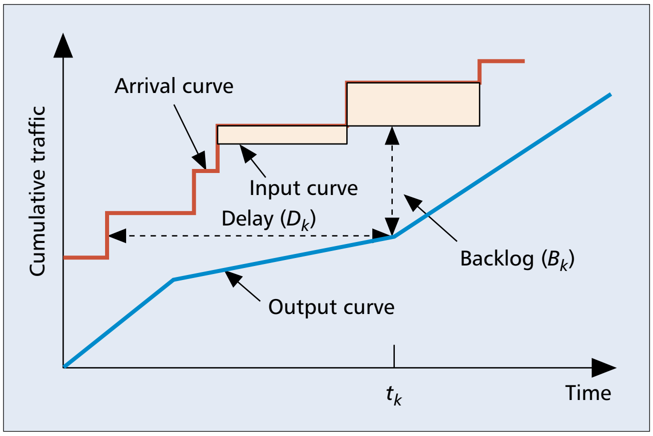
FREQUENCY DOMAIN

Another approach to open-loop ABC is to analyze the input traffic in the frequency domain to obtain a proper cutoff frequency for traffic filtering. A time-domain prediction based on the filtered traffic is then applied for the bandwidth allocation. In [20, 32], by considering the rate of an MPEG video trace as a signal, frequency domain analysis shows that the bandwidth requirement to guarantee no packet loss is essentially captured by the low-frequency traffic filtered at a properly selected cutoff frequency ω_L. In particular, if $\hat{x}_t$ is the filtered traffic of the input traffic $x(t)$ at ω_L, the minimum static bandwidth requirement is $\max_t \hat{x}(t)$, which must be obtained only through an off-line process. The cutoff frequency ω_L depends on the buffer size. At low buffer size, ω_L will be high because the link capacity has to accommodate high-frequency components, i.e., burstiness, of the input traffic. At zero buffer space, we basically need to allocate the minimum bandwidth at the peak input rate to prevent packet loss, corresponding to $\omega_L = 2\pi/T$ where T is the minimum packet interarrival time. As the buffer size increases, the high frequency components can be absorbed by the buffer such that ω_L can be reduced, and higher utilization can be obtained. At infinite buffer space, one can set ω_L to zero and the peak filter rate is simply the average input rate because we obtain the DC term with $\omega_L = 0$.

Statically assigning the bandwidth at $\max_t \hat{x}(t)$ is inefficient because most of the time $\hat{x}(t)$ stays much below its maximum. It is suggested in [32] that since $\hat{x}(t)$ has slow time variation and hence is predictable, dynamic bandwidth allocation based on the online observation and prediction of $\hat{x}(t)$ can be implemented. To implement the above approach in real time, Recursive Least Square (RLS) and Time-Delay Neural Network (TDNN) predictors are used to predict the low-frequency variation, given that the traffic is passed through a low-pass filter with appropriate cutoff frequency [20]. The filtered input rate is predicted M steps (sampling intervals) in advance and the bandwidth is periodically adapted every M-step window. The allocated bandwidth is the maximum of these M



■ **FIGURE 10.** *Allocated bandwidth with local Gaussian predictor [22] under an aggregate of Pareto on/off sources (sampling interval = 5 ms).*



■ **FIGURE 11.** *Delay and backlog in a busy period.*

predicted values multiplied by some safety factor. The number of adjustments can be reduced by introducing a hysteresis threshold.

Although ω_L can be determined analytically for some special class of input processes such as MMPP or Gaussian [32], this is not the case in general and ω_L must be determined on an off-line trial and error basis. For a prerecorded or stored video, the off-line analysis can be accomplished by fitting the traffic to one of the traffic models above, in which ω_L can be obtained analytically. However, for traffic generated on the fly or from live sources, determining ω_L in real time is not possible.

Despite the fact that the appropriate cutoff frequency to filter the traffic for open-loop ABC may not be obtained analytically under traffic models other than MMPP or Gaussian, the lower and upper bounds on the cutoff frequency related to the bandwidth allocation has been identified for a queue with finite buffer K and fixed link capacity C, subject to a deterministic maximum delay constraint $d_{\max} = KC$ [33]. Note that this delay constraint is equivalent to zero loss guarantee. For a given $d_{\max}$, it is shown in [33] that if most of the frequency components of the input traffic is lower than $\omega_L = 2\pi/(200 d_{\max})$, the bandwidth to guarantee zero loss is approximately $\max_t x(t)$. In this case, buffering becomes ineffective because the allocated bandwidth must be assigned at the peak input rate. On the other hand, if most of the frequency components are higher than $\omega_H = 2\pi/d_{\max}$, the bandwidth to guarantee zero loss is approximately the average input rate because the frequency components are absorbed by the buffer. For input traffic whose frequency components are in $[\omega_L, \omega_H]$, the required bandwidth is between the peak rate and the average rate, which is consistent with the results in [32]. For the same input traffic, if the buffer size increases, ω_L and ω_H will decrease, therefore reducing the bandwidth requirement. These results also imply that the traffic correlation and statistics at time scales beyond $200 d_{\max}$ and less than $d_{\max}$ are negligible.

DELAY GUARANTEE

Bandwidth adjustment to guarantee a probabilistic delay bound is considered in [18]. For a single server with service rate C, the probabilistic delay bound QoS requirement is that the probability that a packet queuing delay d will be greater than D is less than ε, i.e., $\mathbb{P}\{d > D\} < \varepsilon$. The delay bound requirement is translated to the bound on the queue tail distribution $\mathbb{P}\{Q \geq \theta\} < \varepsilon$, where $\theta = D \times C$. The problem is to estimate $\mathbb{P}\{Q \geq \theta\}$through online measurement and adjust C from its initial allocation (found from an effective bandwidth formula) so that the QoS requirement is met.

The problem is solved in two steps. In the first step, the queue length distribution is derived from the measured statistics of packet arrivals. Basically, the tail distribution of the queue occupancy Q must be assumed to follow some specific distribution. For example, the Weibull distribution, $\mathbb{P}\{Q \geq x\} = Ae^{-Ix^b}$, can be used if the traffic is known to be long-range dependent. The exponential tail distribution, $\mathbb{P}\{Q \geq x\} = Ae^{-Ib}$, can also be used for a large class of traffic models provided that b is small [2, 34]. For the Weibull distribution, measuring $\mathbb{P}\{Q \geq x\}$ at three different queue thresholds x are sufficient to parameterize the distribution and extrapolate $\mathbb{P}\{Q \geq \theta\}$. In the second step, the change in the queue threshold violation probability $\Delta\varepsilon$ as the bandwidth changes from C to $C + \Delta C$ must be determined, which is derived in [18] only for the case of an exponential tail distribution. This information is then used to adjust the allocated bandwidth to bring ε to the target value. The primary drawbacks of the above method is that the assumption of a specific queue length distribution may not hold, and the relationship between $\Delta\varepsilon$ and ΔC may not be easily determined for other probability distributions. Furthermore, the measurement error may seriously affect the accuracy of the queue length extrapolation.

GUARANTEEING MULTIPLE QOS METRICS

Recently, Christin *et al.* [19] developed a class-level ABC for a quantitative Assured Forwarding (AF) DiffServ service class with absolute and relative differentiation of delay and rate guarantees over the duration of a busy period.[3] By keeping track of every arrival and departure in a given busy period, the performance measures of interest are determined directly from the arrival curve (from cumulative packet arrivals), the input curve (from cumulative packets entering the queue), and the output curve (from cumulative packet departures). As illustrated in Fig. 11, the delay of the departing traffic at the kth event (at time t_k), denoted by D_k, is the horizontal distance between the input curve and the output curve, the backlog (B_k) is the vertical distance between the input curve and the output curve, and the amount of traffic dropped is equal to the shaded tan area. Given the delay and rate constraints (both absolute and relative), the service rate is adjusted for every packet arrival instant to the queue such that all the constraints are satisfied. The absolute delay is guaranteed by ensuring that all backlogged traffic must be sent within its delay bound. Namely, the required bandwidth at the kth event to achieve the delay bound guarantee d is given by

$$C_k \geq \frac{B_k}{d - D_k}. \tag{6}$$

Given the minimum guaranteed rate requirement, the minimum required bandwidth for both rate and delay guarantees is the maximum between the two.

For a proportional delay differentiation among N traffic classes, we want to have $D_k(i) = \tau_{i-1} D_k(i-1)$, where τ_i is the differentiation ratio between class i and $i+1$, and $D_k(i)$ is the class i delay at the kth event, $i = 1, 2, \ldots, N-1$. Let

$$m_i = \prod_{j=1}^{i-1} \tau_j,$$

and

$$D_k^*(i) = D_k(i) \prod_{j=1, j\neq i}^{N} m_j \tag{7}$$

[3] *A loss rate guarantee is also provided by tail-dropping traffic in the buffer, independent of delay and rate guarantees.*

It is shown that the delay differentiation is achieved if $D_k^*(i)$ can be maintained at $D_k^* \equiv N^{-1} \Sigma_{i=1}^{N} D_k^*(i)$. So the error $e_k(i) = D_k^* - D_k^*(i)$ is used to adjust the service rate of class i at the kth event. The amount of adjustment is $G_k e_k(i)$, where G_k is a feedback gain. With some simplified assumptions to model the system as a discrete-time linear control system, the range of G_k to ensure stability has also been given. Under a certain condition shown in [19], the absolute/proportional delay differentiation and rate guarantees are not achievable simultaneously. We believe that this shortcoming is insignificant because guaranteeing such multiple QoS metrics simultaneously is rarely needed. However, attempting to guarantee different metrics at every arrival/departure instant requires computationally expensive processing and leads to an implementation that is not scalable for high-speed links. As shown in [19], a 1-GHz PC router can achieve less than 200 Mb/s throughput for a few number of QoS constraints.

Issues in ABC

Control Time Scale

Of particular importance in ABC is the issue of which time scale should be used to adjust the bandwidth, which we here refer to as a *control time scale*. Presently, most of the work in ABC selects the time scale to adapt the bandwidth based on a trial and error basis, and there has been relatively little work in the literature that provides insights into how to properly select it.[4] Adjusting the bandwidth too frequent can lead to high overheads, highly fluctuated bandwidth, and potentially poor control performance due to inaccurate feedback obtained from a too short measurement period. On the other hand, if the bandwidth is not being adjusted frequently enough, poor control performance may result under traffic non-stationarity. Also, the packet loss can be excessive because the time period during which the allocated bandwidth mismatches the arrival traffic becomes larger. Nevertheless, although the control time scale for ABC algorithms that are based on feedback control is still not well understood, some insights are available into the time scale of those ABC algorithms based on open-loop control of traffic rate. Previous work in [32] shows that the traffic low-frequency component (which is filtered at a cutoff frequency that depends on the buffer size) dominates the average queue length performance. As such, the control time scale should also be able to react to the low-frequency variation of the input traffic filtered at an appropriate cutoff frequency.

Fu and Knightly [35] investigate dynamic traffic rate control whereby the bandwidth is allocated to match the requested aggregate traffic rate by using step allocation with hysteresis. They model the aggregate rate fluctuation with a random-phase sinusoidal signal superimposed with a random white noise and analytically evaluate the performance metrics, including the overload probability, the reserved resource utilization, and the normalized residual bandwidth. The proposed sinusoidal model fits real traces well and is able to predict fairly accurately performance metrics when the variance of the traffic is small, but becomes inaccurate otherwise. The study reveals that the effectiveness of such a control mechanism strongly depends on the relative time scale between the control and the traffic, as well as the variance of the traffic rate itself. When the bandwidth adjustment time period is of shorter duration than the long-term traffic fluctuation, i.e., the traffic low-frequency component, and when the variance of the traffic rate is not high, the reserved resource utilization is close to unity.

Admission Control

Under ABC, the allocated bandwidth will dynamically change over time. As such, admission control becomes a problem because it is unclear when new aggregates can be admitted to the queue. If too much traffic is admitted, there will be a time period over which the bandwidth cannot be allocated as required by the ABC algorithm in use, leading to QoS degradation in those bandwidth violation periods. The problem of admission control under ABC can be divided into two subproblems. The first one is to investigate the relationship between the degree of QoS degradation and the bandwidth violation. With such knowledge available, we can quantify how much bandwidth violation can be tolerated while retaining an acceptable level of QoS. The second one is to discover the impact of admitting new aggregates with respect to the bandwidth violation. By combining the solutions of these two subproblems, admission control in ABC can be exercised successfully. Exactly how to obtain those solutions requires further investigation.

Control in Multiple Queues

When multiple traffic classes share the same link or bandwidth pool, the single-queue control must be extended to deal with fair bandwidth access as well as other issues, as discussed in the following. Palazzo [12] extends the Lyapunov-based control discussed previously to a hybrid Lyapunov and optimal control to deal with the multiple queues. In particular, each queue will receive the bandwidth proportional to its tracking error according to the Lyapunov-based control when the sum of allocated bandwidth is below the link capacity. Otherwise, the expression derived from the optimal control that minimizes the sum of the tracking errors from each queue is used. Likewise, the work by Chong *et al.* [20], also described earlier, consider the case of multiple ATM virtual circuits (VCs) by allocating the link bandwidth at time t in proportion to the VC bandwidth requirements derived from the control algorithm.

ABC for multiqueues must also deal with additional issues that have not been addressed elsewhere in the literature. The interaction among traffic classes complicates the control by introducing the fairness constraint that must be met in addition to the packet-level QoS. Since the link capacity is finite, the bandwidth request due to the control is not always granted. The question is then how to devise the control that allows each queue to equally access the resource in a long run. Apparently, the control must incorporate some mechanisms to prevent the bandwidth hogging of one class or aggregate over the others. In a more general case, priority access has to be considered. The QoS violation of one class may result in higher loss in the revenue or service integrity than the others. As an example, for an equal length of the QoS violation period, users may find it more irritating in voice traffic than video traffic or WWW traffic. Given that class A has higher priority than class B, the request for bandwidth allocation for class A due to the bandwidth control should be rejected less often than that of class B, and the fraction of the QoS violation period of class A should be lower than that of class B in the long run.

Summary and Concluding Remarks

In this paper, we have presented a survey of adaptive bandwidth control (ABC) mechanisms and made the case that ABC can be a viable alternative to static bandwidth allocation

[4] *Our notion of control time scale is different from the critical time scale or dominant time scale discussed in the context of LRD traffic, which states that the amount of traffic correlation beyond the above time scale, which basically increases with the buffer size, has no impact on the steady-state loss performance.*

Algorithms	Guaranteed metric	Descriptions/advantages	Disadvantages
Palazzo [12], Pitsillides [13]	Queue length	Derive the control expression based on an *M*/*M*/1 queue. Analytically tractable.	Unrealistic model assumptions.
Sahinoglu [23]	Queue length	Use the wavelet-decomposition approach. Both energy content in each sub-band frequency and queue length feedback are then used to determine the bandwidth allocation. Yield good performance.	Possibly high complexity in wavelet decomposition algorithms. No control over the target queue length.
Rampal [14], Hsu [15]	Loss	Proportional loss controller. Simple to implement.	Performance strongly depends on the feedback gain and can perform poorly.
Liao [16]	Loss	Map target loss to target utilization and control the target utilization. The estimated utilization is derived through an *M*/*M*/1/*K* queue assumption.	Require measurement of many queue statistics. Control derived from unrealistic model, leading to poor performance.
Siripong [17]	Loss	Map target loss to target average queue length and use fuzzy control to keep the queue length. Robust against input traffic conditions and control parameters.	Difficult control parameter tuning. Need accurate target queue length/loss mapping.
Adas [21]	Zero loss	Allocate bandwidth equal to the predicted rate from an LMS predictor.	LMS could give large prediction error. Simple to implement.
Duffield [22]	Zero loss	Allocate bandwidth equal to the predicted rate from a local Gaussian predictor. Simple to implement.	Ignore the presence of buffers, which may result in overallocation. Require some parameter tuning.
Chong [20]	Zero loss	Allocate bandwidth equal to the peak rate of low-pass filtered traffic. Yield good utilization because buffer is considered.	Require off-line work to determine proper cut-off frequency for filtering.
Kesidis [18]	Delay	Fit measured queue statistics to pre-specified queue length distribution, and then derive the changes in queue length as a function of bandwidth adjustment.	Assume *a priori* knowledge of queue length distribution. Subject to inaccuracy in extrapolation due to measurement error.
Christin [19]	Delay/rate	Guarantee QoS over a busy period, both absolute and relative manners, whereby the control is derived through the relation between arrival curve and output curve. Multiple QoS metrics can be guaranteed.	Computationally expensive.

■ **Table 1.** *ABC algorithms comparison.*

for attaining QoS at high utilization. Static bandwidth allocation techniques are typically based on some specific traffic models, and therefore are subject to errors in the underlying model as well as in the parameterization. ABC, on the other hand, requires minimal prior information on the traffic model. This is the major appeal of the ABC concept. At the same time, the choice of the control time scale is a critical issue in ABC schemes. Further, admission control and multi-queue control in the context of ABC are fertile research areas. In this section, a critical comparison of the various ABC schemes is presented with the pros and cons of each mechanism and directions for future research.

Table 1 summarizes the ABC algorithms discussed in this survey according to their QoS metrics guarantees, which follows the classification in Fig. 3. It may be noted that the bulk of the existing literature has focused on the loss QoS metric, with the remaining schemes focusing on the control of the queue length or the delay QoS metric. The queue length control mechanisms are developed with the intention to qualitatively achieve low packet loss rate and queuing delay. If quantitative guarantees on the loss or delay metrics are needed, then the ABC scheme must explicitly control the respective QoS metric. Due to the attention paid to the control of the loss QoS metric, these schemes are the most well developed, and understanding these schemes can provide insights into the control of other parameters as well. Thus, the rest of the section is spent comparing the ABC techniques to control loss and identifying the shortcomings that still need to be overcome.

The control techniques for loss are classified into those that target arbitrary non-zero loss values and those that target zero loss. The goal of the non-zero loss mechanisms is to maintain the cumulative, or long-term loss metric as close to the target as possible. However, maintaining the short-term loss QoS is very important for real-time audio and video applications. While the long-term loss is kept constant (and close to target), there may be periods of large burst losses occurring in between. The effectiveness of the ABC schemes on the short-term loss QoS needs to be studied. The second issue is the convergence rate, namely, how fast does the cumulative loss converge to the target value. The convergence rate depends on the choice of the different control parameters, and their relationship should be well understood. Also, there is limited work on the performance of ABC algorithms under non-stationary traffic, and traffic traces from real production networks need to be used.

The non-zero packet loss QoS schemes based on direct loss feedback control use an integral controller in adjusting the service rate. In this case, the control performance strongly depends on the choice of feedback gain and the control time scale. A single choice of the feedback gain and control time scale parameters will not work well in every configuration, and the selection of those parameters needs further analysis. The desired packet loss guarantee can also be achieved by maintaining the average queue length at some target value through the use of fuzzy control, which is shown to yield better results than direct loss feedback ABC. However, an effective method to translate a given loss requirement to the target average queue length in the control system is still an outstanding issue. If this issue is resolved, this would be a very promising approach to deploy.

When the target loss rate is very small, most non-zero loss guarantee schemes with feedback control become ineffective, and it is better to provide zero loss guarantee by allocating the bandwidth to match the input traffic rate that is obtained from online prediction. This comes at the expense of lower utilization. For traffic with widely varying bit rates over time such as MPEG-coded video or VBR traffic, ABC based on traffic prediction, or open-loop ABC, has been shown to satisfactorily improve the network utilization as compared to static allocation mechanisms. However, the online prediction of input traffic is a challenging problem and may incur inaccuracies that affect its performance. For the purposes of prediction, the incoming traffic may first be filtered. If the incoming traffic is left unfiltered, the prediction relies on simple linear predictors but the allocated bandwidth does not consider the buffer size, which could have increased the utilization. The prediction of filtered traffic at a properly selected cutoff frequency can yield higher utilization because it considers the buffer size. However, the cutoff frequency in most cases must be predetermined off-line, which incurs more complexity, and also requires some prior knowledge on the input traffic such as the range of its power spectrum. Consequently, the trade-off must be made between simplicity of control and bandwidth utilization. A simple linear predictor such as the local Gaussian predictor is appealing because it requires no prior traffic information, although the Gaussian assumption is made in deriving the predictor. In our opinion, it has the best performance trade-off overall. However, under non-Gaussian traffic, such a predictor requires some (manual) parameter tuning to increase the *safety margin* in allocating the bandwidth for a zero loss guarantee.

Acknowledgments

This research was supported in part by the NSF CAREER award ANI-9702389. We thank the anonymous reviewers for their insightful comments that significantly improved the quality of this paper.

References

[1] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks," *IEEE JSAC*, vol. 9, no. 7, Sept. 1991, pp. 968–81.

[2] F. Kelly, "Notes on Effective Bandwidths," *Stochastic Networks: Theory and Applications*, Clerendon Press, Oxford, 1996, pp. 141–68.

[3] H. S. Kim and N. B. Shroff, "Loss Probability Calculations and Asymptotic Analysis for Finite Buffer Multiplexers," *IEEE/ACM Trans. Net.*, vol. 9, no. 6, Dec. 2001, pp. 755–68.

[4] L. Breslau and S. Jamin, "Comments on the Performance of Measurement-Based Admission Control Algorithms," *Proc. IEEE INFOCOM'00*, Tel Aviv, Israel, Mar. 2000, pp. 1233–42.

[5] Z. Dziong, M. Juda, and L.G. Mason, "A Framework for Bandwidth Management in ATM Networks: Aggregate Equivalent Bandwidth Estimation Approach," *IEEE/ACM Trans. Net.*, vol. 5, no. 1, Feb. 1997, pp. 134–47.

[6] Q. Ren and G. Ramamurthy, "A Real-Time Dynamic Connection Admission Controller Based on Traffic Modeling, Measurement, and Fuzzy Logic Control," *IEEE JSAC*, vol. 18, no. 2, Feb. 2000, pp. 184–96.

[7] M. Grossglauser and D. Tse, "A Time-Scale Decomposition Approach to Measurement-Based Admission Control," *Proc. IEEE INFOCOM'99*, New York, NY, Mar. 1999, vol. 3, pp. 1539–47.

[8] P. Gevros *et al.*, "Congestion Control Mechanisms and the Best Effort Service Model," *IEEE Network*, vol. 15, no. 3, pp. 16–26, May/June 2001.

[9] C. V. Hollot *et al.*, "Analysis and Design of Controllers for AQM Routers Supporting TCP Flows," *Proc. IEEE INFOCOM'01*, Anchorage, AK, Apr. 2001, vol. 3, pp. 1726–34.

[10] R. Fengyuan *et al.*, "A Robust Queue Management Algorithm Based on Sliding Mode Variable Structure Control," Proc. *IEEE INFOCOM'02*, New York, NY, July 2002, vol. 1.

[11] U. Mocci, P. Pannunzi, and C. Scoglio, "Adaptive Capacity Management of Virtual Path Networks," *Proc. IEEE GLOBECOM'96*, Queen Elizabeth II Conf. Centre Westminster, London, 1996, pp. 750–54.

[12] R. C. Palazzo, "A Lyapunov Approach to Dynamic Bandwidth Allocation in B-ISDN," M.S. thesis, School of Electrical Engineering, Clemson University, 1994.

[13] A. Pitsillides, P. Ioannou, and L. Rossides, "Congestion Control for Differentiated Services using Non-linear Control Theory," *IEEE 6th Proc. Symp. Comp. and Commun.*, 2001, pp. 726–33.

[14] S. Rampal *et al.*, "Dynamic Resource Allocation Based on Measured QoS," Technical Report TR 96-2, Center for Advanced Computing and Commun., North Carolina State University, Jan. 1996.

[15] I. Hsu and J. Walrand, "Dynamic Bandwidth Allocation for ATM Switches," *J. Applied Probability*, vol. 33, no. 3, 1996, pp. 758–71.

[16] R. R.-F Liao and A. T. Campbell, "Dynamic Core Provisioning for Quantitative Differentiated Service," *Proc. IEEE/IFIP IWQoS'01*, Karlsruhe, Germany, June 2001.

[17] P. Siripongwutikorn, S. Banerjee, and D. Tipper, "Adaptive Bandwidth Control for Efficient Aggregate QoS Provisioning," *Proc. IEEE GLOBECOM'02*, Taipei, Taiwan, Nov. 2002.

[18] G. Kesidis, "Bandwidth Adjustments Using Online Packet-level Measurements," *SPIE Conf. Performance and Control of Network Systems*, Boston, MA, Sept. 1999.

[19] N. Christin, J. Liebeherr, and T. F. Abdelzaher, "A Quantitative Assured Forwarding Service," *Proc. IEEE INFOCOM'02*, New York, NY, June 2002.

[20] S. Chong, S.-Q. Li, and J. Ghosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM," *IEEE JSAC*, vol. 13, no. 1, pp. 12–23, Jan. 1995.

[21] A. M. Adas, "Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model," *IEEE/ACM Trans. Net.*, vol. 6, no. 5, Oct. 1998, pp. 635–45.

[22] N. G. Duffield *et al.*, "A Flexible Model for Resource Management in Virtual Private Networks," *Proc. ACM SIGCOMM'99*, Cambridge, MA, Oct. 1999, pp. 95–108.

[23] Z. Sahinoglu and S. Tekinay, "A Novel Approach Bandwidth Allocation: Wavelet-Decomposed Signal Energy Approach," *Proc. IEEE GLOBECOM'01*, San Antonio, TX, Dec. 2001.

[24] A. Pitsillides, J. Lambert, and D. Tipper, "Dynamic Bandwidth Allocation in Broadband ISDN using a Multilevel Optimal Control Approach," *Telecommun. Sys.*, vol. 4, no. 2, July 1995, pp. 71–95.

[25] A. Pitsillides, P. Ioannou, and D. Tipper, "Integrated Control of Connection Admission, Flow Rate, and Bandwidth for ATM based Networks," *Proc. IEEE INFOCOM'96*, San Francisco, CA, Mar. 1996, vol. 2, pp. 785–93.

[26] E. Fulp and D. Reeves, "Online Dynamic Bandwidth Allocation," *Proc. Int'l. Conf. Network Protocols*, Atlanta, Georgia, Oct. 1997, pp. 134–41.

[27] Y. Afek *et al.*, "Dynamic Bandwidth Allocation Policies," *Proc. IEEE INFOCOM'96*, San Francisco, CA, Mar. 1996, pp. 880–87.

[28] H. Zhang and E. W. Knightly, "RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive VBR Video," *Multimedia Systems*, vol. 5, 1997, pp. 164–76.

[29] E. W. Knightly and H. Zhang, "D-BIND: An Accurate Traffic Model for Providing QoS Guarantees to VBR Traffic," *IEEE/ACM Trans. Net.*, vol. 5, no. 2, Apr. 1997.

[30] R. G. Garroppo *et al.*, "Statistical Multiplexing of Self-Similar VBR Videoconferencing Traffic," *Proc. IEEE GLOBECOM'97*, Phoenix, AZ, Nov. 1997.

[31] J. R. Gallardo, D. Makrakis, and M. Angulo, "Dynamic Resource Management Considering the Real Behavior of Aggregate Traffic," *IEEE Trans. Multimedia*, vol. 3, no. 2, June 2001, pp. 177–85.

[32] S.-Q. Li, S. Chong, and C.-L. Hwang, "Link Capacity Allocation and Network Control by Filtered Input Rate in High-Speed Networks," *IEEE/ACM Trans. Net.*, vol. 3, no. 1, Feb. 1995, pp. 10–25.

[33] Y. Kim and S.Q. Li, "Timescale of Interest in Traffic Measurement for Link Bandwidth Allocation Design," *Proc. IEEE INFOCOM'96*, San Francisco, CA, Mar. 1996, pp. 738–48.

[34] G. Kesidis, J. Walrand, and C.-S. Chang, "Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources," *IEEE/ACM Trans. Net.*, vol. 1, no. 4, Aug. 1993, pp. 424–28.

[35] H. Fu and E. W. Knightly, "Aggregation and Scalable QoS: A Performance Study," *Proc. IEEE/IFIP IWQoS'01*, Karlsruhe, Germany, June 2001.

Biographies

PEERAPON SIRIPONGWUTIKORN (peerapon@mail.sis.pitt.edu) received the B.Eng. degree in telecommunications from King Mongkut's Institute of Technology, Ladkrabang, Thailand, in 1995, and the M.S. degree in telecommunications at the University of Pittsburgh in 1998. He is currently working toward the Ph.D. degree in telecommunications at the University of Pittsburgh. His current research interests are resource allocation and control, and QoS guarantees in communication networks.

SUJATA BANERJEE (sujata@exch.hpl.hp.com) is a senior research scientist at the Hewlett-Packard Laboratories and an adjunct associate professor of telecommunications and computer science at the University of Pittsburgh. She holds the Ph.D. degree in electrical engineering from the University of Southern California in Los Angeles and bachelors and masters degrees in electrical engineering from the Indian Institute of Technology, Bombay, India. Her research interests are in quality of service issues in networked systems. She is a recipient of the National Science Foundation CAREER award in Networking Research. She is a senior member of IEEE, and is serving as the technical program co-chair of the ACM MobiDE 2003 workshop.

DAVID TIPPER (dtipper@mail.sis.pitt.edu) is an associate professor in the Department of Information Science and Telecommunications at the University of Pittsburgh. Prior to joining Pitt in 1994 he was an associate professor of electrical and computer engineering at Clemson University in Clemson, SC. He is a graduate of the University of Arizona (Ph.D. EE, M.S. SIE) and Virginia Tech (B.S. EE). His current research interests are network design and traffic restoration procedures for survivable networks, network control (i.e., routing, flow control, etc.), performance analysis, and wireless and wired network design. Professor Tipper's research has been supported by grants from various government and corporate sources such as NSF, DARPA, NIST, IBM, and AT&T. He is a senior member of IEEE and has been on numerous conference technical committees, including serving as the technical program chair of DRCN 2003. He is currently a member of the editorial board of the *Journal of Network and Systems Management*.