A Real-time Loss Performance Monitoring Scheme

Guoqiang Mao

School of Electrical and Information Engineering, The University of Sydney

Abstract

Performance monitoring systems are becoming increasingly important in providing Quality-of-Service (QoS) based services and service guarantees. Performance monitoring can occur at different levels and different timescales. Either passive measurements and active measurements can be employed. Currently, a large amount of work has gone into developing mechanisms and protocols for performance and traffic monitoring at network level and large timescales. In comparison, little work has been done to use measurement information at small timescales for tackling network performance degradation and managing congestion in operational networks in real time. In this paper, we investigate a real-time loss performance monitoring and estimation scheme based on virtual buffer techniques. The proposed scheme is used to perform measurements at lower level, i.e. network node level, and small timescales, which can be used to provide real-time QoS information for network traffic and congestion control in ATM networks. The proposed scheme only requires modest computation resources. Virtual buffer techniques enable the scheme to obtain QoS information in a much reduced monitoring period, which is important for instantaneous network traffic and congestion control actions.

Key words: quality of service, monitoring, cell loss ratio

1 Introduction

Bandwidth hungry computer and communication applications are on the rise with a variety of services, as a few examples, voice over IP, video on demand, high-definition television, etc. Different from traditional data communication applications, these applications are real-time applications and they have stringent quality-of-service (QoS) requirements. It has been a big challenge for

Email address: g.mao@ieee.org (Guoqiang Mao).

network operators to satisfy the QoS requirements of these applications and provide value-added services to the customer. Performance monitoring systems are becoming increasingly important in providing QoS based services and service guarantees.

A monitoring system can provide information for the following three categories of tasks [1]:

- Assist traffic engineering in making provisioning decisions for optimizing the usage of network resources according to small to medium timescale changes. A real-time performance monitoring system can provide information for traffic and congestion control, routing and re-routing, load balancing, etc.
- Assist traffic engineering in providing analyzed traffic and performance information for preventive, reactive and predictive maintenance, as well as long-term network planning and design in order to optimize network resource usage and avoid undesirable conditions. The analyzed information includes traffic growth patterns, bottleneck links, and the location and causes for the performance degradation.
- Verify whether the performance guarantees committed in service level agreements (SLA) are in fact being met. SLAs are becoming more and more popular as part of the efforts of Internet Service Providers (ISP) to provide better QoS to the customer.

To achieve these tasks, monitoring at different levels and different timescales is required. Monitoring can be used to derive packet level, connection level, traffic class level, node level, and network-wide level information. Measurements can occur at small timescales (e.g. millisecond, second) and large timescales (e.g. hourly, daily and yearly). Monitoring at lower levels and smaller timescales is mainly useful for real-time traffic and congestion control and monitoring at network level and larger timescales can be used for long-term network planning and design. Monitored information includes delay, packet delay variation, packet loss, traffic load and throughput, etc.

Measurements in a monitoring system can be classified into two categories: active and passive measurements. Active measurements inject synthetic traffic into the network based on scheduled sampling (by sending probing packets) to observe network performance. Passive measurements are used to observe actual traffic without injecting extra traffic into the network. The capability of active measurements is often limited by constraints on network capacity consumed by synthetic traffic. Practically, the rule used by some network operators is that synthetic traffic should not exceed approximately 1% of the total network capacity [1]. Moreover, due to the statistical and dynamic (There are always connections in and out in a large-scale network.) nature of the network traffic and long time delay involved in observing the probing packets, it is difficult for active measurements to obtain real-time monitoring information

and accurate estimates of QoS parameters. Passive measurements observe the network in a non-intrusive manner and do not constitute a burden on network capacity. However, on high-speed links, the large amount of user traffic that needs to be observed constitutes challenges on the design of passive measurement schemes.

In this paper, we investigate a real-time loss performance monitoring and estimation scheme based on virtual buffer techniques. The proposed scheme is used to perform measurements at lower level, i.e. network node level, and small timescales in ATM networks. It is used to provide real-time QoS information for network traffic and congestion control to allow them to respond dynamically to changing traffic pattern and improve QoS and network resources utilization (i.e. bandwidth, buffer). The proposed scheme is efficient and only requires modest computation resources. Virtual buffer techniques enable the scheme to obtain QoS information in a much reduced monitoring period, which is important for real-time network traffic and congestion control actions.

The rest of the paper is organized as follows: in section 2, the principle of the monitoring scheme will be introduced; in section 3, we shall establish the relationship between buffer size and traffic loss; in section 4, the design of the monitoring scheme will be explained; simulation results will be presented in section 5; finally, some conclusions and further research will be discussed in section 6.

2 Virtual buffer based monitoring scheme

Currently, a large amount of work has gone into developing mechanisms and protocols for performance and traffic measurements [1]. The best known is the work of Internet Engineering Task Force (IETF) working group IP Performance Metrics (IPPM) [1], which develops a set of standard metrics that can be applied to the quality, performance, and reliability of Internet services. Other activities include the RIPE Network Coordination Center [2] that has implemented a number of the measurement metrics developed by IPPM; the CAIDA [3] (Cooperative Association for Internet Data Analysis) that develops tools, such as cflowd, CoralReef and skitter, for traffic analysis and monitoring; the NLANR [4] (National Laboratory for Applied Network Research) measurement and network analysis group that develops the Network Analy-

[1] See http://www.ietf.org.
[2] See http://www.ripe.net.
[3] See http://www.caida.org.
[4] See http://www.nlanr.net.

sis Infrastructure (NAI); and NIMI [5] (National Internet Measurement Infrastructure) that combines active and passive measurements for network path monitoring. The main goals of these measurement efforts are to understand the complexity of high-speed large scale networks, traffic trends and load patterns. Their common practice on delay analysis is to obtain daily or hourly minimum, maximum, standard deviation and percentiles of end-to-end paths. These statistics certainly can reflect the traffic trends and load patterns on large timescales, therefore, are very useful for network long-term planning. However, small timescale monitoring may also be desirable for both users and network operations. In comparison, little work has been done to use measurement information at small timescales for tackling network performance degradation and managing congestion in operational networks in real time.

Garcá-Hernández *et al.* propose an active monitoring method using Operation, Administration and Management (OAM) cells [2]. Siler *et al.* propose a passive monitoring method by fitting a distribution function to observed packet delays [3]. However one potential problem for real-time monitoring scheme is that some QoS indicators are specified in terms of the probability of occurrence of certain rare events. For example, in ATM networks, cell loss ratio (CLR), which is the ratio of lost cells to the total transmitted cells, is often specified to be as small as 10^{-9}. Monitoring using direct statistical methods is impractical for such small CLR. As an example, in an OC3 link with a link utilization of 0.5 and a cell loss ratio of 10^{-9}, at least 10 billion cells have to be observed before any statistically meaningful information can be collected [6]. This will take 15 hours. The statistical information obtained after such long monitoring period may be obsolete and the network management system's reaction may be too late. Even in a network with moderate traffic loss, a monitoring scheme with a much reduced monitoring period than the direct monitoring method is still desirable to provide faster system response.

Zhu *et al.* propose an in-service QoS monitoring and estimation method, which is used to obtain a real-time estimation of CLR [4]. Their method is based on the asymptotic relationship between CLR and buffer size. The observed cell loss ratios of several small virtual buffers are used to estimate the CLR of the actual system. However, their method still requires a long monitoring period and the accuracy of their method is not good.

In this paper, we shall present a real-time loss performance monitoring and estimation scheme that employs simple measurements and computations. The proposed scheme is based on the virtual buffer techniques. The basic idea of virtual buffer techniques is to use the traffic loss observed from several

[5] See http://ncne.nlanr.net/nimi/.

[6] It is assumed that at least 10 cell loss samples have to be observed to obtain any statistically meaningful information

virtual buffers with much smaller buffer sizes than that of the real buffer to estimate the traffic loss of the real system. Virtual buffers have the same input and output as the real buffer. Since virtual buffers have much smaller buffer sizes, the traffic loss in the virtual buffers is much larger than that of the real buffer. In order to obtain statistical meaningful observations of the traffic loss of these virtual buffers, fewer packets need to be observed compared to those required for a large real buffer. Hence the observation of the traffic loss of the virtual buffers requires much less monitoring period. Then based on the asymptotic relationship between traffic loss and buffer size, the traffic loss observed in these virtual buffers are used to obtain an estimate of traffic loss in the real buffer. Therefore much less monitoring period is required in our algorithm to obtain an estimate of traffic loss in the real buffer than that using direct monitoring method. Arguably the virtual buffer based monitoring scheme may also have better accuracy than direct monitoring method when non-stationarity of network traffic is considered. In a real network, there are always connections in and out. Thus network traffic is always changing. For direct monitoring methods requiring a large monitoring period, the obtained loss information may not make any sense for real time traffic and congestion control because the traffic may have changed dramatically during the period. Fig. 1 shows the system model of the virtual buffer based monitoring scheme.

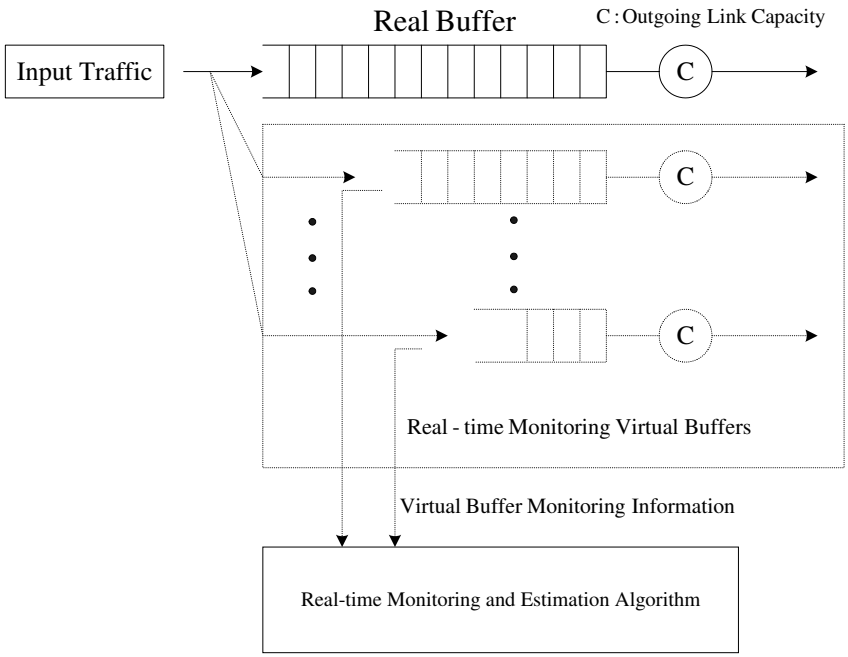

Fig. 1. System model of the monitoring scheme

3 Asymptotic Relationship Between Traffic Loss and Buffer Size

In this section we shall establish the validity of using a generalized relationship between buffer size and the logarithm of traffic loss, denoted by $log(clr)$. Analytical results in the literature will be reviewed and summarized to demonstrate this relationship for both Markovian traffic and long-range dependent traffic.

3.1 Markovian Traffic Process

For Markovian traffic, $log(clr)$ is known to decrease proportionally with increasing buffer size. This linear relationship is a natural outgrowth of the fluid flow analysis of network traffic [5]. This relationship is also widely used in connection admission control schemes based on the effective bandwidth approach for bandwidth allocation.

Based on fluid flow model, Anick *et al.* show that for an infinite queue fed by N homogeneous exponential on-off sources, and drained at a rate c, where c is normalized by the peak rate of the on-off source, when the buffer size x becomes large, the buffer overflow probability is approximately given by [5]:

$$G(x) \sim \rho^N \left\{ \prod_{i=1}^{N-\lfloor c \rfloor - 1} \frac{z_i}{z_i - z_0} \right\} e^{z_0 x}. \tag{1}$$

where z_0 is given by: $z_0 = -\frac{(1-\rho)(1+\lambda)}{1-c/N}$, ρ is link utilization and λ is the ratio of the average on period to the average off period of the on-off source. z_i are negative roots of the following quadratics by setting $k = i$:

$$A(k)z^2 + B(k)z + C(k) = 0, \qquad k = 0,\, 1,\, \ldots, N$$

where

$$\begin{aligned}
A(k) &= \left(\frac{N}{2-k}\right)^2 - \left(\frac{N}{2-c}\right)^2 \\
B(k) &= 2(1-\lambda)\left(\frac{N}{2-k}\right)^2 - N(1+\lambda)\left(\frac{N}{2-c}\right) \\
C(k) &= -(1+\lambda)^2 \left\{ \left(\frac{N}{2}\right)^2 - \left(\frac{N}{2-k}\right)^2 \right\}.
\end{aligned}$$

Buffer overflow probability is often used as an approximation of CLR. Thus, Eq. 1 indicates an asymptotic linear relationship between buffer size and $log(clr)$.

This linear relationship is shown to hold within a much more general context by Elwalid *et al* [6], [7]. Based on theoretical analysis using large deviations theory and simulation validation, Elwalid *et al.* propose the following asymptotic relationship between buffer size B and $log(clr)$ for general Markovian traffic:

$$log(clr) \sim -\alpha - \delta B \tag{2}$$

where α and δ are both positive constants determined by the traffic process and the link capacity. This generalized result has been used in many situations to develop algorithms for connection admission control and routing in ATM networks [7], [8].

3.2 Long-Range Dependent Traffic Model

A lot of research shows convincingly that long-range dependence exists extensively in network traffic, especially in video traffic [9], [10], [11]. For long-range dependent traffic process, traffic loss decays more slowly with increasing buffer size, and the linear relationship between buffer size and $log(clr)$ no longer exists.

Taqqu *et al.* show that the superposition of many on-off sources with strictly alternating on and off periods, and whose on or off periods exhibit the *Noah effect* (i.e. high variability or infinite variance) produces an aggregate traffic that exhibits the *Joseph effect* (i.e. self-similarity or long-range dependence) [12]. They also present extensive statistical analysis of high time-resolution Ethernet LAN traffic traces, which confirms that data at the level of individual sources or source-destination pairs are consistent with the Noah effect. In another research on heavy-tailed on-off sources by Heath *et al.* , they show independently that on-off sources with heavy-tailed on and/or off distributions lead to strong long-range dependence in the aggregate traffic [13].

Likhanov *et al.* consider the superposition of a large number of on-off sources with Pareto distributed (heavy-tailed) active (on) periods [14]. For M i.i.d. on-off sources, the aggregate traffic converges to a self-similar traffic process as $M \to \infty$. Using queueing theory, they show that the overflow probability of the aggregate traffic has an asymptotic relationship with buffer size B:

$$Pr(buffer\,content > B) \sim \alpha B^{\beta} \tag{3}$$

where α and β are constants determined by the traffic process. Parulekar *et al.* [15] obtain the same asymptotic relationship using large deviation theory and the $M/G/\infty$ model for self-similar process [16].

In addition to the above models for self-similar traffic, Fractional Brownian Noise (FBN) model is also used for self-similar traffic modelling. The FBN

model results in the following generalized relationship between overflow probability and buffer size [15], [17]:

$$Pr(buffer\,content > B) \sim \exp\left(-\delta B^{\gamma}\right) \tag{4}$$

where δ and γ are constants determined by the traffic process.

It can be argued that some self-similar traffic models are more appropriate than others. However, as a quick review of the existing literature indicates, all traffic models have provided good fit for diverse applications. Here we choose Eq. 3 as the asymptotic relationship between overflow probability and buffer size. That is, we consider that $log(clr)$ has the following asymptotic relationship with buffer size:

$$log(clr) \sim log(\alpha) + \beta log(B). \tag{5}$$

Eq. 3 is chosen because although Gaussian distribution provides a good fit for the distribution of the aggregate traffic when the number of connections is very large, the CLR estimate resulting from Gaussian distribution is usually too optimistic, especially when the aggregate traffic distribution deviates from Gaussian. The under-estimated CLR may compromise our efforts in providing robust QoS guarantees.

4 Design of the monitoring scheme

In the virtual buffer based monitoring scheme, since the size of the virtual buffer is smaller than that of the real buffer, a virtual buffer can be simply implemented as a counter. The value of the counter varies between a threshold, which is the size of the virtual buffer, and zero. The counter is incremented by one with each cell arrival. When the threshold is reached, further cell arrivals are considered as lost and the counter remains constant at the threshold. At the same time, the counter is decremented by one with each cell departure in the real buffer. Therefore the implementation of virtual buffers will not add too much burden onto the real system.

Four virtual buffers are employed in the algorithm. Denote by B_1, B_2, B_3, B_4 the sizes of the four virtual buffers. B_1, B_2, B_3, B_4 are chosen such that:

$$B_1 < B_2 < B_3 < B_4 << B,$$

where B is the size of the real buffer. Denote by clr_t^1, clr_t^2, clr_t^3 and clr_t^4 the cell loss ratios observed in the four virtual buffers 1, 2, 3 and 4 respectively at discrete time t. These cell loss ratios of the virtual buffers are used to obtain an estimate of the CLR in the real buffer, which is denoted by $\widehat{clr_t}$.

For Markovian traffic, traffic loss has an asymptotic relationship with buffer size as given in Eq. 2. Using this relationship, given the cell loss ratio observations in the virtual buffer 1 and virtual buffer 2, the logarithm of cell loss ratio in a buffer of size x can be estimated:

$$\log\left(\widehat{clr_t^x}_M\right) = a + b \times x \tag{6}$$

where $\widehat{clr_t^x}_M$ denotes the CLR estimate in a buffer with buffer size x at time t, which is estimated from the CLR-buffer size relationship for Markovian traffic,

$$a = \frac{\log\left(clr_t^1\right) \times B_2 - \log\left(clr_t^2\right) \times B_1}{B_2 - B_1}$$
$$b = \frac{\log\left(clr_t^2\right) - \log\left(clr_t^1\right)}{B_2 - B_1}.$$

The logarithm of cell loss ratio in a buffer with buffer size x can also be estimated using the CLR-buffer size relationship for self-similar traffic given in Eq. 5:

$$\log\left(\widehat{clr_t^x}_S\right) = c + d\log\left(x\right) \tag{7}$$

where $\widehat{clr_t^x}_S$ denotes the CLR in a buffer with buffer size x at time t, which is estimated from the CLR-buffer size relationship for self-similar traffic,

$$c = \frac{\log\left(clr_t^1\right)\log\left(B_2\right) - \log\left(clr_t^2\right)\log\left(B_1\right)}{\log\left(B_2\right) - \log\left(B_1\right)}$$
$$d = \frac{\log\left(clr_t^2\right) - \log\left(clr_t^1\right)}{\log\left(B_2\right) - \log\left(B_1\right)}.$$

In our algorithm, the CLR of the real buffer is estimated using Eq. 8:

$$\log\left(\widehat{clr_t}\right) = \alpha \times \log\left(\widehat{clr_t^B}_M\right) + \beta \times \log\left(\widehat{clr_t^B}_S\right), \tag{8}$$

where α and β are non-negative constants estimated using the cell loss ratio observations in the virtual buffer 3 and 4. Here a simple multivariate linear model [18] is used. Together with the least square algorithm for estimating α and β, it allows the CLR estimation algorithm to adjust parameters α and β dynamically to select the best equation to use. Values of α (β) close to 1 indicate Eq. 2 (Eq. 5) for Markovian traffic (self-similar traffic) is a better choice for estimating CLR in the real buffer. Values of α (β) close to 0 indicate no relationship. See Fig. 10 for an example. An advantage of using the linear combination of Eq. 2 and Eq. 5 rather than selecting one of them is both Eq. 2 and Eq. 5 are asymptotic results only. Moreover real traffic conditions are usually more complex than the ideal models assumed in the theoretical analysis leading to Eq. 2 and Eq. 5. Therefore, the $log(clr)$ - buffer size relationship

often deviates from that given in Eq. 2 and Eq. 5. Eq. 8 allows us to have the flexibility to adjust parameter α and β dynamically to compensate for inaccuracy in Eq. 2 and Eq. 5 and achieve better estimation.

It should be noted that due to the bursty nature of traffic, a cell loss ratio of zero can often be observed in these virtual buffers. A cell loss ratio of zero becomes a problem in our CLR estimation algorithm using Eq. 8. In our algorithm this problem is dealt with in the following way: if any of the virtual buffers has a cell loss ratio of zero, the cell loss ratio in the real buffer is estimated to be zero too.

4.1 Estimation of Parameters α and β

Based on the CLR-buffer size relationship for Markovian traffic, estimates of the logarithm of cell loss ratios in virtual buffers with size B_3 and B_4 at time t can be obtained using Eq. 6. Denote them by $\log\left(\widehat{clr_t^{B_3}}_M\right)$ and $\log\left(\widehat{clr_t^{B_4}}_M\right)$ respectively. Based on the CLR-buffer size relationship for self-similar traffic, estimates of the logarithm of cell loss ratios in buffers with size B_3 and B_4 at time t can be obtained using Eq. 7. Denote them by $\log\left(\widehat{clr_t^{B_3}}_S\right)$ and $\log\left(\widehat{clr_t^{B_4}}_S\right)$ respectively. These estimated cell loss ratio values and the observed cell loss ratio values for the virtual buffers 3 and 4 are taken as input-output pairs to estimate the parameters α and β in Eq. 8. For ease of expression, denote $\log\left(\widehat{clr_t^{B_3}}_M\right)$ and $\log\left(\widehat{clr_t^{B_4}}_M\right)$ by x_t^3 and x_t^4 respectively, and denote $\log\left(\widehat{clr_t^{B_3}}_S\right)$ and $\log\left(\widehat{clr_t^{B_4}}_S\right)$ by y_t^3 and y_t^4 respectively. The logarithm of observed cell loss ratios in the virtual buffers 3 and 4, $\log\left(clr_t^3\right)$ and $\log\left(clr_t^4\right)$, are denoted by z_t^3 and z_t^4 respectively. $\left(\left(x_t^3,\, y_t^3\right),\, z_t^3\right)$ and $\left(\left(x_t^4,\, y_t^4\right),\, z_t^4\right)$ are used as input-output pairs in Eq. 9 to obtain estimates of non-negative parameters α and β used in Eq. 8:

$$z = \alpha x + \beta y. \tag{9}$$

In order to increase the precision of the estimation, past CLR information is taken into account in estimating parameters α and β. Specifically, CLR information from time interval $(t-K,\, t]$ is used in estimating α and β. There are altogether $2K$ input-output pairs $\left(\left(x_{t-K+1}^3,\, y_{t-K+1}^3\right),\, z_{t-K+1}^3\right)$, $\left(\left(x_{t-K+1}^4,\, y_{t-K+1}^4\right),\, z_{t-K+1}^4\right)$, ..., $\left(\left(x_t^3,\, y_t^3\right),\, z_t^3\right)$, $\left(\left(x_t^4,\, y_t^4\right),\, z_t^4\right)$ that are used to estimate the parameters α and β according to Eq. 9. Least-square algorithm is employed to obtain the optimum parameters $\alpha \geq 0$ and $\beta \geq 0$ which minimize the error term:

$$f(\alpha,\beta)=\sum_{i=0}^{K-1}[(z_{t-i}^{3}-\alpha x_{t-i}^{3}-\beta y_{t-i}^{3})^{2} + (z_{t-i}^{4}-\alpha x_{t-i}^{4}-\beta y_{t-i}^{4})^{2}]$$

Solving the equations:

$$\begin{cases} \frac{\partial f(\alpha,\beta)}{\partial \alpha}=0 \\ \frac{\partial f(\alpha,\beta)}{\partial \beta}=0 \end{cases},$$

the optimum α and β which minimize the error term $f\,(\alpha,\,\beta)$ can be obtained:

$$\begin{cases} \alpha=\frac{BE-AC}{DE-A^{2}} \\ \beta=\frac{CD-AB}{DE-A^{2}} \end{cases}, \tag{10}$$

where

$$A=\sum_{i=1}^{K-1}\left(x_{t-i}^{3}y_{t-i}^{3}+x_{t-i}^{4}y_{t-i}^{4}\right)$$
$$B=\sum_{i=1}^{K-1}\left(x_{t-i}^{3}z_{t-i}^{3}+x_{t-i}^{4}z_{t-i}^{4}\right)$$
$$C=\sum_{i=1}^{K-1}\left(y_{t-i}^{3}z_{t-i}^{3}+y_{t-i}^{4}z_{t-i}^{4}\right)$$
$$D=\sum_{i=1}^{K-1}\left(\left(x_{t-i}^{3}\right)^{2}+\left(x_{t-i}^{4}\right)^{2}\right)$$
$$E=\sum_{i=1}^{K-1}\left(\left(y_{t-i}^{3}\right)^{2}+\left(y_{t-i}^{4}\right)^{2}\right).$$

It is often the case that Eq. 10 results in a negative α or β. Care must be taken when this happens. If parameter α given in Eq. 10 is negative, then Eq. 11 is used to obtain the optimum non-negative parameters α and β:

$$\begin{cases} \alpha=0 \\ \beta=\frac{C}{E} \end{cases}. \tag{11}$$

Or, if parameter β given in Eq. 10 is negative, then Eq. 12 is used to obtain the optimum non-negative parameters α and β:

$$\begin{cases} \alpha=\frac{B}{D} \\ \beta=0 \end{cases}. \tag{12}$$

The parameter K is determined as:

$$K = \min\{J,\, M\},$$

where J is the maximum integer such that the cell loss ratios observed in the virtual buffers in the interval $(t - J,\, t]$ are all non-zero values, and M is an integer determined by the CLR estimation algorithm. Therefore there are a maximum of $2M$ input-output pairs that are used by the least square algorithm in estimating the parameters α and β using Eq. 9, which enables us to obtain accurate estimates of α and β.

4.2 Low-pass FIR Filter

In our CLR estimation algorithm, CLR estimated using Eq. 8 is not used directly as a CLR estimate of the real buffer. CLR estimated from Eq. 8 is used as an input to a low-pass filter. It is the output of the low-pass filter that is used as the CLR estimate for the real buffer.

Here the application of the low-pass filter does not change the asymptotic relationship between cell loss ratio and buffer size. Cell loss ratio is a statistical quantity. An accurate measurement of the cell loss ratio can only be obtained when measurement is taken over a long period and large enough samples are collected. However, in our system, measurement and estimation is taken over a much shorter time period. As a result, the measurement is easily affected by some temporary surge in traffic rate, which does not reflect the true value of the cell loss ratio. The application of a low-pass filter helps to remove this effect and obtain the moving average of the estimated cell loss ratio, which is a better representation of the true value.

To make it clearer, let us consider the following example. In a system with a CLR of 10^{-3}, in order to obtain an accurate measurement of the CLR, the number of cells observed must be at least $10,000$. However, if a short observation period, say $1,000$ cells, is chosen, due to the statistical nature of CLR, it may happen that in one $1,000$-cell period, no cell loss occurs; in another $1,000$-cell period, multiple cell losses occur. Therefore the observed CLR will fluctuate significantly. This fluctuation will become more severe with the reduction of the observation period. In this situation, the moving average will be a better representation of the true value.

The low-pass filter used in our algorithm is a FIR (Finite Impulse Response) filter of the form:

$$y(t) = \sum_{i=0}^{j} b_i \times x(t - i).$$

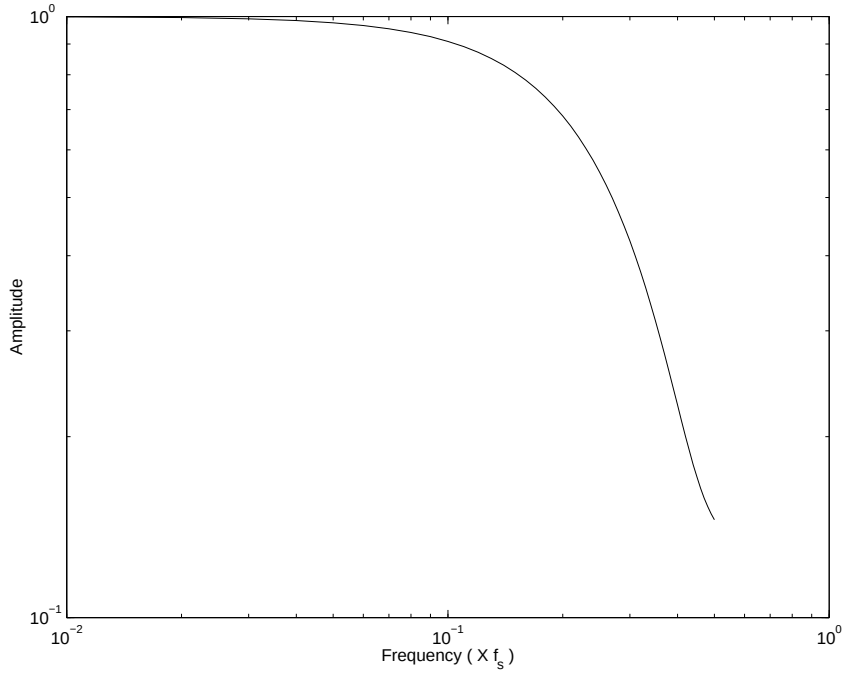

Fig. 2. Amplitude-frequency response of the FIR filter

Fig. 2 shows the amplitude-frequency response of the FIR filter. In Fig. 2, f_s denotes the inverse of the CLR monitoring period in our algorithm.

Our simulation results show that the application of the low-pass filter improves the performance of the proposed algorithm significantly. Otherwise, the estimated CLR will fluctuate dramatically around the true value.

Fig. 3 shows the architecture of the CLR estimation algorithm.

4.3 Choice of Parameters

There are several critical parameters in the CLR estimation algorithm that need to be determined.

Parameter M determines the maximum number of past CLR information taken into account in the least square algorithm for estimating parameters α and β. Taking into account some past CLR information helps to improve the precision of the estimation. However as more information is taken into account, more memory is required to store these past information and more computation resources is required in the least square algorithm. Moreover, parameters α and β are to some extent indicators of the nature of network traffic (Markovian or self-similar). They change when network traffic changes. When too much past information is taken into account, it may restrict the ability of the algorithm to track variations in network traffic, which in turn

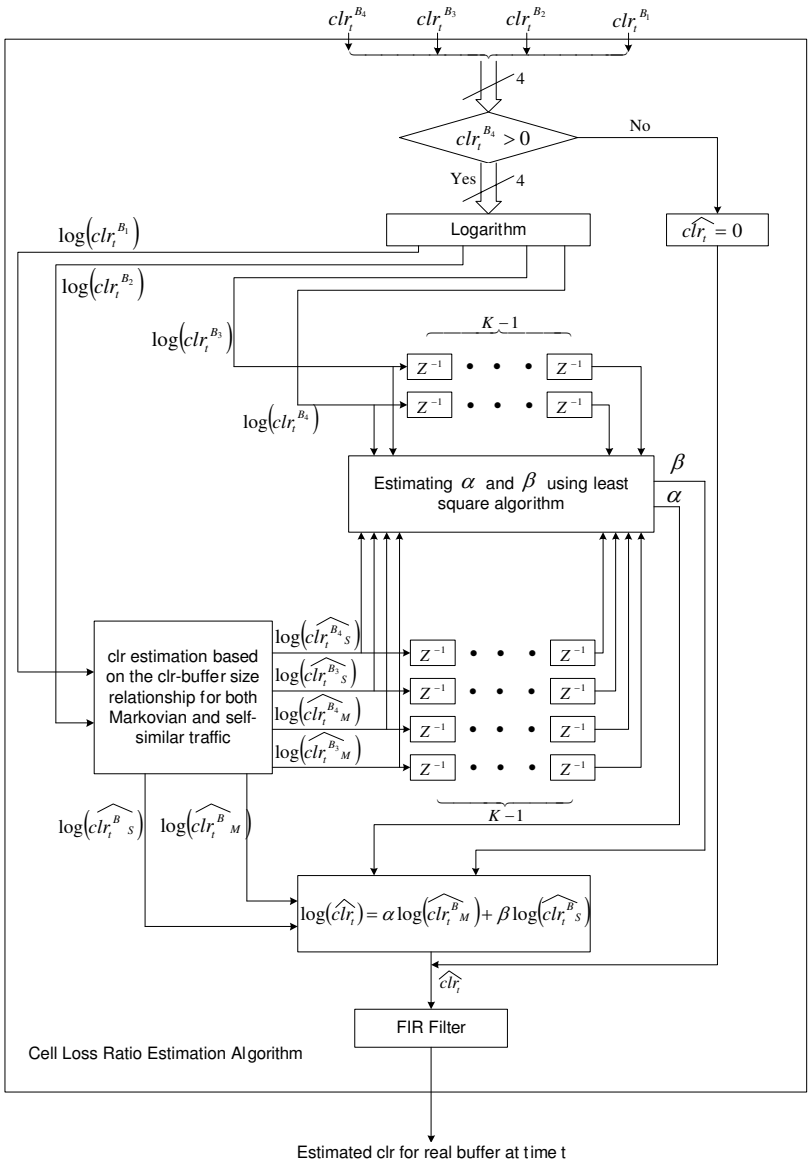

Fig. 3. Architecture of the CLR estimation algorithm

decreases the precision of the estimation. In our algorithm, M is chosen to be 10. That is, there is a maximum number of 20 input-output pairs used in the least square algorithm for estimating α and β.

The sizes of the virtual buffers are also critical parameters which determine the performance of the CLR estimation algorithm. The smaller the sizes of the virtual buffers are, the larger the cell loss ratios in these virtual buffers. Hence less monitoring period is required to make a CLR estimation when virtual buffer sizes are small. It is therefore desirable that the sizes of the virtual buffers are chosen to be as small as possible. On the other hand, the CLR estimation algorithm is based on the asymptotic relationships between CLR and buffer size. These asymptotic relationships only apply for large buffers,

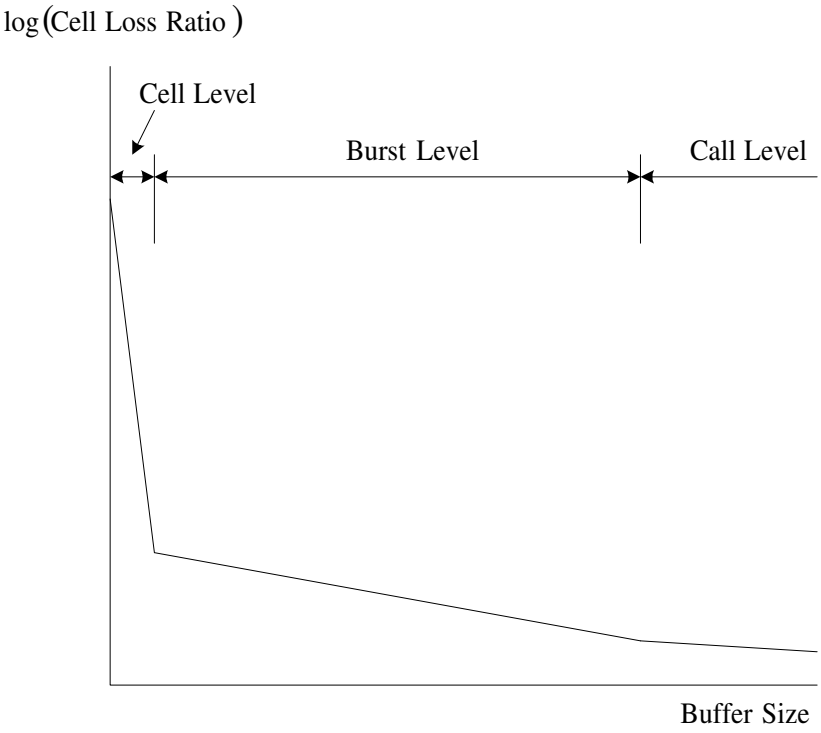

Fig. 4. Cell level, burst level and call level cell loss ratio

therefore the sizes of virtual buffers can not be chosen to be too small.

Hui suggests that congestion should be evaluated at different levels, namely, the cell level, the burst level and the call level [19]. Accordingly the CLR-buffer size relationship can be approximately divided into three regions, namely cell level region, burst level region and call level region. This has been verified by a lot of experimental studies [20], [21], [22], [23]. Fig. 4 illustrates the three regions.

When buffer size falls into the cell level region, cell loss occurs because of the simultaneous arrivals of cells from independent sources. However the aggregate traffic rate of these independent sources may be below the link capacity. When network buffer size falls into the burst level region, cell loss occurs when the aggregate traffic rate is momentarily greater than the link capacity. Buffer content grows to the limit that cell loss occurs, as long as the aggregate rate excess exists. When network buffer size falls into the call level region, cell loss occurs because excessive number of connections are admitted into the network. Therefore aggregate traffic rate exceeds link capacity for a large timescale (comparable to the connection duration time). This different nature of cell loss determines that CLR-buffer size relationship is different in different regions.

The boundary of cell level region is a non-decreasing function of the number of connections on the link. Our empirical observations show that the boundary of cell level region usually varies between a buffer size of several ATM

cells and a size of up to 20 cells, depending on the number of connections on the link. This conforms to the observation reported in [22]. Burst level region spans from several ATM cells to several thousands of cells, depending on the system size (e.g. number of connections, link rate, etc.). The size of the real buffer usually lies in this region. Moreover, when analyzing the asymptotic relationship between CLR and buffer size, a lot of researchers adopt the fluid flow model [7], [15], [17], and their loss performance analysis naturally rests on the burst level. At last, no call level dynamics are considered in the aforementioned analysis of the asymptotic relationship between CLR and buffer size. Therefore, the asymptotic relationship between CLR and buffer size actually applies for a buffer size in the burst level region. Accordingly the sizes of virtual buffers should be chosen in the burst level region.

In our CLR estimation algorithm, for a small system where the number of connections is small (e.g. $<$10), the minimum virtual buffer size B_1 is chosen to be 10 cells. For a large system where there are a lot of connections on the link, the minimum virtual buffer size B_1 is chosen to be 20 cells. Sizes of virtual buffers are chosen to be 5 cells apart.

Another important parameter in our CLR estimation algorithm is the CLR monitoring period, which is the minimum time required to make valid CLR observations in the virtual buffers. There are a lot of factors that affect the CLR monitoring period, including the maximum size of the virtual buffers, the size of the real buffer, link capacity, link utilization, the resolution of the CLR estimation algorithm (the minimum cell loss ratio that can be estimated by the algorithm), etc. This makes it difficult to formulate the process of determining the CLR monitoring period. The CLR monitoring period is determined empirically. Here we try to illustrate the process through the following example. Consider a case where the CLR estimation algorithm is designed to estimate a cell loss ratio as small as 10^{-9} in a system with a buffer size of $1,000$ cells. The link capacity is $150Mbps$. The maximum virtual buffer size B_4 is chosen to be 35 cells in our CLR estimation algorithm. Our empirical observations show that when the CLR in the real buffer is around 10^{-9}, the cell loss ratio in the virtual buffer B_4 usually varies in the range from 10^{-3} to 10^{-5}. Assume that when the cell loss ratio in the real buffer is 10^{-9}, the lowest link utilization is 0.5 (when the link utilization is too low, no cell loss will occur in the real buffer). Then, if a CLR monitoring period of $50s$ is chosen, it implies that a minimum number of 88 cell losses can be observed in the monitoring period, which is enough to generate a valid observation of CLR in the virtual buffers. Therefore for this specific example, the CLR monitoring period can be chosen to be approximately $50s$. The CLR monitoring period is significantly reduced compared with the CLR monitoring period of 15 hours required using direct monitoring method.

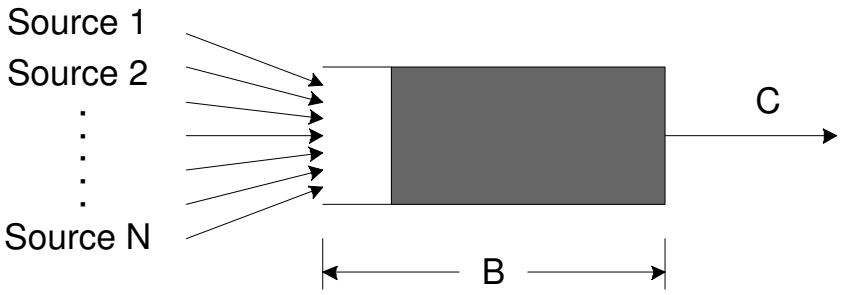

Fig. 5. Simulation model for the CLR estimation algorithm

5 Simulation Study

In this section, we investigate the performance of the CLR estimation algorithm using simulation. Both on-off traffic source model and real variable bit rate (vbr) video traces are used in the simulation. The objectives of the simulation are to:

- validate the effectiveness of the CLR estimation algorithm for a variety of traffic sources, and
- evaluate the accuracy of the CLR estimation algorithm compared to other algorithms in the literature.

The simulation is performed using OPNET. Fig. 5 illustrates the simulation model. There are N traffic sources in the simulation. The traffic generated by the traffic source is encapsulated into ATM cells using AAL5 protocol and then transported into a single server queue drained at a speed equal to the link capacity C. The queue has a size of B. The link capacity C is engineered such that an average bandwidth utilization of 0.8 is achieved in the simulation. This high link utilization is chosen to make CLR in the real buffer easier to observe so that the accuracy of the proposed ISME scheme can be validated.

Two typical scenarios are considered in the simulation. In the first scenario, referred to as the Markovian scenario, exponential on-off sources are used. In the second scenario, referred to as the Self-similar scenario, real vbr video traffic sources, which are known to be self-similar, are used. Since Markovian traffic and self-similar traffic are the two most typical traffic processes, these two scenarios are used to establish the effectiveness of our CLR estimation algorithm for a variety of traffic sources.

5.1 Markovian Scenario

30 exponential on-off sources, which are typical Markovian traffic sources, are used in this scenario. Each on-off source has independent exponentially

distributed on and off periods with means τ and γ respectively. The peak cell rate of the on-off source is pcr and the mean cell rate of the on-off source is mcr. Table 1 shows the traffic parameters of the exponential on-off source. These parameters are typical parameters of telephone voice traffic [24].

Table 1
Traffic parameters of the exponential on-off source

pcr (kb/s)	mcr (kb/s)	τ (sec)	γ (sec)
64	22	0.384	0.734

Seven simulations are run in the Markovian scenario with buffer size B varying from 100 cells to 700 cells. Each simulation is run for $10,000$ seconds. When buffer size B is increased above 700 cells, it becomes difficult to make valid cell loss ratio observations in the real buffer within the limited simulation time. In our simulation, the CLR of the real buffer and the estimated CLR are observed and compared to investigate the performance of the CLR estimation algorithm. As a typical example, Fig. 6 shows the CLR of the real buffer and our estimation using a simulation where buffer size B is 300 cells. For comparison purpose, we also present the estimated CLR by the CLR estimation algorithm of Li [25].

The CLR estimation algorithm of Li is briefly described here for completeness. In the algorithm of Li, three virtual buffers are used. Whenever n ATM cells arrive, linear least square regressions are performed separately on the CLR samples of the virtual buffers based on Eq. 2 or Eq. 5. n is the number of cells to be observed to get one CLR sample of the virtual buffer. Both regression results are stored. R^2 tests are performed to choose the appropriate equations for regression. Two running counters C_M and C_L are kept in the algorithm. If the R^2 result associated with the Markovian model is greater than that associated with the self-similar model, C_M is incremented by one, otherwise C_L is incremented by one. The above procedure is performed until N monitoring periods have finished. If C_M is greater than C_L, the Markovian regression model is chosen, otherwise the self-similar regression model is chosen. They then extrapolate to get the N CLR estimates of the real buffer based on the chosen regression model.

Fig. 6 shows that significant improvements are achieved by our algorithm. The improvements are in two respects:

- First, our algorithm requires much less monitoring period. In our algorithm only one valid CLR observation of the virtual buffers is sufficient in order to generate a CLR estimate. In the algorithm by Li, a number of CLR observations of the virtual buffers have to be made in order to generate a CLR estimate of the real buffer.
- Second, our algorithm is more accurate than the algorithm by Li. This is

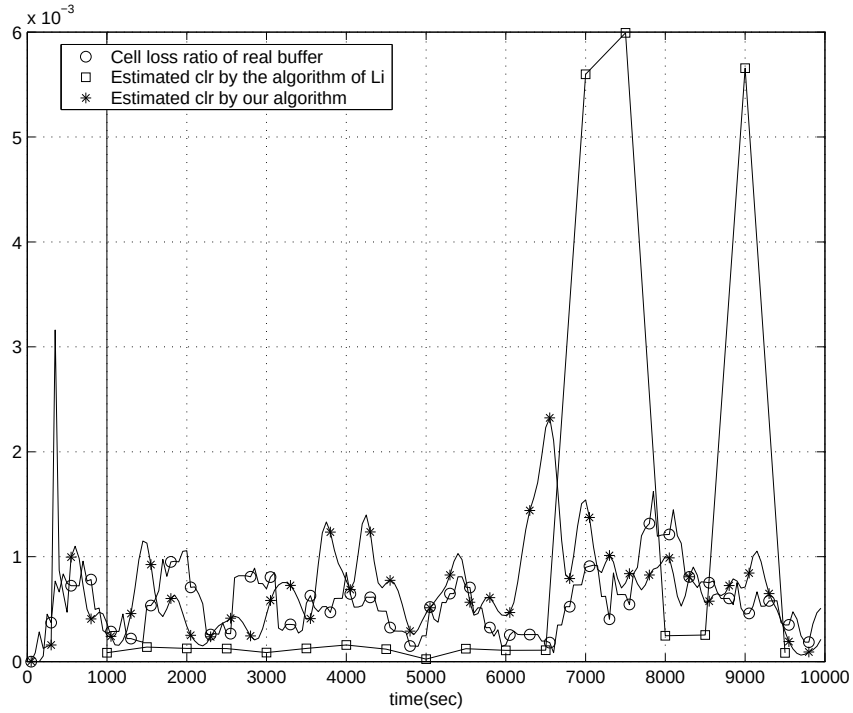

Fig. 6. CLR estimation in the Markovian scenario with a buffer size of 300 cells

clearly shown in the figure.

Estimation errors do exist. In addition to the estimation error due to the algorithm itself, part of the error shown in the figure comes from CLR observation. The estimated CLR of our algorithm in a monitoring period T gives an estimate of the CLR of the real buffer in the same interval. However, it is difficult to make a valid observation of CLR of the real buffer in the small monitoring period. Therefore the CLR of the real buffer shown in the figure is the ratio of lost cells to the total cells offered to the link for transmission in a much larger time interval. This mismatch becomes an error source in Fig. 6 as well.

Fig. 7 shows the estimated CLR as well as the CLR of the real buffer for different buffer sizes in the Markovian scenario. Since the Markovian scenario is a stationary scenario in the sense that there is a fixed number of connections on the link, for ease of comparison, averages of the CLR are shown in the figure. For example, the CLR of the real buffer shown in the figure for a buffer size of 400 cells is the average value of CLR observations of the real buffer in the simulation where buffer size is set to be 400 cells. Accordingly the estimated CLR shown in the figure is also the average value of estimated cell loss ratios.

Fig. 7 shows that the estimation error of our CLR estimation algorithm increases with buffer size. When buffer size is below 300 cells, the estimation algorithm is able to give an accurate estimate of CLR in the real buffer. When the buffer size is increased to 700 cells, the estimation algorithm can only give

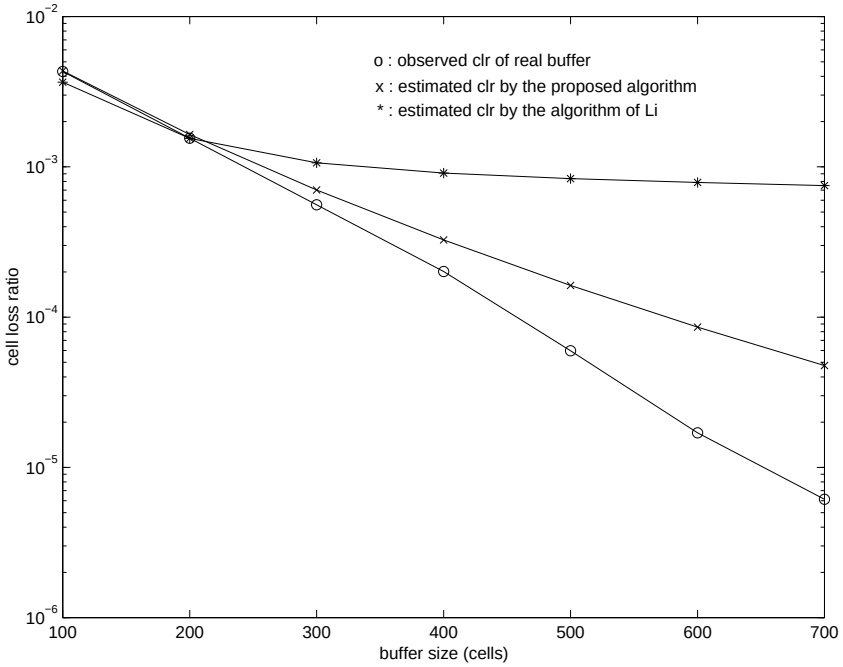

Fig. 7. Cell loss ratio estimation in the Markovian scenario

a CLR estimate that is accurate within one order of magnitude of the actual value. Fig. 7 also shows that the accuracy of our algorithm is much better than that proposed by Li. In our algorithm, instead of using the theoretical asymptotic relationship for Markovian and self-similar traffic directly, some amendments are made by introducing the parameters α and β so that the asymptotic relationship is captured more accurately. As an example, Fig. 8 shows the value of the two parameters in the simulation where buffer size is 300 cells. It can be seen that the parameters α and β are neither close to 0 nor close to 1, but fluctuate around 0.5. This implies that these two parameters do contribute to making our CLR estimation algorithm more accurate than that of Li.

5.2 Self-Similar Scenario

In the self-similar scenario, MPEG encoded movie "Starwars" is used as vbr video traffic source [26], [27]. This video trace is well-known to be self-similar. Five vbr video traffic sources are used in this scenario. At the beginning of the simulation, each traffic source starts to read from a random position in the vbr video file and generates traffic accordingly. When the end of the video file is reached, the traffic source continues generating traffic from the beginning of the file. Ten simulations are run in the self-similar scenario with buffer size varying from 100 cells to 1000 cells. Each simulation is run for $10,000$ seconds. As a typical example, Fig. 9 shows the results of the simulation with a buffer

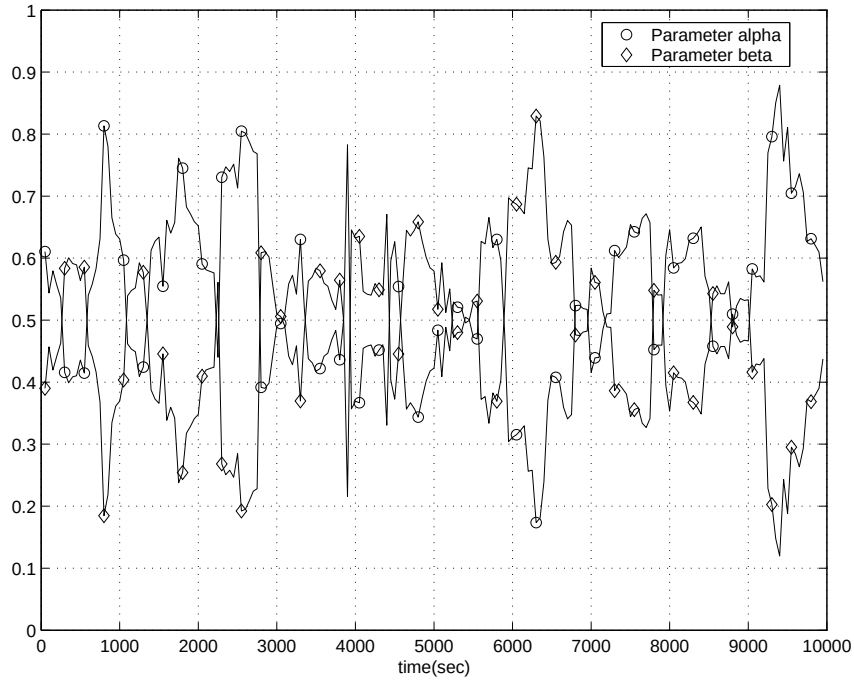

Fig. 8. Parameters α and β in the Markovian scenario with a buffer size of 300 cells

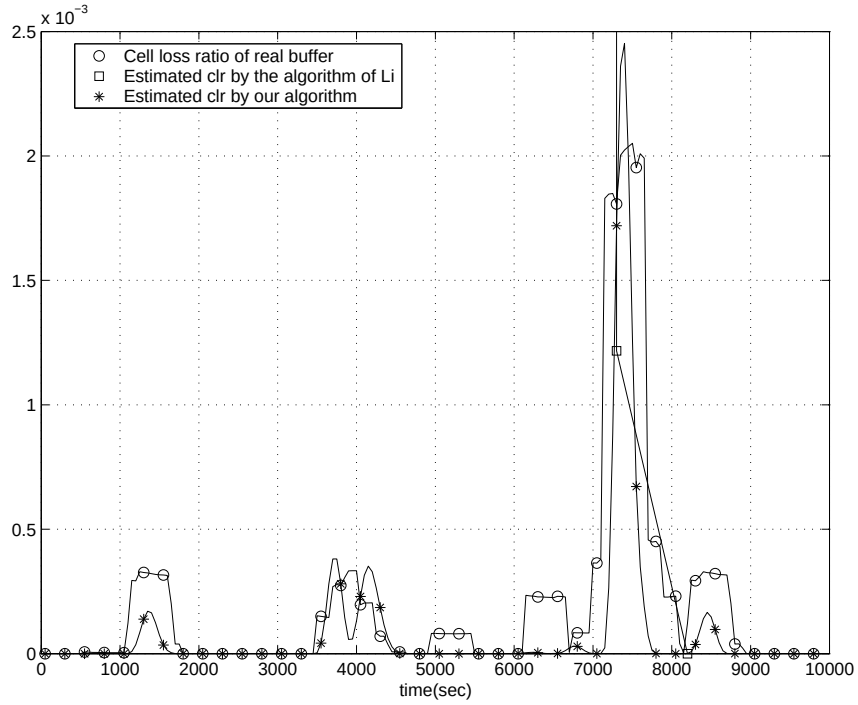

Fig. 9. CLR estimation in the Self-similar scenario with a buffer size of 500 cells

size of 500 cells. Fig. 10 shows the variations of parameter α and β in the simulation.

Fig. 9 shows that our estimation algorithm is able to estimate the cell loss ratio

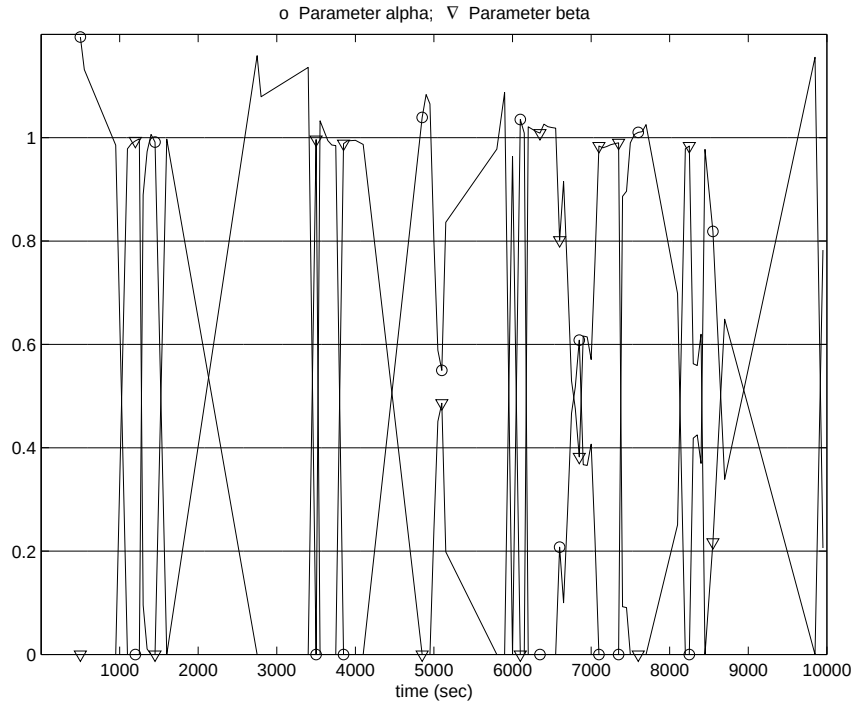

Fig. 10. Parameters α and β in the Self-similar scenario with a buffer size of 500 cells

of self-similar traffic accurately. In comparison, the algorithm of Li can only generate two CLR estimates during the $10,000$ seconds simulation time, i.e. a CLR estimate of 1.217×10^{-3} at $7300s$ and a CLR estimate of 1.942×10^{-17} at $8200s$.

Fig. 11 shows the estimated CLR as well as the CLR of the real buffer for different buffer sizes in the Self-similar scenario. Since the Self-similar scenario is a stationary scenario in the sense that there is a fixed number of connections on the link, for ease of comparison, only averages of the CLR are shown in the figure.

Simulation results show that the proposed algorithm generally gives a CLR estimate that is accurate within one order of magnitude around the true value. This estimation error must be taken into account when the proposed CLR estimation algorithm is applied to real applications.

6 Conclusion and Further Research

In this paper, we designed a real-time loss performance monitoring and estimation scheme. The proposed scheme is based on the asymptotic relationship between the cell loss ratio and the buffer size for both Markovian and self-

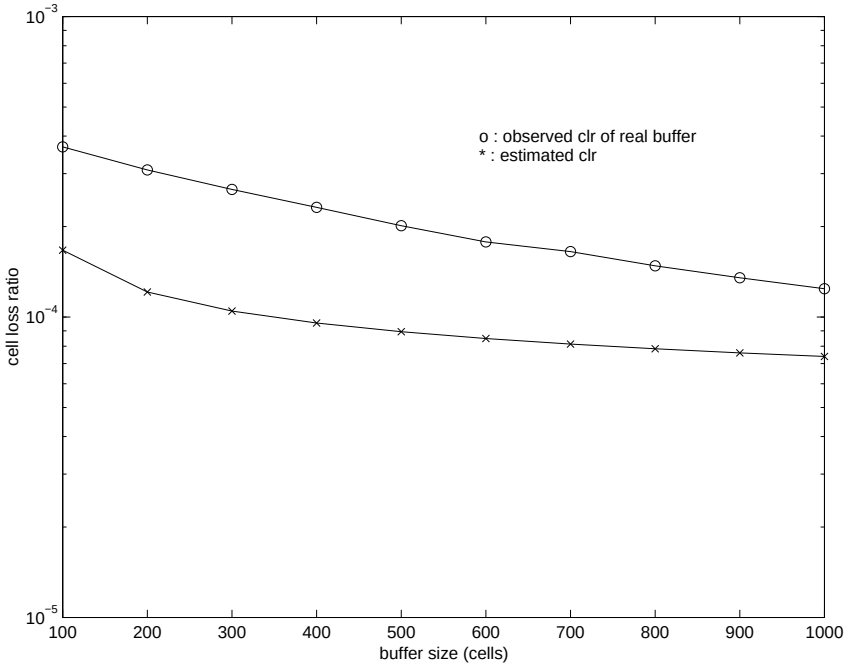

Fig. 11. Cell loss ratio estimation in the Self-similar scenario

similar traffic.

Simulation using both theoretical Markovian traffic model and real vbr video source was performed to investigate the performance of the proposed scheme. Simulation results indicate that the proposed scheme requires much less monitoring period and achieves better accuracy than that proposed in the literature. The proposed scheme is therefore suitable for real applications.

Currently, this research is performed in the framework of ATM networks. Motivated by the ubiquitous implementation of IP network, we shall extend the applications of the scheme to IP networks. The major challenge is that IP network uses variable-sized packets and ATM network uses 53-byte fixed-size packet, called ATM cells. As a result, the application of the virtual buffer techniques to IP network is more complicated. It will possibly need a major revision of the aforementioned theoretical framework. In addition to traffic loss, the proposed scheme shall also be extended to incorporate the monitoring of delay and delay variations, which can be possibly achieved by extrapolating the traffic loss - buffer size relationship obtained from virtual buffers for different buffer sizes.

References

[1] A. Asgari, P. Trimintzios, M. Irons, G. Pavlou, R. Egan, S. V. d. Berghe, A scalable real-time monitoring system for supporting traffic engineering, in: IEEE

Workshop on IP Operations and Management 2002, 2002, pp. 202–207.

[2] J. Garcia-Hernandez, M. Ghanbari, In-service monitoring of quality of service in atm networks using oam cells, IEE proceedings-Communications 146 (2) (1998) 102–106.

[3] M. Siler, J. Walrand, Monitoring quality of service: measurement and estimation, in: 37th IEEE Conference on Decision and Control, 1998.

[4] H. B. Zhu, V. S. Frost, In-service monitoring for cell loss quality of service violations in atm networks, IEEE/ACM Transactions on Networking 4 (2) (1996) 240–248.

[5] D. Anick, D. Mitra, M. M. Sondhi, Stochastic theory of data-handling systems with multiple sources, Bell System Technical Journal 61 (8) (1982) 1871–1893.

[6] A. I. Elwalid, D. Mitra, Effective bandwidth of general markovian traffic sources and admission control of high speed networks, IEEE/ACM Transactions on Networking 1 (1993) 329–343.

[7] A. Elwalid, D. Heyman, T. V. Lakshman, D. Mitra, Fundamental bounds and approximations for atm multiplexers with applications to video teleconferencing, IEEE Journal on Selected Areas in Communications 13 (6) (1995) 1004–1016.

[8] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, R. Weber, Admission control and routing in atm networks using inferences from measured buffer occupancy, IEEE Transactions on Communications 43 (2/3/4) (1995) 1778–1784.

[9] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, On the self-similar nature of ethernet traffic (extended version), IEEE/ACM Transactions on Networking 2 (1) (1994) 1–15.

[10] J. Beran, R. Sherman, M. S. Taqqu, W. Willinger, Long-range dependence in variable-bit-rate video traffic, IEEE Transactions on Communications 43 (2/3/4) (1995) 1566–1579.

[11] M. E. Crovella, A. Bestavros, Self-similarity in world wide web traffic: Evidence and possible causes, IEEE/ACM Transactions on Networking 5 (6) (1997) 835–846.

[12] M. S. Taqqu, W. Willinger, R. Sherman, Proof of a fundamental result in self-similar traffic modeling, Computer Communication Review 27 (1997) 5–23.

[13] D. Heath, S. Resnick, G. Samorodnitsky, Heavy tails and long range dependence in on/off processes and associated fluid models, Mathematics of Operations Research 23 (1) (1998) 145–165.

[14] N. Likhanov, B. Tsybakov, N. Georganas, Analysis of an atm buffer with self-similar ("fractal") input traffic, in: IEEE INFOCOM 1995, Vol. 3, Boston, MA , USA, 1995, pp. 985–992.

[15] M. Parulekar, A. M. Makowski, Tail probabilities for a multiplexer with self-similar traffic, in: IEEE INFOCOM 1996, San Francisco, CA , USA, 1996, pp. 1452–1459.

[16] D. R. Cox, Long-range dependence: A review, in: H. A. David, H. T. David (Eds.), Statistics: An Appraisal, The Iowa State University, Ames, Iowa, 1984, pp. 55–74.

[17] I. Norros, A storage model with self-similar input, Queueing Systems 16 (1994) 387–396.

[18] W. J. Krzanowski, F. H. C. Marriott, Multivariate analysis, Library of Congress Cataloguing in Publication Data, 1994.

[19] J. Y. Hui, Resource allocation for broadband networks, IEEE Journal on Selected Areas in Communications 6 (9) (1988) 1598–1608.

[20] J. W. Roberts, Variable-bit-rate traffic control in b-isdn, IEEE Communications Magazine 29 (9) (1991) 50–56.

[21] C. L. Hwang, S. Q. Li, On input state space reduction and buffer noneffective region, in: IEEE INFOCOM 1994, Toronto, Ont., Canada, 1994, pp. 1018–1028, tY - CONF.

[22] K. Shiomoto, S. Chaki, N. Yamanaka, A simple bandwidth management strategy based on measurements of instantaneous virtual pacth utilization in atm networks, IEEE/ACM Transactions on Networking 6 (5) (1998) 625–634.

[23] G. L. Choudhury, D. M. Lucantoni, W. Whitt, Squeezing the most out of atm, IEEE Transactions On Communications 44 (2) (1996) 203–217.

[24] B. Bensaou, S. T. C. Lam, H. W. Chu, D. H. K. Tsang, Estimation of the cell loss ratio in atm networks with a fuzzy system and application to measurement-based call admission control, IEEE/ACM Transactions on Networking 5 (4) (1997) 572–584.

[25] J.-S. Li, Measurement and in-service monitoring for qos violations and spare capacity estimations in atm network, Computer Communications 23 (2000) 162–170.

[26] M. W. Garrett, Contributions towards real-time services on packet switched networks, Phd thesis, Columbia University (1993).

[27] A. Wong, C.-T. Chen, D. J. L. Gall, F. C. Heng, K. M. Uz, Mcpic: A video coding algorithm for transmission and storage applications, IEEE Communications Magazine 28 (11) (1990) 24–32.