Design of new real-time models for tight upper bound approximation of cell loss ratio in ATM networks

A.T. Haghighat, K. Faez*, S. Khorsandi

Department of Electrical Engineering, Amirkabir University of Technology, Tehran 15914, Iran

Received 18 December 2001; revised 11 July 2002; accepted 29 September 2002

Abstract

ATM as a high-speed cell switching technology can support multiple classes of traffic with different quality of service (QoS) requirements and diverse traffic characteristics. A main QoS requirement is the cell loss ratio (CLR). We need a real-time expression for the CLR calculation in ATM networks where the statistical multiplexing is an important factor. The existing analytical methods for the CLR estimation are mostly based on fluid-flow and stationary approximate models. In this paper, we first evaluate these methods against the results obtained through simulation. The simulation is done at the cell level that provides very accurate results with buffer size as a variant. It is shown that the CLR estimation based on existing analytical models are widely overestimated. We have, then, proposed three new approaches that yield significant improvement in the accuracy of the CLR approximation. First, we have found global correction coefficients to compensate for the error of the current analytical methods. Second, we have proposed a new upper bound based on exact modeling of system behavior in the finite buffer case. This is a novel approach that combines fluid-flow and stationary approximate models and outperforms all the previous ones. The accuracy of the proposed model is verified by simulation. Third, we have found a tight piece-wise linear approximation that can be calculated in real-time. We have studied application of these bounds in non-homogeneous as well as homogeneous cases.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: ATM; QoS; CLR approximation; Tight upper bound; Real-time; Piece-wise linear approximation

1. Introduction

Asynchronous transfer mode (ATM) is a cell switching technology that can handle multiple types of traffic with different quality of service (QoS) requirements and diverse traffic characteristics. To facilitate the coexistence of multiple traffic classes, virtual path (VP) subnetworks within the ATM network have been proposed. Therefore, a VP is a single logical direct link between two nodes that can be shared by many virtual circuits (VCs) with similar bandwidth characteristics and QoS requirements. The VP concept simplifies traffic control and resource management. As a consequence, processing requirement for call establishment decreases and routing becomes more flexible. Statistical multiplexing of VCs enables efficient use of transmission capacity for bursty sources.

We are interested in finding accurate practical expressions for the cell loss ratio in VP-based ATM networks. This expression will be used in the call admission control (CAC) mechanism and routing algorithms of ATM networks. The CAC scheme determines whether a call can be routed on a VP without violating the guaranteed QoS requirements of the existing calls. The essential issue of ATM CAC is exact estimation of network performance through real-time calculations. Finite-buffer fluid-flow model [1] and the equivalent capacity (effective bandwidth) concept proposed by Guerin, et al. [2] are the foundations of many CAC algorithms [7–11]. The simplifying assumption of ($\beta = 1$) used in Ref. [2] results in ignoring the effect of statistical multiplexing. Therefore, the equivalent capacity and the cell loss probability expressions, which are obtained using this assumption, are not accurate.

In this work, we first study four expressions for approximating CLR based on stationary and fluid-flow models. We have built an accurate numerical (simulation) model for a finite buffer system at the cell level. We use the results of the simulations for evaluation of the analytical models and it is shown that the existing models are overly loose. It is also shown that these models are complementary
and their combination through a minimum operator provides a tighter upper bound for the CLR. We have, then, proposed three new approaches that yield significant improvements in the accuracy of CLR approximation. First, we have found global correction coefficients to compensate for the error of the analytical methods while preserve their upper bound property. Second, we have proposed a new upper bound based on exact modeling of system behavior in the finite buffer case. This is a novel approach that combines fluid-flow and stationary approximate models and outperforms all the previous ones. The accuracy of the proposed model is verified by simulation. Third, we have found a tight piece-wise linear approximation that can be calculated in real-time. We have studied application of these bounds in non-homogeneous as well as homogeneous cases.

The remainder of this paper is organized as follows: Section 2 gives an introduction to the traffic model of sources. The equivalent capacity and existing analytical cell loss probability expressions are discussed in Section 3. In Section 4, we propose an accurate numerical model for finding the cell loss probability for the finite buffer case. Also, the simulation results and evaluations of the existing models for CLR estimation are given in this section. In Section 5, we propose two new models for CLR approximation in ATM networks. In Section 6, we propose a real-time upper bound for CLR in ATM networks based on a piece-wise linear approximation. Section 7 gives guidelines on the application of the proposed piece-wise linear CLR estimation in the non-homogeneous networks. Section 8 contains the main conclusions and related discussions.

2. Traffic model

A single traffic source has a variable bit rate bounded by the maximum bit rate of its physical attachment. In order to characterize the effective bit rate or equivalent capacity of a connection, we need to select an appropriate model to specify its characteristics in terms of known parameters or metrics. In this paper, we consider a two-state On–Off fluid-flow model [12], which captures the basic behavior of the data source associated with a connection. Each source independently and asynchronously alternates between the On and Off states. Such a source in an On period transmits at the peak rate and in an Off period does not generate any traffic. The duration of the On and Off periods are exponentially distributed for each source. Moreover, the sources are mutually independent. Without loss of generality, the time unit is selected to be the average On period; with this unit of time, the average Off period is denoted by $1/\delta$ ($\delta$ is equal to the average On period divided by the average Off period).

The On–Off Markov model is simple and flexible, as it can be used for modeling traffic streams ranging from burst to continuous bit. This model can be used for VBR as well as CBR sources [14]. This model has also been successfully used to characterize the On–Off nature of an individual source or source element, like packetized voice and video [15].

In the literature, many studies have been reported on the characterization of ATM statistical multiplexers using two-state On–Off model ([1–6] and [13–17]). These studies are based on exponential-type tail probabilities for the cell loss distribution. There are also traffic models based on the long-range dependence, or self-similarity, which is shown to be applicable to video traffic and LAN traffic. However, for most traffic streams, and especially for superposition of several traffic sources, models with exponential-type tails work well for a wide range of buffer sizes of interest (e.g. real-time services) [13].

3. Equivalent capacity and cell loss probability

The equivalent capacity of a set of VCs statistically multiplexed on a VP is defined as the amount of bit rate required to achieve a desired QoS requirement, such as the cell loss probability $P_{\text{loss}}$. The cell loss probability is equal to the probability of buffer overflow. It is a function of the traffic characteristics of sources and the available network resources such as buffers.

Guerin et al. [2] proposed two approximate models, one of which is based on the fluid-flow approximation and the other one relies on the stationary approximation. In the stationary approximation, the distribution of the aggregate stationary bit rate is approximated by a binomial distribution in the case of identical two-state Markov sources and also can be approximated by a Gaussian distribution in general (e.g. heterogeneous sources). The first approximate model, i.e. fluid-flow approximate model, accurately estimates the equivalent capacity if the impact of individual connection characteristics is critical. The two approaches of the second approximate model (Stationary models) are good representatives of bandwidth requirements when the effect of statistical multiplexing is significant [2]. However, because both approximations are conservative and are inaccurate for different ranges of connection characteristics (which will be shown by simulation results), these models complement each other. In the following, we review these three analytical models (1- fluid-flow approximate model, 2- stationary approximation using binomial distribution, 3- stationary approximation using Gaussian distribution). The general characteristics of these analytical models are shown in Table 1. Also, we find CLR expressions for each of these models to be used in ATM CAC and routing mechanisms.

3.1. Fluid-flow approximation

The fluid-flow model for two-state Markov sources is proposed in Refs. [1,2]. In this model, the bit rate generated by a number of statistically multiplexed VCs is represented as a continuous flow of bits with varying intensity according
to the state of an underlying continuous-time Markov chain. We first consider the case of a single two-state Markov source described by a triplet $(r, \rho, b)$, where $r$ is the peak rate, $\rho$ is the fraction of time the source is active and $b$ is the mean of the $On$ period. Other parameters of interest, such as the mean $m$ and the variance $\sigma^2$ of the bit rate are identified completely from the source metric vector $(r, \rho, b)$. In this case, the distribution of the buffer contents can be derived using standard techniques for either infinite or finite buffer systems. In the case of finite buffer size $x$, the capacity required, $c'$, so that the CLR is limited to $e$ is defined as the equivalent capacity and is found from the following equation [2]:

$$e = \beta \exp \left( -\frac{x(c' - r \rho)}{b(1 - \rho)(r - c')c'} \right)$$

(1)

Where

$$\beta = \frac{(c' - r \rho) + \epsilon r (r - c')}{(1 - \rho)c'}$$

The notations used in the text are listed in Table 2.

In Section 7, we will show that an infinite buffer system satisfies the same equation, only with different value of $\beta$. From Eq. (1), it can be seen that, even for a single VC, there is no explicit expression for the equivalent capacity, and Eq. (1) must be solved numerically. However, $\beta$ is typically close to (in fact, always smaller than) 1 and approximating $\beta$ by 1 provides explicit expressions for $c'$ and $\epsilon$, which are slightly greater than the exact values.

In the case of multiple heterogeneous superposed sources, the approach is more complex than a single source. In the special case of $n$ multiplexed two-state Markov sources, the VP equivalent capacity is of the form [2]:

$$c = \sum_{i=1}^{n} c'_i$$

(2)

Let $c$ be the VP capacity and $l - 1$ is the number of VCs present in the VP. Our objective is to determine the admissibility of the $l$th call without violating the target cell loss probability ($e$). Let $f$ be the ratio of the VP capacity $c$ to the VC peak rate $r(f = c/r)$. In a homogeneous environment, through Eq. (2), we have $c' = c/l$. In other words, the VP capacity is divided up equally among the $l$ identical VCs. Therefore, with simplifying assumption $\beta = 1$, and after some manipulation (see more details in Ref. [3]), the cell loss probability, $P_{1_{\text{loss}}}(l)$, is found as follows:

$$P_{1_{\text{loss}}}(l) = \begin{cases} \exp \left( -\frac{x}{r} (1 + \delta - \frac{16 \delta}{f}) (1 - \frac{f}{7}) \right) & \text{if } (l > f) \\ 0 & \text{if } (l \leq f) \end{cases}$$

(3)

Nevertheless, the simplifying assumption $\beta = 1$ results in ignoring the effect of statistical multiplexing on the cell loss probability. Therefore, a modification is needed to accurately estimate the $P_{\text{loss}}$ for cases in which statistical multiplexing is significant. Although we cannot find an explicit expression for the equivalent capacity from Eq. (1), but we can find an expression for cell loss ratio, $P_{2_{\text{loss}}}(l)$, without assuming $\beta = 1$, as follows:

$$P_{2_{\text{loss}}}(l) = \frac{\beta_1 P_{1_{\text{loss}}}(l) - \beta_2 \gamma P_{1_{\text{loss}}}(l)}{1 - \beta_2 \gamma}$$

(4)

### Table 1
General characteristics of the analytical models

<table>
<thead>
<tr>
<th>Model</th>
<th>Base of model</th>
<th>Heterogeneous/Homogeneous</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluid-flow approximation</td>
<td>Continuous-time Markov chain</td>
<td>Homogeneous sources (Two-state Markov sources)</td>
<td>This model is accurate when the impact of individual connection characteristics is critical</td>
</tr>
<tr>
<td>Stationary approximation</td>
<td>The distribution of the aggregate stationary bit rate</td>
<td>Homogeneous sources (Identical two-state Markov sources)</td>
<td>This model is accurate when the effect of statistical multiplexing is significant</td>
</tr>
<tr>
<td>using binomial distribution</td>
<td></td>
<td>Heterogeneous sources (General case)</td>
<td>This model is accurate when the effect of statistical multiplexing is significant</td>
</tr>
<tr>
<td>Stationary approximation</td>
<td>The distribution of the aggregate stationary bit rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>using Gaussian distribution</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2
List of the notation used in the text

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/\delta$</td>
<td>Average $Off$ period (average $On$ period is 1)</td>
</tr>
<tr>
<td>$r$</td>
<td>Peak rate of the source (VC)</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Fraction of time the source is active</td>
</tr>
<tr>
<td>$m$</td>
<td>Mean aggregate bit rate</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Standard deviation of the aggregate bit rate</td>
</tr>
<tr>
<td>$1_0$</td>
<td>Desired CLR</td>
</tr>
<tr>
<td>$x$</td>
<td>Buffer capacity</td>
</tr>
<tr>
<td>$l$</td>
<td>Number of sources (VCs) present in the link (VP)</td>
</tr>
<tr>
<td>$c$</td>
<td>Link (VP) capacity</td>
</tr>
<tr>
<td>$c'$</td>
<td>Equivalent capacity of the source (VC)</td>
</tr>
<tr>
<td>$f$</td>
<td>Ratio of the link capacity to the source peak rate($f = c/r$)</td>
</tr>
</tbody>
</table>
The stationary approximation then gives:

\[ \beta_1 = 1 + \delta \left( 1 - \frac{k}{\tau} \right) \]  
and  

\[ \beta_2 = \delta \left( \frac{k}{\tau} - 1 \right) \]

### 3.2. Stationary approximation

In the following, we introduce two equations for \( P_{\text{loss}} \) based on the stationary approximation proposed in Ref. [2].

#### 3.2.1. Stationary approximation using binomial distribution

In the special case of \( n \) identical two-state Markov sources, we can consider the stationary bit rate distribution as a binomial distribution. Let \( c \) be the VP capacity and \( \varepsilon \) the aggregate bit rate generated by \( n \) sources and \( \varepsilon \) the desired cell loss probability, we have:

\[ \Pr(z > c) \leq \varepsilon \]

This means that, the frequency of overload periods must be less than \( \varepsilon \). In the case of \( n \) identical two-state Markov sources, the probability \( P_k \), that \( k \) out of \( n \) sources are active is given by a binomial distribution [2].

\[ P_k = \binom{n}{k} \rho^k (1 - \rho)^{n-k} \]

The value of \( c \), i.e. the smallest VP capacity needed to satisfy the desired cell loss probability, is then obtained by finding the smallest integer \( k' \) such that:

\[ \sum_{k=k'+1}^{n} p_k \leq \varepsilon \]

The stationary approximation then gives:

\[ c = k' \tau \]

Where \( \tau \) is the peak rate of each source. We need to find \( P_{3_{\text{loss}}} (l) \), which is the cell loss probability computed from stationary approximation using binomial distribution. From Eq. (5), we obtain:

\[ P_{3_{\text{loss}}} (l) = \sum_{k=l+1}^{\infty} P_k, \]

\[ f = k' = \left[ \frac{c}{\tau} \right], \quad P_k = \binom{l}{k} \rho^k (1 - \rho)^{l-k} \]

#### 3.2.2. Stationary approximation using Gaussian distribution

In a general case (e.g. non-homogeneous sources), the computation of \( c \) is more complex than the special case discussed earlier. However, in most cases when the effect of statistical multiplexing is of significance, the distribution of the stationary bit rate can be rather accurately approximated by a Gaussian distribution [2]. A good approximation is given by:

\[ c = m + \alpha' \sigma, \quad \alpha' = \sqrt{-2 \ln(\varepsilon) - \ln(2 \pi)} \]

When \( m \) is the mean aggregate bit rate and \( \sigma \) is the standard deviation of the aggregate bit rate, we have:

\[ \sigma^2 = \sum_{i=1}^{n} \sigma_i^2, \quad m = \sum_{i=1}^{n} m_i \]

Now, we should obtain \( P_{4_{\text{loss}}} (l) \), which is the cell loss probability based on stationary approximation using Gaussian distribution. From Eq. (7), and after some manipulation (see more details in Ref. [3]), we have:

\[ P_{4_{\text{loss}}} (l) = \exp \left( \frac{f^2}{2} \right) \left( - \frac{f - ml}{2 \sigma^2} \right)^2 - \frac{0.5 \ln(2 \pi)}{2} \]

### 4. Numerical study and simulation results

In this section, an accurate numerical model for obtaining the cell loss ratio through simulation will be introduced and the existing analytical models will be evaluated by the simulation results. Just like Section 3.2.1, here again, we consider a finite buffer with the capacity of \( x \) (Mbits), the FIFO queuing, and two-state Markov (On–Off) arrival traffic, such that \( On \) and \( Off \) periods have the exponential distribution, with respective means of \( \frac{1}{s} \) (s) and \( \frac{1}{\delta}(s) \). The source bit rate is zero during the \( Off \) periods and \( r \) (Mbps) in the \( On \) periods. The VP capacity is \( c \) (Mbps) and we have a finite buffer of size \( x \), receiving traffic from \( l \) \( On–Off \) sources and is discharged at the constant rate of \( c \). The objective is to find the buffer overflow probability (the loss probability). A discrete event simulation is built in C++ to obtain the \( P_{\text{loss}} \) for different values of \( l, x, \delta \), and \( f \).

- The simulation is done at the cell level and the results of the simulations are accurate with a confidence interval of \( P_{\text{loss}} \pm 10^{-10} \) and confidence level of 99.9%.
- We compare the results of the simulation with the results of the fluid-flow approximation, stationary approximation using Gaussian distribution and stationary approximation using binomial distribution methods.
- We study the applicability of the analytical methods for the \( P_{\text{loss}} \) approximation as a function of \( l \) and \( \delta \). These results lead to a new expression, based on the combination of the three analytical methods.
- The result of the simulation will help us to determine the minimum, the maximum and the average error of each of the analytical methods. These results lead us to the error correction factors applied on analytical results to compensate for their errors.

In the following, we will first explain the event processing method, which is used in this numerical model, and then we will discuss the details of our model.
4.1. Event processing model

We model the finite-buffer case as a discrete event system. Such a system is described by its state variables. A transition in the state variables occurs only as a result of an event. A transition in the state of a source from On to Off or vice versa is described by an event. Each traffic source is represented by a chain of events triggered by an initial random seed event. The events are processed in chronological order.

4.2. Event generation

The system maintains a Future Event List (FEL). For each source, upon completion of an event, the next event in the chain is scheduled and stored in the FEL. In the presence of l two-state Markov source, the FEL contains l events at each time. Each event can be of the two On-to-Off and Off-to-On types, which depends on the current state of the source. The event inter-arrival times are generated using an array of exponential random number generators. Table 3, explains the system parameters used in the simulation. Fig. 1, shows the details of the model used in this work.

4.3. Numerical evaluation of existing models

In this section, we present the results of the simulation. We compare \(P_{1\text{loss}}, P_{2\text{loss}}, P_{3\text{loss}}\) and \(P_{4\text{loss}}\) expressions (3), (4), (6), and (8) against the simulation results. The system parameters are assumed to be \(c = 150 \text{ Mbps}, r = 3 \text{ Mbps}, f = 50\), and \(x = 24 \text{ Mbits}\). We select these values like the values, which have been used in numerical experiments in Ref. [11]. In Section 6, we explain that these values are in the practical ranges, concerning the ATM technology.

Figs. 2–4 compare the \(P_{1\text{loss}}, P_{2\text{loss}}, P_{3\text{loss}}\) and \(P_{4\text{loss}}\) expressions with the \(PN_{\text{loss}}\) obtained from simulation for

Table 3

<table>
<thead>
<tr>
<th>Variable/Event</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>An array with (l) elements used for storing the occurrence time of the system events</td>
</tr>
<tr>
<td>State</td>
<td>An array with (l) elements used for storing the state(On or Off) of each of the (l) sources</td>
</tr>
<tr>
<td>Count</td>
<td>Counter of the number of cycles in the main loop of simulation</td>
</tr>
<tr>
<td>(j)</td>
<td>The number of the source which its state changes earlier than others (the number of the current event)</td>
</tr>
<tr>
<td>Previous-event</td>
<td>The occurrence time of the previous event</td>
</tr>
<tr>
<td>No-On</td>
<td>The number of sources which presently are in the On state</td>
</tr>
<tr>
<td>Time-bet-event</td>
<td>The time interval between the occurrence of the current event and the previous event</td>
</tr>
<tr>
<td>Extra-rate</td>
<td>The rate at which the buffer is filled (or emptied) in the time interval between the two recent events</td>
</tr>
<tr>
<td>Extra-bits</td>
<td>The number of bits (Mbits) increased to (or decreased from) the buffer in the time interval between the two recent events</td>
</tr>
<tr>
<td>Tot-bits</td>
<td>The counter of the total bits (Mbits) generated by the (l) sources</td>
</tr>
<tr>
<td>Loss</td>
<td>The counter of the lost bits (overflowed from the buffer)</td>
</tr>
<tr>
<td>(P_{\text{loss}})</td>
<td>Shows the buffer fullness</td>
</tr>
<tr>
<td>Bits</td>
<td>The number of bits (Mbits) generated in the time interval between the two recent events</td>
</tr>
</tbody>
</table>

Fig. 1. Numerical model of finite buffer with On_Off Markov sources.
different values of $\delta$, 0.125, 1.0, and 5.0, respectively. Note that in Fig. 4, there is a singularity at $l = 60$ for $P_{2\text{loss}}$, because at this point $\beta_1$ is equal to zero. CLR can be calculated at this point using linear interpolation:

$$P_{2\text{loss}}(l) = \frac{P_{2\text{loss}}(l - 1) + P_{2\text{loss}}(l + 1)}{2} \tag{9}$$

Figs. 2–4 demonstrate that the $P_{4\text{loss}}$ is completely different from $P_{1\text{loss}}, P_{2\text{loss}}, P_{3\text{loss}}$ and $P_{N\text{loss}}$. Therefore, the stationary model using Gaussian distribution is not a good approximation for CLR in the case of two-state Markov sources. This conclusion was not unpredictable, because this model is a general approximation, but the other two analytical models are specialized for two-state Markov sources.

Also, it is shown that $P_{1\text{loss}}, P_{2\text{loss}}, P_{3\text{loss}}$, and $P_{4\text{loss}}$ are all overly loose at least in some practical range of the CLR. We have repeated the study for a wide range of $\delta$ and $x$ and similar results have been found.

5. New accurate CLR approximation techniques

The results of the simulation show that the existing analytical models are not accurate. In this section two new techniques are proposed. We have shown that they yield significant improvement in CLR approximation.

First, we use the simulation results to find correction coefficients to compensate for the error of the analytical methods. We find these correction coefficients, so that the CLR expressions remain upper bound. Therefore, we combine these improved expressions through a Min operator.

Second, we propose a new more accurate analytical model for CLR approximation. This model is based on a modified version of the fluid-flow approximate model. In the fluid-flow approximate model, the buffer is considered infinite and the CLR is assumed to be equal to the probability of the buffer content being greater than a fixed threshold ($x$). In the improved model, we propose a more accurate definition for CLR. This new CLR definition leads to a new model based on combination of fluid-flow and stationary approximate models. The simulation results verify that this model outperforms all the previous models.

5.1. Error compensation of the existing models

Let us define $\alpha_i$ to be equal to the ratio of actual CLR (estimated through simulation) to that obtained from the analytical model $P_{i\text{loss}}$. We denote $\alpha_i$ as the error correction coefficient. Table 4 shows the minimum, the maximum and the average of the error correction coefficient for the three analytical models in the practical range. Let us define $\alpha_1, \alpha_2$ and $\alpha_3$ as the maximum values of the error correction coefficients for the three analytical models, respectively. We maintain that new
CLR estimations through the following still provide valid upper bounds for CLR:

\[
P_{1\text{loss}} = \alpha_1 \times P_{1\text{loss}} \text{ (old)}
\]
\[
P_{3\text{loss}} = \alpha_3 \times P_{3\text{loss}} \text{ (old)}
\]
\[
P_{4\text{loss}} = \alpha_4 \times P_{4\text{loss}} \text{ (old)}
\]

Where \( \alpha_1 = 1.04 \times 10^{-2}, \alpha_3 = 9.55 \times 10^{-3}, \) and \( \alpha_4 = 7.37 \times 10^{-3}. \) Since \( \alpha_1 < 1, \) new upper bounds are much more accurate than old ones.

Our simulation study in Section 4 indicated that the inaccuracy of the analytical models vary for different ranges of connection parameters. Since all of these models are valid upper bounds, the minimum of these values will give us a more accurate estimation in all ranges:

\[
P_{\text{loss}}(l) = \min\{P_{1\text{loss}}(l), P_{3\text{loss}}(l), P_{4\text{loss}}(l)\}
\]

\[
\delta = 0.125 \\
\delta = 1.0 \\
\delta = 5.0
\]

\[
\begin{array}{c|c|c|c}
\hline
 & P_{1\text{loss}}/P_{1\text{loss}} & P_{3\text{loss}}/P_{3\text{loss}} & P_{4\text{loss}}/P_{4\text{loss}} \\
\hline
\delta = 0.125 & 4.48 \times 10^{-3} & 9.55 \times 10^{-3} & 2.41 \times 10^{-3} \\
\delta = 1.0 & 2.51 \times 10^{-4} & 4.15 \times 10^{-3} & 1.89 \times 10^{-3} \\
\delta = 5.0 & 3.31 \times 10^{-3} & 1.11 \times 10^{-3} & 5.29 \times 10^{-4} \\
\hline
\end{array}
\]

Table 4: The maximum, minimum and average of the error correction coefficients

5.2. A new accurate model for CLR approximation

In this section we propose a new more accurate analytical model for CLR approximation. This approach is based on a modified version of the fluid-flow approximation designed for accurate CLR estimation with a finite buffer.

In the infinite-buffer fluid-flow model with \( n \) two-state Markov sources, we have the following:

\[
F_i(x) = \text{equilibrium probability that } i \text{ sources are On and buffer content does not exceed } x
\]

The following expression represents the model in a matrix notation:

\[
D \frac{d}{dx} F(x) = MF(x), \quad x \geq 0,
\]

\[
D = \text{diag}\{-f, 1-f, 2-f, \ldots, n-f\},
\]

and \( G(x) = \text{Pr} \text{(Buffer content} x) = F(x), \quad x \geq 0. \)

where \( I \) denotes the unity vector and prime denotes transposition. \( G(x) \) is referred to as the 'Probability of buffer overflow beyond x', and is obtained as follows [1]:

\[
G(x) = - \sum_{j=0}^{n-[f]} e^{x} a_i (I^f \Phi_i) \quad (11)
\]

Where \( \{z_i\} \) are the eigenvalues of \( D^{-1} M \) and \( \{\Phi_i\} \) are the associated right eigenvectors and \( \{a_i\} \) are the coefficients, which are obtained from the following expression:

\[
a_j = - \left( \frac{\delta}{\delta + 1} \right) \prod_{i=0}^{n-[f]} \frac{z_i}{z_i - z_j}, \quad 0 \leq j \leq n-[f] - 1
\]

Since the form of \( G(x) \) expression is a sum of exponential terms and the computation of this expression is complex for call admission and routing mechanisms of high speed networks, for simplicity of computation, \( G(x) \) is
approximated by the term with the largest negative exponent:

\[ G(x) \equiv -a_0(t' \Phi_0)e^{\alpha x} \]  

(12)

Where

\[ t' \Phi_0 = \left( \frac{n}{T} \right)^n \]

And \( z_0 \) is the largest negative eigenvalue of \( D^{-1} M \) and is obtained from the following expression:

\[ z_0 = -\frac{1 + \delta - \frac{n^2}{T}}{1 - \frac{z}{n}} \]

\( G(x) \) is used in buffer sizing, CLR approximation and CAC. A more accurate approximation of \( G(x) \) results in a more efficient resource dimensioning and smaller resource wastage due to approximation errors. This is the main motivation of this work.

We first address the case of a single two-state Markov source. Assuming \( n = 1 \) in the above model, we find the following expression, which is the same as Eq. (1), only with a different value for \( \beta \) (Note that the unit of time is selected to be the average On period (\( b = 1 \) in this model):

\[ P_{\text{loss}} = \beta \exp\left( -\frac{x(c' - rp)}{(1 - \rho)(r - c')c} \right) \]  

(13)

Where:

\[ \beta = \frac{rp}{c} \]

In the special case of \( n \) multiplexed two-state Markov sources, based on Eq. (2), we have \( c' = c/l \) and hence the following expression for the CLR is obtained.

The CLR expression (14) is not accurate (see Figs. 6–8), mainly because of the following reason. In this model, the buffer is considered infinite and the CLR is assumed to be equal to “the probability of the buffer content being greater than the threshold of \( x \)”. A more accurate assumption is “the probability that the buffer content exceeds \( x \) and the aggregate rate of the On sources exceeds \( c \).” In other words, the percentage of the times in which the content of an infinite buffer exceeds the threshold \( x \) and the aggregate arrival rate is less than the service rate (the buffer being emptied) must be eliminated from the cell loss probability. In a finite buffer case, if the aggregate arrival rate is less than the service rate, cells are not lost. In an infinite buffer case, if the

buffer content is greater than \( x \) and the aggregate arrival rate is less than the service rate, the buffer content will decrease toward \( x \) and although the buffer content is still more than \( x \), but in practice cells are not lost.

We, therefore, propose the following scheme for the CLR estimation:

\[ P_{\text{loss}} = \Pr(\{\text{aggregate rate of the On sources} > c\} \cap \{\text{Buffer content} > x\}) \]

In the extreme case of a bufferless system, the CLR is reduced to the probability that the aggregate rate of the On sources exceeds \( c \). Based on the independence assumption, we have:

\[ P_{\text{loss}} \approx \Pr(\text{Number of On sources} > f) \cdot \Pr(\text{Buffer content} > x) \]

The first term represents the stationary approximation using binomial distribution and the second term is the buffer overflow probability obtained from the fluid-flow approximation. Therefore, our new expression for the CLR is obtained as follows:

\[ P_{\text{loss}} = P_{1,\text{loss}}^{1} \cdot P_{2,\text{loss}}^{1} \]  

(15)

In the case of the finite buffer, the same rationale applies and an accurate CLR estimation is obtained from the following.

\[ P_{\text{loss}}^{1} = P_{2,\text{loss}}^{1} \cdot P_{3,\text{loss}}^{1} \]  

(16)

Figs. 6–8 compare seven CLR expressions (\( P_{1,\text{loss}}^{1} \ldots P_{7,\text{loss}}^{1} \)) with the \( PN_{\text{loss}} \) (the simulation results) for \( \delta \) equal to 0.125, 1.0, and 5.0, respectively. These figures show that for different traffic, from nearly burst traffic to nearly constant bit rate, \( P_{7,\text{loss}}^{1} \) is the most accurate upper bound approximation for CLR. The execution time is of the order of a few microseconds using a typical commercial processor. We can write expression (16) in the following form:

\[ P_{\text{loss}}^{1} = P_{2,\text{loss}}^{1} \sum_{k=1}^{l-f+1} \left( \prod_{i=1}^{l-k} \frac{k+i}{i} \right) \rho^{l-i} \]  

(14)

There are similar ideas but different approaches in Refs. [14,18]. In Ref. [14], Thuy and Ha have approximated
the overall cell loss ratio to be equal to the probability of cell loss without the buffer multiplies by the probability of the queue exceeding the buffer size $x$, or the overflow probability:

$$CLR = CLR_{ub} \cdot \text{Pr}(\text{Buffer content } x)$$

In their paper, the bufferless cell loss ratio has been approximated by the following integral:

$$CLR_{ub} = \frac{E[l(t) - C]}{E[l(t)]} = \int_c^\infty \frac{z - c}{m\sigma\sqrt{2\pi}} \exp\left(\frac{-(z-c)^2}{2\sigma^2}\right) dz$$

In this model, the aggregate rate process $l(t)$ with set of parameters $(m, \sigma^2)$ has been approximated as a Gaussian process with mean $m$ and variance $\sigma^2$. Finally, after solving the above integral, the following expression has been found ($\Gamma$ denoting the Gamma function):

$$CLR_{ub} = \frac{1}{2\sqrt{\pi}}(c - m)\left(\frac{\Gamma(m + \sigma^2)}{\Gamma(m)}\right)$$

Also, in Ref. [18], Yan and Beshai have considered the cell loss ratio as follows:

$$\ln(P_{loss}) = \ln(\beta) + \ln(\eta)$$

Where $\beta$ is the probability of joining the buffer, $\nu = (llf) \times \rho$ is the link mean cell occupancy, and $b$ is the blocking probability in a bufferless loss system with $f = clr$ servers and $l$ sources ($E$ denoting the Erlang function):

$$b = \left(1 - \frac{f}{l}\right)E(l\rho, f)$$
$$\beta = \frac{b}{1 - \nu + \nu b}$$

Fig. 6. Comparison of the $P_{N_{\text{loss}}}$ with $P_{1_{\text{loss}}}$ to $P_{7_{\text{loss}}}$ when $\delta = 0.125$, $f = 50$, and $x = 24$.  

Fig. 7. Comparison of the $P_{N_{\text{loss}}}$ with $P_{1_{\text{loss}}} - P_{7_{\text{loss}}}$ when $\delta = 1.0$, $f = 50$, and $x = 24$.  

```
Also $\eta$ is the conditional cell overflow probability as:

$$\ln(\eta) = \frac{x}{r} \left( 1 - \frac{c}{lr - c} \right)$$

Figs. 6–8 demonstrate that $P_7$ is more accurate than both of the above CLR expressions, which have been proposed in Refs. [14,18].

6. Piece-wise linear approximation

The analytical models discussed so far suffer from computational complexity and they may not be directly usable in the ATM routing algorithm or call admission function. We need an explicit and real-time expression, which can approximate $P_{\text{loss}}$ as a function of different parameters such as $l, d, x$, and $f$. In this section, we develop an accurate computationally efficient upper bound expression for real-time calculation of CLR that can be used in very fast CAC mechanisms. CAC in ATM networks can be used based on two different methods: a direct performance evaluation method or an inverse resource-requirement-estimation method [18]. In the direct method, the estimated CLR resulting from the admission of a new call is calculated. In the inverse method, the equivalent capacity of the new arrival is determined. The call is accepted if the remaining unassigned capacity of the route is not less than the calculated equivalent capacity. The second method is suitable for heterogeneous sources as well as the homogeneous ones. From Eqs. (16) and (17), it can be seen that there is no explicit expression for the equivalent capacity, and these expressions must be solved numerically. Therefore, we need a CLR expression that gives us an explicit expression for equivalent capacity as well.

Usually the desired cell loss probability is considered in the range of $10^{-6}$ to $10^{-9}$ [2]. We have, however, considered the range of $10^{-3}$ to $10^{-9}$ for $P_{\text{loss}}$ to cover a wide range of parameters. Our objective is to find a numerically efficient expression for $P_{\text{loss}}$ in this range with high accuracy.

Fig. 9 shows $P_{\text{loss}}$ (the simulation results) for different $\delta$ as a function of $l$. It can be seen that the logarithm of $P_{\text{loss}}$ in the range of $10^{-3}$ to $10^{-9}$ is a linear function of $l$. We, therefore, use a piece-wise linear approximation to obtain an expression for the $P_{\text{loss}}$. The slope of $\ln(P_{\text{loss}})$ is a function of $f, x, r, l$. Let us define $\psi$ as follows:

$$\ln(P_{\text{loss}}) = \psi(l, f, x, \delta)$$

Since $\rho = \delta/(\delta + 1)$, we can write:

$$\ln(P_{\text{loss}}) = \psi(l, f, x, \rho)$$

and

$$\partial \psi(l, f, x, \rho)/\partial l = f(f, x, \rho)$$

Considering that $\psi$ is a linear function of $l$, Table 5 shows the end points of 27 lines, which estimate $\psi$ for $x$ of 24, 48, and 96, $f$ of 25, 50, and 100, and $\delta$ of 0.125, 1, and 5. Our linear approximate model heavily depends on the ranges of the parameters ($f, x, \text{and} \delta$) used in simulation. Therefore, it is very important that the values of the parameters cover the whole practical ranges concerning the ATM technology. Usually the desired cell loss probability is considered in the range of $10^{-6}$ to $10^{-9}$ [2]. However, we have considered the range of $10^{-3}$ to $10^{-9}$ for $P_{\text{loss}}$. Moreover, a wide range of traffic patterns from nearly burst ($\delta = 0.125$) to nearly constant bit rate ($\delta = 5.0$) is considered. Additionally, we have considered the range of $x$ (the buffer capacity) up to 96 Mbits, because a bigger buffer size leads to a higher delay. For
instance, in a high-speed 1 Gbps network, a 100 Mbits switch buffer size creates a 100 ms switch delay. Due to the fact that in many real-time multimedia traffics, such as audio—video conferencing, the maximum delay must be 200 ms [19], the packet delay in each switch must be smaller than 100 ms. Consequently, the buffer size in network switches must be smaller than 100 Mbits. However, we have considered the ranges of the parameters, so that they cover the whole practical ranges used in the numerical examples in ATM technology literature (e.g. [2,7–11]). These ranges are considered wide enough to achieve a general expression, which is valid in all ranges of $f_x, x, l,$ and $\delta$. We need to find a line, in which the end points ($\{(x_1,y_1)\}$ and $(x_2,y_2)$) are the functions of $f_x$, and $\rho$ and these parametric end points must be fitted to all of 27 end points $(\{(x_1,y_1)\}$ and $(x_2,y_2))$ listed in Table 5.

Our first attempt is to find a linear approximation in the range of $10^{-3}$ to $10^{-9}$ for the CLR, which is sufficient for all practical purposes. In all of these cases, $5 \times 10^{-7}$ is an upper bound for $y_1$ and $6 \times 10^{-3}$ is an upper bound for $y_2$. We can, therefore, write:

$$y_1 = \ln(5 \times 10^{-7}) = -14.5$$

$y_2 = \ln(6 \times 10^{-3}) = -5.11$

Due to the fact that the above fixed values are considered for $y_1$ and $y_2$, only $x_1$ and $x_2$ must be determined as functions of the network and traffic parameters. In Table 5, we have several sample points of two multiple parameter functions ($x_1(f_x, x, \rho)$ and $x_2(f_x, x, \rho)$). Now, we want to fit two curves to these sample points so that the Mean Square Error (MSE) becomes minimum. We have considered that these functions can be estimated as expression (18). It should be noted that, based on existing analytical models, there is not any second order (and higher order) term of the parameters in this expression.

$$x_1(f_x, x, \rho) = k_1 f_x + k_2 x + k_3 \rho + k_4 f_x x + \cdots + k_n f_x^n$$

$\vdots$ $+ k_n f_x^n \rho$  

(18)

We have proposed a new extended version of the Least Mean Square Error (LMSE) method, to be used for estimating the coefficients of a full free multiple parameter function by minimizing the Mean (or sum) Square Error for all the sample points. This algorithm is shown in Fig. 10.

We ran this algorithm for the sample points of Table 5 and the following expressions were found. It should be noted that some of the coefficients are zero (or nearly zero).

$$x_1 = 0.53 \frac{f_x}{\rho} + 0.5f_x + \frac{5}{\delta} \left( \frac{f_x}{25} - 1 \right) + \frac{2.5}{\delta} \left( \frac{x}{24} - 1 \right)$$

$$x_2 = 0.8 \frac{f_x}{\rho} + 0.24f_x + \frac{2}{\delta} \left( \frac{f_x}{25} - 1 \right) + \frac{2.5}{\delta} \left( \frac{x}{24} - 1 \right)$$

The linear estimation can be written as:

$$\ln(P_{\text{Lin,loss}}) = \frac{y_2(x_1 - l) + y_1(l - x_2)}{x_1 - x_2}$$

By substituting for $x_1, x_2, y_1$ and $y_2$ in the above expression, we can write:

$$P_{\text{Lin,loss}} = \exp \left( \frac{179.5 - 60.2 f_x - 6.53 l - 62.6 l \delta - 65.44 \delta f_x}{20 + f_x + 0.066 f_x \delta} \right)$$

(19)

Figs. 11–13 demonstrate that the expression (19) is an accurate approximation of CLR in the practical range of operation. It should be noted that $PN_{\text{loss}}$ is considered the actual value of the CLR.

The desired practical range of CLR in ATM networks is $10^{-6}$ to $10^{-3}$ and therefore the expression (19) can be used for calculating $P_{\text{loss}}$ in the routing algorithms and CAC mechanism of ATM networks. However, for the purpose of generality, we develop a piece-wise linear

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f$</th>
<th>$\delta$</th>
<th>$\rho$</th>
<th>$x$</th>
<th>$y$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>25</td>
<td>0.125</td>
<td>1/9</td>
<td>130</td>
<td>$1.15 \times 10^{-7}$</td>
<td>186</td>
<td>$1.62 \times 10^{-3}$</td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>29</td>
<td>$1.92 \times 10^{-3}$</td>
<td>30</td>
<td>$3.05 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0.125</td>
<td>1/9</td>
<td>300</td>
<td>$1.24 \times 10^{-7}$</td>
<td>390</td>
<td>$1.56 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>82</td>
<td>$1.40 \times 10^{-7}$</td>
<td>94</td>
<td>$1.34 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>58</td>
<td>$4.00 \times 10^{-8}$</td>
<td>60</td>
<td>$2.94 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.125</td>
<td>1/9</td>
<td>641</td>
<td>$9.01 \times 10^{-8}$</td>
<td>796</td>
<td>$2.18 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>172</td>
<td>$1.38 \times 10^{-7}$</td>
<td>191</td>
<td>$1.34 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>116</td>
<td>$2.78 \times 10^{-8}$</td>
<td>120</td>
<td>$2.84 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>25</td>
<td>0.125</td>
<td>1/9</td>
<td>150</td>
<td>$5.95 \times 10^{-8}$</td>
<td>206</td>
<td>$2.29 \times 10^{-3}$</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>42</td>
<td>$9.90 \times 10^{-8}$</td>
<td>49</td>
<td>$1.96 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>29</td>
<td>$1.05 \times 10^{-8}$</td>
<td>31</td>
<td>$2.06 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0.125</td>
<td>1/9</td>
<td>320</td>
<td>$1.87 \times 10^{-7}$</td>
<td>405</td>
<td>$1.35 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>86</td>
<td>$1.29 \times 10^{-7}$</td>
<td>98</td>
<td>$1.29 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>58</td>
<td>$1.00 \times 10^{-10}$</td>
<td>60</td>
<td>$1.64 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.125</td>
<td>1/9</td>
<td>665</td>
<td>$1.15 \times 10^{-7}$</td>
<td>815</td>
<td>$1.23 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>174</td>
<td>$1.29 \times 10^{-7}$</td>
<td>194</td>
<td>$1.84 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>117</td>
<td>$5.1 \times 10^{-8}$</td>
<td>122</td>
<td>$1.11 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>25</td>
<td>0.125</td>
<td>1/9</td>
<td>192</td>
<td>$1.25 \times 10^{-7}$</td>
<td>248</td>
<td>$1.93 \times 10^{-3}$</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>47</td>
<td>$1.05 \times 10^{-7}$</td>
<td>54</td>
<td>$1.74 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>30</td>
<td>$9.67 \times 10^{-8}$</td>
<td>32</td>
<td>$3.08 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0.125</td>
<td>1/9</td>
<td>364</td>
<td>$1.83 \times 10^{-7}$</td>
<td>446</td>
<td>$2.69 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>91</td>
<td>$1.01 \times 10^{-7}$</td>
<td>102</td>
<td>$1.01 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>60</td>
<td>$7.5 \times 10^{-8}$</td>
<td>62</td>
<td>$1.07 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.125</td>
<td>1/9</td>
<td>705</td>
<td>$8.33 \times 10^{-8}$</td>
<td>854</td>
<td>$2.62 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>180</td>
<td>$9.99 \times 10^{-8}$</td>
<td>198</td>
<td>$2.21 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>118</td>
<td>$6.1 \times 10^{-8}$</td>
<td>123</td>
<td>$1.31 \times 10^{-3}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5
The end points of 27 lines, which estimate the CLR in a wide ranges of $x_f, \delta$. 
approximation for the whole range of CLR in a finite buffer system. For this, we have proposed a piece-wise model consisting of four line segments. Based on the results in Fig. 9, we have considered these line segments, as follows (see Table 6).

Fig. 9 shows that the logarithm of $P_{\text{loss}}$ in each of the above ranges is a linear function of $L$. Just like the above linear approximation method, which has estimated the $P_{\text{loss}}$ in the range of $10^{-9}$ to $10^{-3}$, here again we have found the following upper bound values based on the Table 6.

$$y_3 = \ln(4 \times 10^{-1}) = -0.91$$
Also, we have found the following expressions based on the Table 7, which are approximation of \( x_3, x_4, \) and \( x_5 \):

\[
x_3 = 1.108 \frac{f}{\rho}
\]

\[
y_3 = \ln(9 \times 10^{-1}) = -0.105
\]

\[
y_4 = \ln(5 \times 10^{-1}) = -0.69
\]

We can now find CLR expressions for line segments of 2, 3, and 4, based on the linear approximation. For example, for line segment of 2, we have:

\[
\ln(P_{\text{lin,loss}}) = \frac{y_3(x_2 - l) + y_2(l - x_2)}{x_2 - x_3}
\]

After replacing of \( x_2 \) and \( x_3 \) by their approximations and \( y_2 \) and \( y_3 \) by their values, we have:

\[
P_{\text{lin,loss}} = \exp\left( \frac{4.095 + 4.861f - 0.0948x - 4.21\delta + 4.716f\delta}{-4.5 - 0.228f + 0.104x - 0.068\delta} \right)
\]

Repeating this process for the third and forth segments, the following piece-wise linear approximation of the CLR is obtained:

\[
P_{\text{Piece,wise,loss}} = \begin{cases} 
\exp\left( \frac{179.5 + 60.2f - 6.53x + 62.6l\delta - 65.44\delta}{20 + f + 0.066f\delta} \right) & l < x_2 \\
\exp\left( \frac{4.095 + 4.861f - 0.0948x - 4.21\delta + 4.716f\delta}{-4.5 - 0.228f + 0.104x - 0.068\delta} \right) & x_2 < l < x_3 \\
\exp\left( \frac{0.246f}{f^2 - 1.183} \right) & x_3 < l < x_4 \\
\exp\left( \frac{0.073f}{f^2 - 0.836} \right) & x_4 < l
\end{cases}
\]

Where \( x_1 \) to \( x_5 \) are defined by the following expressions:

\[
x_1 = 0.53 \frac{f}{\rho} + 0.5f + \frac{5}{\delta} \left( \frac{f}{25} - 1 \right) + \frac{2.5}{\delta} \left( \frac{x}{24} - 1 \right)
\]

\[
x_2 = 0.8 \frac{f}{\rho} + 0.24f + \frac{2}{\delta} \left( \frac{f}{25} - 1 \right) + \frac{2.5}{\delta} \left( \frac{x}{24} - 1 \right)
\]

\[
x_3 = 1.108 \frac{f}{\rho}
\]

<table>
<thead>
<tr>
<th>Line segment</th>
<th>Ln of CLR in the first end point</th>
<th>Ln of CLR in the second end point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line segment of 1</td>
<td>( y_1 = \ln(5 \times 10^{-3}) )</td>
<td>( y_2 = \ln(6 \times 10^{-3}) )</td>
</tr>
<tr>
<td>Line segment of 2</td>
<td>( y_2 = \ln(6 \times 10^{-3}) )</td>
<td>( y_3 = \ln(4 \times 10^{-4}) )</td>
</tr>
<tr>
<td>Line segment of 3</td>
<td>( y_3 = \ln(4 \times 10^{-3}) )</td>
<td>( y_4 = \ln(5 \times 10^{-3}) )</td>
</tr>
<tr>
<td>Line segment of 4</td>
<td>( y_4 = \ln(5 \times 10^{-3}) )</td>
<td>( y_5 = \ln(9 \times 10^{-3}) )</td>
</tr>
</tbody>
</table>
Table 7

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f$</th>
<th>$\delta$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>25</td>
<td>0.125</td>
<td>1/9</td>
<td>249</td>
<td>450</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>56</td>
<td>100</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>34</td>
<td>60</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0.125</td>
<td>1/9</td>
<td>498</td>
<td>900</td>
<td>4500</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>111</td>
<td>200</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>67</td>
<td>120</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.125</td>
<td>1/9</td>
<td>996</td>
<td>1800</td>
<td>9000</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>222</td>
<td>400</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>133</td>
<td>240</td>
<td>1200</td>
<td></td>
</tr>
</tbody>
</table>

$48 = \frac{x}{f}$

$x_5 = 10 \frac{f}{\delta}$

**Fig. 14** illustrates the accuracy of the piece-wise linear approximation technique.

Fig. 14. Comparison of the $P_{\text{loss}}$ with $P_{\text{piece_wise_loss}}$, when $\delta = 1.0, f = 50$, and $x = 24$.

7. **Non-homogeneous traffic sources**

The equivalent capacity is a useful concept for analysis of non-homogeneous sources. The equivalent capacity of a set of VCs statistically multiplexed on a VP is defined as the amount of transmission capacity required to achieve the desired QoS requirements, such as the cell loss ratio. In this section, we show that the proposed piece-wise linear approximation is not limited to the homogeneous systems. By using the concept of equivalent capacity, we develop a new form of the proposed expression that can be used in a finite buffer with non-homogeneous sources.

In the special case of $n$ multiplexed two-state Markov sources, which are not necessarily identical, we define an equivalent capacity $c_i^*$ for the $i$th connection.

In order for a new call to be accepted, the sum of equivalent capacity of all connections routed through a VP must not exceed the VP capacity ($c$), that is $\sum_{i=1}^{n} c_i^* \leq c$.

The equivalent capacity $c_i^*$ is equal to $cl_i$, where $l_i$ is the maximum number of similar connections that can be routed through the VP without violating the desired CLR ($1$). In order to find $l_i$ as a function of other network and traffic parameters, we replace $P_{\text{loss}}$ by $1 - e$ (the desired CLR) in Eq. (19), which yields the following expression:

$$I = \frac{\ln(e)}{62.6 \delta} (20 + f + 0.066f \delta) - \frac{2.87}{\delta} + \frac{0.96 f}{\delta} + 0.104 \frac{x}{\delta} + 1.045f$$

The call admission mechanism in the non-homogeneous networks, therefore, will have the following steps:

- We calculate $l_i$ by using expression (22).
- We find $c_i^*$ by using expression $c_i^* = cl_i$.
- We can admit the $n$th call, if: $\sum_{i=1}^{n} c_i^* \leq c$

Note that we have found $\sum_{i=1}^{n-1} c_i^*$ in the admission phase of the previous connection.

8. **Conclusion**

In this paper, first we discussed three analytical approximate models for cell loss ratio in the finite buffer system. Second, we provided an accurate numerical model for simulation of a buffer with the buffer size as a variant. We used the simulation results to evaluate the analytical models and we showed that the existing analytical models are all overly loose at least in some practical range of the CLR. Then we proposed three new approaches to increase the accuracy of CLR approximation. First, we found global correction coefficients to compensate for the error of the analytical methods. Second, we proposed a new tight upper bound based on exact modeling of system behavior in the finite buffer case. We combined the fluid-flow and the stationary
approximate models and we showed that this novel approach outperforms all the previous ones. The simulation results verified the accuracy of the proposed model. Third, because of the calculation complexity of previous models, we found a tight piece-wise linear approximation that can be calculated in real-time. It was shown that the piece-wise linear approximate model is applicable in non-homogeneous as well as homogeneous cases.

References