A Real-Time Resource Allocation Scheme for Broadband Access

Krishnamurthy Nagarajan and G. Tong Zhou*

Abstract— **Access to broadband networks has to be designed intelligently to maximize the utilization of network resources while avoiding congestion. In order to provide service guarantees, the network access node first computes the required resources before admitting the connection. Resource allocation algorithms based on simplified assumptions are commonly employed to facilitate real-time implementation. The network traffic however exhibits complex behavior and resources allocated based on simplified assumptions may be excessive or fail to maintain the specified guarantees. In this paper, we describe a novel resource allocation algorithm that is based on a log-normal fractional Autoregressive Integrated Moving Average (fARIMA) model. We implemented the algorithm on a Texas Instruments TMS320C6701 DSP processor and evaluated its performance on MPEG traffic streams. The results show that although our algorithm is more sophisticated than existing ones (because it allocates resources more efficiently), real-time implementation is feasible.**

I. Introduction

Broadband networks have to share the available bandwidth among different classes of network traffic (e.g., voice, video and data). In order to improve the utilization level, network resources (bandwidth, buffer size etc.) have to be shared intelligently among the users based on the statistical characteristics of the traffic they offer. Multimedia applications, while requiring strict transport guarantees, can tolerate infrequent losses. In such cases, the network can further improve efficiency by providing statistical guarantees on the quality of service (QoS). Based on the nature of the QoS requested by the application and its traffic characteristics, the network access node decides whether to accept or reject the user connection. The mechanism which makes the accept/reject decision is called Connection Admission Control (CAC).

CAC runs as a software module on a switch-control processor. It is important that the hardware and processing overhead of the resource allocation algorithm be minimal. Ideally, one would like to have a simple closed-form analytical expression to compute the required resources. But in practice, the network traffic exhibits complex behavior. For example, an MPEG traffic stream can exhibit both short-term and long-term memory characteristics. Short-term memory (or short-range dependent) means that the autocovariance function (ACF) defined as $c_{2x}(\tau) = E\{x(n)x(n+\tau)\} - E^2\{x(n)\}$, decays faster than $|\tau|^{-1}$ as $\tau \to \infty$. On the other hand, long-term memory (or long-range dependent) implies that $c_{2x}(\tau) \approx K \cdot |\tau|^{-m}$ for τ large, with $0 < m < 1$ and $K \neq 0$. Resource allocation algorithms that are based on short-term memory assumptions fail to provide guarantees on the QoS for MPEG traffic data.

Recently, we proposed a novel parametric CAC algorithm [8] for MPEG video traffic based on the effective bandwidth theory [5], which provides guarantees on the loss probability. In this paper, we will evaluate the real-time performance of our CAC algorithm using the Texas Instruments TMS320C6701 floating-point DSP processor. The processor's dedicated multiplier-accumulator circuitry, multiple access and special memory addressing modes designed to speed up repetitive operations make it an ideal platform for CAC algorithm implementation.

The rest of the paper is organized as follows: Section II provides background materials on our MPEG traffic model and the effective bandwidth theory. Section III presents a resource allocation framework for log-normal fARIMA traffic sources. Section IV provides the real-time implementation details and results of performance evaluation using real MPEG traffic traces. Finally, conclusions are drawn in Section V.

II. Resource Allocation Framework

A. MPEG Video Traffic Model

An MPEG video stream consists of I, P and B frames. A deterministic ordering of the I, P and B frames is called a Group of Picture (GOP) which gives rise to a random process $x(n)$ (assumed stationary) studied in this paper. In [6], we showed that $x(n)$ is approximately log-normal; i.e., its marginal probability density function (PDF) is given by:

$$f_X(x) = \frac{1}{x\sigma_y\sqrt{2\pi}} \exp\left\{-\frac{(\ln x - \mu_y)^2}{2\sigma_y^2}\right\}, \qquad (1)$$

where the parameters μ_y and σ_y^2 represent respectively, the mean and variance of the Gaussian process $y(n) = \ln\{x(n)\}$. To capture both the short-term and long-term memory characteristics of $x(n)$, we model $x(n)$ as a log-normal fARIMA process; i.e., $y(n)$ is a Gaussian fARIMA process. An fARIMA process is a natural extension of the familiar ARMA process. In the z-domain, the relation between the fARIMA(p, d, q) process $y(n)$ and the driving noise $w(n)$ is as follows

$$\mathbf{A}(z)Y(z) = (1-z^{-1})^{-d}\,\mathbf{B}(z)W(z), \quad -0.5 < d < 0.5, \quad (2)$$

where $\mathbf{A}(\mathbf{z}) = 1 + a(1)z^{-1} + \cdots + a(p)z^{-p}$, $\mathbf{B}(\mathbf{z}) = 1 + b(1)z^{-1} + \cdots + b(q)z^{-q}$ and $w(n)$ is i.i.d. Gaussian. The presence of a fractional pole at $z = 1$ introduces

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA. E-mail: {krishna.nagarajan, gtz}@ece.gatech.edu

*Contacting author: Tel. (404)-894-2907; Fax (404)-894-8363.

This work was supported in part by National Science Foundation grant MIP 9703312.

long-memory. The AR and MA parameters $\{a_i\}_{i=1}^{p}$ and $\{b_j\}_{j=1}^{q}$ give rise to short memory characteristics in the process. Whittle's method ([1] and references therein) is the most reliable way to estimate the model parameters of the fARIMA process $y(n)$. Once the parameters d, $\mathbf{a} = [1,\ a(1), \cdots,\ a(p)]$ and $\mathbf{b} = [1,\ b(1), \cdots,\ b(q)]$ are found, the theoretical $c_{2y}(\tau)$ can be calculated as follows:

$$c_{2y}(\tau) = \left[\frac{(-1)^{\tau}\Gamma(1-2d)}{\Gamma(\tau-d+1)\Gamma(1-\tau-d)}\right] \star \left[\sum_{t} h(t)h(t+\tau)\right], \quad (3)$$

where $\star$ denotes convolution, $H(z) = B(z)/A(z)$ and $\Gamma(\cdot)$ is the familiar Gamma function.

Note that we do not directly model the MPEG data $x(n)$, but instead model the log transformed data $y(n)$. We then infer the autocovariance structure of $x(n)$ through that of $y(n)$. The relationship between the autocovariance function $c_{2x}(\tau)$ of $x(n)$ and that of $y(n)$ is [6]

$$c_{2x}(\tau) = \exp\{2\mu_y + \sigma_y^2\} \cdot (\exp\{c_{2y}(\tau)\} - 1)\,. \quad (4)$$

Therefore, by parametrically modeling the Gaussian fARIMA process $y(n)$, we can characterize completely the correlation structure of $x(n)$ using (3)-(4).

B. Effective Bandwidth Theory

The effective bandwidth theory attempts to provide a measure of bandwidth and buffer size, which adequately represents the trade-off between sources of different types, and takes into account their statistical characteristics and QoS requirements [2]. We only study one type of QoS constraint here, which is the loss probability.

When $x(n)$ is short-range dependent ($\sum_{\tau} c_{2x}(\tau) < \infty$) with mean μ, the required capacity or effective bandwidth C is obtained as [2]

$$C = \mu + \frac{\delta}{2}\sum_{\tau} c_{2x}(\tau), \quad \delta = \frac{-\ln(\epsilon)}{B}. \quad (5)$$

When $x(n)$ is white with variance σ^2, we have $C = \mu + (\delta/2)\sigma^2$ as a special case of (5). Therefore, for a given loss probability ϵ and buffer size B, one can first compute δ and then find the effective bandwidth C using (5). Alternatively, if the network can only afford a service rate C, then δ can be found and subsequent substitution of δ into (5) gives the required buffer size B.

Several recent studies have confirmed that Variable Bit Rate (VBR) video traffic (such as MPEG) exhibits both short-term *and* long-term memory characteristics [1], [6], which implies that $\sum_{\tau} c_{2x}(\tau) = \infty$ and hence (5) cannot be used. In [9], a fractional Gaussian Noise (fGN) model was used for network traffic data, which is capable of capturing only the long-term memory characteristics. In [7], we proposed a resource allocation framework for Gaussian sources exhibiting both short-term and long-term memory characteristics by modeling the traffic source as a Gaussian fARIMA process. In the next section, we present a CAC strategy that allocates network resources for log-normal data such as the MPEG video traffic.

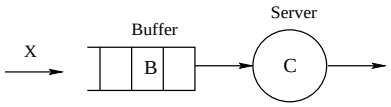

Fig. 1. A single-server queue.

III. Resource Allocation Scheme for Log-Normal fARIMA Processes

In general, if a user offers a bursty traffic $x(n)$ to a server having a buffer size B and capacity C (Figure 1), then a burst of size k starting when the buffer is empty, will cause overflow if $\{x(1) + \cdots + x(k)\} > k \cdot C + B$. Denote the sample mean of $\{x(n)\}_{n=1}^{k}$ by $\bar{X}_k = \frac{1}{k}\{x(1) + \cdots + x(k)\}$. According to the large deviation theory [4], if the user demands a loss probability no larger than ϵ, the capacity C and buffer size B should be such that

$$Pr\left[\bar{X}_k > C + \frac{B}{k}\right] \le \epsilon, \quad \forall \ \ k. \quad (6)$$

Rigorous calculation of the network resources B and/or C based on (6) requires the knowledge of the PDF of $\bar{X}_k$, the sample mean of a burst traffic of size k.

Unfortunately for $x(n)$ log-normal, the true PDF of the k-sample mean $\bar{X}_k$ does not have a closed form expression. In the context of communications theory, several authors have approximated the average of log-normal random variables by another log-normal random variable with appropriately chosen parameters. A simple and effective method that matches the first two moments of $\bar{X}_k$ was developed by Fenton [3]. Following Fenton's approach, we approximate $\bar{X}_k$ by a log-normal random variable $\tilde{X}_k$ for which $\ln \tilde{X}_k$ is Gaussian with mean $\tilde{\mu}_k$ and variance $\tilde{\sigma}_k^2$. The parameter values can be obtained by equating (matching) the mean and variance of $\bar{X}_k$ and $\tilde{X}_k$; i.e.,

$$\bar{\mu}_k = E\{\bar{X}_k\} = E\{\tilde{X}_k\} = \exp\left\{\tilde{\mu}_k + \frac{\tilde{\sigma}_k^2}{2}\right\}, \quad (7)$$

$$\bar{\sigma}_k^2 = \mathrm{Var}\{\bar{X}_k\} = \mathrm{Var}\{\tilde{X}_k\} = \exp\left\{2\tilde{\mu}_k + \tilde{\sigma}_k^2\right\}\left(\exp\{\tilde{\sigma}_k^2\} - 1\right). \quad (8)$$

Since $\bar{\mu}_k = E\{\bar{X}_k\} = E\{x(n)\} = \mu$, $\bar{\mu}_k$ can be estimated directly from the MPEG data $x(n)$. The variance of $\bar{X}_k$ can be shown to be

$$\bar{\sigma}_k^2 = \frac{1}{k}\sum_{|\tau|<k}\left(1 - \frac{|\tau|}{k}\right) c_{2x}(\tau). \quad (9)$$

Based on (7)-(8), we can express $\tilde{\mu}_k$ and $\tilde{\sigma}_k^2$ in terms of $\bar{\mu}_k$ and $\bar{\sigma}_k^2$ as follows:

$$\tilde{\sigma}_k^2 = \ln\left(\frac{\bar{\sigma}_k^2}{\bar{\mu}_k^2} + 1\right), \quad \tilde{\mu}_k = \ln(\bar{\mu}_k) - \frac{\tilde{\sigma}_k^2}{2}. \quad (10)$$

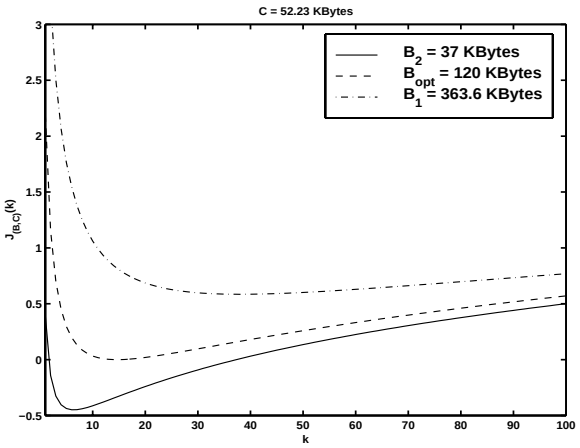

Fig. 2. Plot of $J_{(B,C)}(k)$ for different (B, C) pairs .

Now going back to our resource allocation framework (6), we replace $\bar{X}_k$ by $\tilde{X}_k$ and write

$$Pr\left\{\ln \tilde{X}_k > \ln(C + B/k)\right\} \leq \epsilon, \quad \forall\ k.$$

Since $\ln \tilde{X}_k$ is a Gaussian random variable with mean $\tilde{\mu}_k$ and variance $\tilde{\sigma}_k^2$, we can utilize a known upper bound on the Gaussian tail probability,

$$Pr\left\{\ln \tilde{X}_k > \ln(C + \frac{B}{k})\right\} \leq \frac{1}{2}\exp\left\{\frac{(\ln(C + \frac{B}{k}) - \tilde{\mu}_k)^2}{2\tilde{\sigma}_k^2}\right\}. \quad (11)$$

If we select C and B such that the r.h.s. of (11) satisfies

$$\frac{1}{2}\exp\left\{-\frac{(\ln(C + B/k) - \tilde{\mu}_k)^2}{2\tilde{\sigma}_k^2}\right\} \leq \epsilon, \quad \forall\ k, \quad (12)$$

then the QoS condition (6) is satisfied.

Taking the logarithm on both sides of equation (12) and rearranging, we obtain

$$J_{(B,C)}(k) \triangleq \frac{(\ln(C + B/k) - \tilde{\mu}_k)^2}{2\tilde{\sigma}_k^2} + \ln(2\epsilon) \geq 0, \quad \forall\ k. \quad (13)$$

Figure 2 shows a plot of $J_{(B,C)}(k)$ for different (B, C) pairs obtained for an MPEG video source. It is seen that for each (B, C) pair, $J_{(B,C)}(k)$ has a unique minimum at $k = k_o$. If $J_{(B,C)}(k_o) \geq 0$, then (13) is satisfied $\forall\ k$. A (B, C) pair is optimum if $\min_k J_{(B,C)}(k) = J_{(B,C)}(k_o) = 0$.

IV. Real-Time Implementation

Resource allocation algorithms typically runs as a software module in a network device such as a switch. It is desirable that the hardware and processing overhead required to implement the algorithm be minimal. In order to analyze the real-time performance of the proposed resource allocation algorithm, we selected the Texas Instruments (TI) TMS320C6701 floating point DSP processor. We operated the processor at a clock frequency of 133 MHz. The processor's on-chip program and data memory (64 K-Bytes each) were found to be sufficient for implementing our algorithm. The entire algorithm was implemented in the C-programming language. We compiled the program using the level-3 optimization provided by the TI C-compiler.

A. Implementation Details

Figure 3 shows the flowchart of the algorithm. The algorithm first obtains the QoS requirement and the traffic model parameters from the user. Then, it calculates the autocovariance function $c_{2x}(\tau)$ of the traffic based on (3). In our current implementation, we fixed the maximum lag value ($\tau_{\max}$) at 1024. The correlation values are stored in a double-precision floating-point array. An implication of using an upper limit on τ is that the algorithm can only compute those optimal $(B,\ C)$ pairs for which the burst size $k_o = \arg\min_k J_{(B,C)}(k)$ is less than $\tau_{\max}$.

Based on the QoS parameter specified by the user and existing traffic conditions in that class of service, the network fixes an operating load for the user traffic. The operating load is defined as the ratio between the mean traffic rate (μ) and the alloted bandwidth C. Once the load is fixed, the optimal buffer size B_{opt} needed to support the connection is computed. Recall that when $B = B_{opt}$, we have $\min_k J_{(B,C)}(k) = 0$. From equation (13), we observe that $J_{(B,C)}(k)$ is monotonically increasing with B. Therefore if $B > B_{opt}$, we have that $\min_k J_{(B,C)}(k) > 0$ whereas if $B < B_{opt}$, we have that $\min_k J_{(B,C)}(k) < 0$. This prompts us to employ an iterative search algorithm to find B_{opt}. We first pick B_1 and B_2 such that $\min_k J_{(B_1,C)}(k) > 0$ and $\min_k J_{(B_2,C)}(k) < 0$. We can be sure that B_{opt} lies between B_1 and B_2, i.e., $B_1 > B_{opt} > B_2$, and we call B_1 and B_2 the bracket points. Next, we would like to narrow down this bracket. At the i^{th} iteration, pick $B_1 > B_i > B_2$. If $\min_k J_{(B_i,C)}(k) > 0$, then we infer that $B_i > B_{opt} > B_2$ and we replace B_1 with B_i. On the other hand, if $\min_k J_{(B_i,C)}(k) < 0$, then we must have $B_1 > B_{opt} > B_i$ and we replace B_2 with B_i. By successively narrowing down the range, we bring the bracket points together and soon they converge to B_{opt}. Brent's method in particular provides superlinear convergence to the optimal solution [10].

In the process of searching for an optimal buffer size, the algorithm needs to repeatedly search for the burst size k_o that results in the minimum cost function. For a given buffer size B, the algorithm first obtains a three point bracket $(k_a,\ k_b,\ k_c)$ that captures the minimum [10, page 400]. Then, applying the "Golden Search" method, it identifies the burst size k_o within the interval $(k_a,\ k_c)$.

If the network can afford to allocate B_{opt} to the user, it goes ahead and accepts the user connection. If it does not have the required buffer size, the algorithm provides the user with an option to renegotiate its QoS requirement.

B. Performance Evaluation

We experimented with three real MPEG video traces "Dino", "Term", and "Mtv" which were obtained from [11]. Table I shows their fARIMA parameter estimates and traffic statistics based on 2,048 samples of GOP. Suppose these sources are to be admitted to the network with a desired loss probability $\epsilon = 10^{-5}$. Based on this QoS parameter, the CAC algorithm allocates buffer size B and bandwidth C according to the assumed traffic model (c.f. Section III).

The speed with which the algorithm converges depends on the following factors:

- Operating load (ρ) - higher values of ρ require a large buffer size to support the connection. The algorithm has to find the optimal buffer size from a larger search space thereby increasing the execution time.
- Bracketing strategy - the rate of convergence is affected by the number of times the algorithm evaluates different buffer sizes to obtain the bracket points.
- Strategy for obtaining B_{opt} within the bracket points.
- Strategy for obtaining the minimum of $J_{(B,C)}(k)$.

It must be noted that the number of computations required to evaluate $J_{(B,C)}(k)$ is proportional to the burst size k. This is because the algorithm has to first calculate the values of $\tilde{\sigma}_k$ and $\tilde{\mu}_k$ using (10). These parameters can be either computed *a priori* for all k or can be dynamically computed on an as needed basis. In our current implementation, we find that dynamic computation of $\tilde{\sigma}_k$ and $\tilde{\mu}_k$ is more efficient.

Tables II shows the computational resources required to calculate the optimal buffer sizes for MPEG traces under different operating loads. The index 'BufferCount' represents the number of different buffer sizes tried by the algorithm before converging to the optimal value. The index 'CostCount' represents the number of times the cost function $J_{(B,C)}(k)$ was evaluated. Together, the indexes 'BufferCount' and 'CostCount' provide a measure of the efficiency of the resource allocation algorithm and can be used to benchmark different bracketing and search strategies respectively. The table also presents the total CPU cycles required for the algorithm to converge and the corresponding execution time in seconds using the TMS320C6701 DSP processor operated at 133 MHz clock frequency.

From these results, we observe the following:

- As the operating load is decreased, the algorithm converges faster since the search space for an optimal buffer size is now smaller.
- For a given load, B_{opt} is proportional to the mean, standard deviation, long-term and short-term memory characteristics. For example, "MTV" has the largest mean, variance and long-term memory parameter d. For a load of 0.35, the network requires a buffer size of 3 M-Bytes to support a loss probability of 10^{-5} whereas for "Dino" and "Term" the required buffer size for the same operating load is considerably smaller.
- Due to delay constraints, the operating loads are restricted to the range of 0.2 to 0.4. In this range, it is seen that the proposed resource allocation algorithm typically converges within 300 milli-seconds if B_{opt} lies within 1 M-Bytes.

The execution time can be further reduced by (i) increasing the clock frequency (the new generation of TI TMS320C67X DSP processors operate at higher clock frequencies), (ii) identifying modules that can be implemented in parallel and (iii) developing better bracketing and search strategies. We are currently looking into the problem of extending our CAC framework to exploit the statistical multiplexing phenomena seen in ATM networks and study its effects on the algorithm execution times.

V. Conclusions

In our earlier work, we proposed a log-normal fARIMA model for MPEG traces which is more efficient than existing closed form resource allocation algorithms. The price paid is the increase in computation load. In this paper, we demonstrate that it is feasible to implement our more sophisticated algorithm in real-time using TI TMS320C6701 processor. At 133 MHz clock frequency, our algorithm converges within 300 milli-seconds. Therefore, high performance DSP processors make complex algorithms practical and the focus should be to devise efficient resource allocation algorithms that satisfy QoS requirements.

Acknowledgment: The authors would like to thank Dr. Thomas P. Barnwell at Georgia Tech for providing access to the TI TMS320C6701 based evaluation board and its associated development tools.

References

[1] J. Beran, R. Sherman, M. Taqqu, and W. Willinger, "Long-range dependence in variable bit rate video traffic," *IEEE Trans. Comm.*, Vol. 43, pp 1566-1579, 1995.

[2] C. Courcoubetis and R. Weber, "Effective bandwidths for stationary sources," *Prob. Eng. Inf. Sci.*, Vol. 9, pp 285-296, 1995.

[3] L. F. Fenton, "The sum of lognormal probability distributions in scatter transmission systems," *IRE Trans. Comm. Sys.*, pp 57-67, March, 1960.

[4] P. W. Glynn and W. Whitt, "Logarithmic asymptotics for steady-state tail probabilities in a single-server queue," *J. Appl. Prob.*, Vol. 31A, pp 131-159, 1994.

[5] F. Kelly, "Notes on effective bandwidth," *Stochastic Networks: Theory and Applications*, Oxford Univ. Press, 1996.

[6] K. Nagarajan and G. T. Zhou, "Modeling the short and long memories of VBR video streams," *Proc. Intl. Workshop on Applications of Heavy-Tail Distributions in Economics, Engineering and Statistics*, Washington, DC, June 1999.

[7] K. Nagarajan and G. T. Zhou, "A new resource allocation scheme for Gaussian traffic sources," *Proc. Intl. Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000 (to appear).

[8] K. Nagarajan and G. T. Zhou, "MPEG video traffic sources: modeling and resource allocation," *IEEE Signal Processing Letters* (submitted), May 2000.

[9] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE J. Selected Areas Comm.*, Vol. 13, No. 6, pp 953-962, 1995.

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge Univ. Press, 1992.

[11] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," *Tech. Rep. No. 101*, University of Wurzburg, 1995.

Trace	p	d	q	ARMA Parameters	Mean	Standard Deviation
Dino	1	$\hat{d} = 0.3462$	0	$\hat{\mathbf{a}} = [1, \ -0.5586]$	21	8
Term	0	$\hat{d} = 0.3886$	1	$\hat{\mathbf{b}} = [1, \ 0.2817]$	16.36	5.86
MTV	1	$\hat{d} = 0.4064$	0	$\hat{\mathbf{a}} = [1, \ -0.2090]$	37.6	18.13

TABLE I

fARIMA parameter estimates for three MPEG video traces. Mean and standard deviation are represented in K-Bytes.

(a) "Dino"

Load ($\rho = \mu/C$)	B_{opt} (K-Bytes)	BufferCount	CostCount	CPU Cycles	Execution Time (Seconds)
0.35	168	9	70	20689022	0.156
0.40	341	12	118	23670956	0.178
0.45	675	14	163	33573985	0.252
0.50	1361	15	211	69922970	0.526
0.55	2860	14	185	97560423	0.733

(b) "Term"

Load ($\rho = \mu/C$)	B_{opt} (K-Bytes)	BufferCount	CostCount	CPU Cycles	Execution Time (Seconds)
0.30	17	8	52	20007470	0.15
0.35	49	8	54	20160583	0.152
0.40	131	9	79	21438244	0.161
0.45	358	14	177	38211701	0.287
0.50	1021	15	211	91081901	0.684

(c) "MTV"

Load ($\rho = \mu/C$)	B_{opt} (K-Bytes)	BufferCount	CostCount	CPU Cycles	Execution Time (Seconds)
0.20	61	7	46	19923789	0.15
0.25	260	11	93	21075742	0.158
0.30	911	12	121	25182707	0.189
0.35	3076	15	206	71302621	0.536

TABLE II

Real-time implementation of the resource allocation algorithm for MPEG video traffic traces

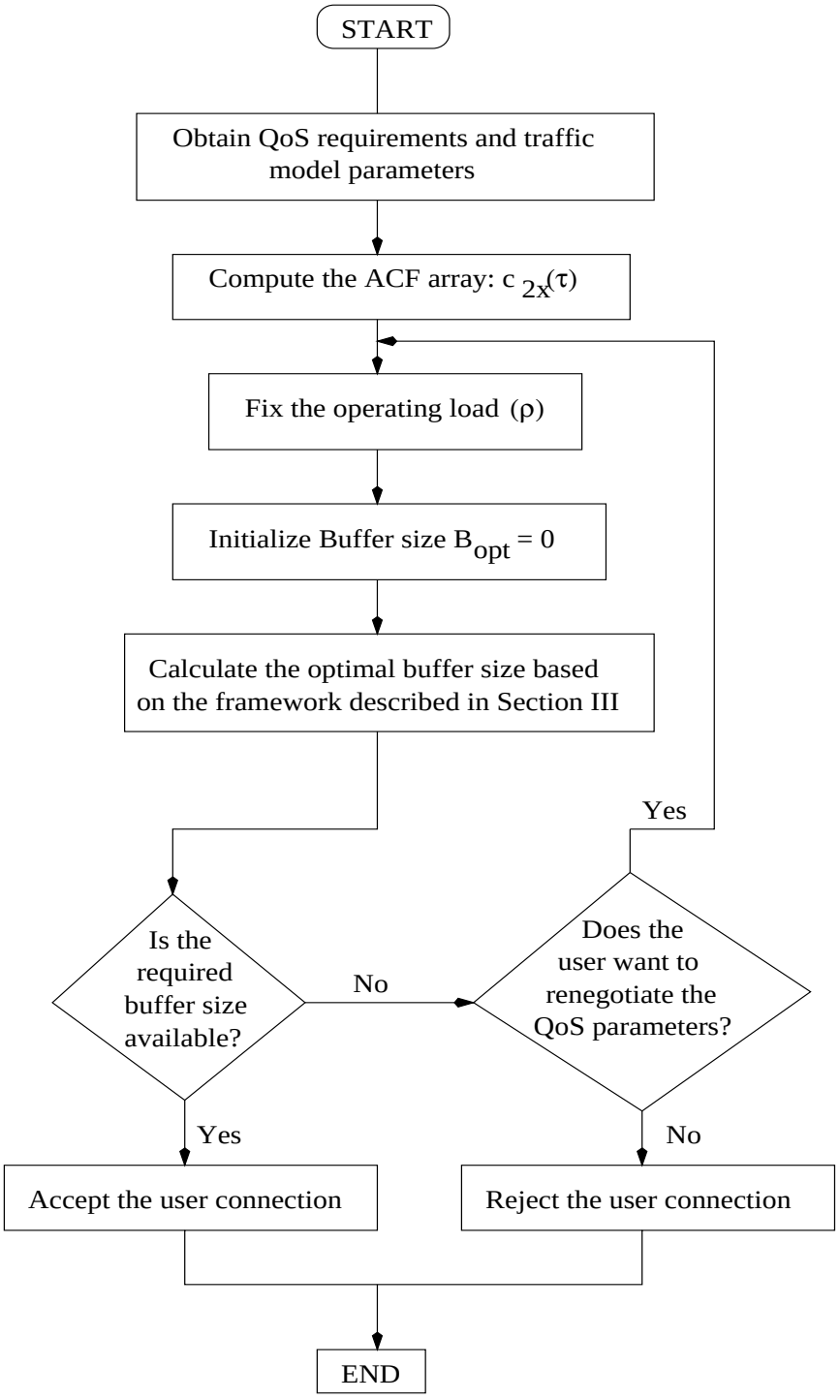

Fig. 3. Flowchart of the resource allocation algorithm.