ELSEVIER

Computer Communications 24 (2001) 1283–1298

computer communications

www.elsevier.com/locate/comcom

Adaptive resource negotiation based control for real time applications

T.-Y. Tan, T.-H. Cheng*, S.K. Bose, T.-Y. Chai

School of Electrical and Electronic Engineering, Network Technology Research Centre, Nanyang Technological University, Nanyang Avenue, Singapore, Singapore 693798

Received 5 September 2000; revised 12 November 2000; accepted 13 November 2000

Abstract

In this paper, we focus on providing quality-of-service (QoS) control in terms of packet loss and delay for interactive real-time applications. We propose a new scheme to find the optimal values of token bucket parameters, token generation rate r and token bucket size b, from the observed real-time traffic. This can be adjusted based on the user's QoS requirement and the different classes of real-time applications. Based on these optimal token bucket parameters, we introduce an adaptive resource negotiation control scheme. Our proposed admission control scheme offers the re-negotiation feature in the resource reservation process allowing higher admission ratios and higher resource utilization to be achieved.

Keywords: QoS; Resource negotiation; Token bucket

1. Introduction

The 'best-effort' quality-of-service (QoS) of existing packet-switched networks, such as Internet, does not provide good support for continuous transmission of data from interactive, real-time network applications. To provide QoS support, the admission control mechanism must ensure that a new flow will get its requested QoS without violating those for the ones which already exist in the network. Current approaches to admission control are typically either parameter-based or measurement-based [1]. A parameter-based approach depends on a priori traffic characteristics. The drawback is that much bursty traffic cannot be accurately characterized in advance. A measurement-based approach [1–4] focuses on achieving higher resource utilization for real-time applications that can tolerate some QoS deterioration. It is based on measurements of actual traffic. A measurement-based admission control procedure may therefore be able to utilize the bandwidth available for real-time traffic better.

In the general IntServe framework put forth in RFC 2215, there is way for data senders to describe the parameters of traffic it expects to generate, and by QoS control services to describe the parameters of traffic for which the reservation should apply [5]. The parameters that are defined based on the *Token Bucket Filter* model are collectively known as TOKEN_BUCKET_TSPEC Traffic Specification. The TOKEN_BUCKET_TSPEC takes the form of token bucket specifications, i.e. Token Generation Rate r and Token Bucket Size b, plus three other parameters, namely, Peak Bit Rate P, Maximum Packet Size M, Minimal Policing Unit m. These parameters are self-explanatory and their exact definitions are given in [5]. Assume that only the two token bucket parameters of r and b are used for flow characterization, we propose measurement-based admission control scheme to handle changes in the network conditions by tuning r and b to automatically adapt the conservativeness according to the observed real-time source traffic, the different mixtures of QoS requirements and other performance measures.

Section 2 presents the method for obtaining an optimal value of the token bucket parameters $\{b,r\}$ for our adaptive resource negotiation admission control scheme. Our proposed admission control algorithm is then described in Section 3. This has been tested through simulations using a wide variety of network topologies and source traffic models [4]. Section 4 presents these simulation results and shows that our proposed scheme provides reasonably good performance in terms of both utilization and QoS parameters. The concluding section reports our findings on the effects of various system and traffic parameters on our proposed admission control algorithm.

* Corresponding author.
E-mail addresses: tytan@ieee.org (T.-Y. Tan), ethcheng@ntu.edu.sg (T.-H. Cheng), skb@ieee.org (S.K. Bose), pg786608@ntu.edu.sg (T.-Y. Chai).

0140-3664/01/$ - see front matter
PII: S0140-3664(00)00362-5

2. Traffic characterization using token bucket model

In our proposed admission control scheme, each source is characterized using a token bucket with parameter $\{b,r\}$, where b is the token bucket size (in bits) and r is the token generation rate (in bits/s). We assume that packets are of random length and that the source can only inject a complete data packet into the network whenever it actually sends such a packet. This implies that a packet cannot be transmitted if the number of currently available tokens is less than the packet size. Such a packet will queue in the input buffer until the number of tokens accumulated becomes equal to or greater than the packet size. (Note that a token serves only one bit of the packet). We assume that the token bucket can only hold a maximum of b tokens; tokens are discarded when the token bucket overflows. Note that a source may also generate packets in bursts. If this happens such that the total number of bits is greater than b then only the first few packets may get enough tokens to be immediately allowed on to the network; the others will have to wait until sufficient tokens become available for them.

Packets queued in the input buffer are served in a FCFS fashion and may experience some queuing delay. The parameters of the token bucket should be chosen (according to the source's traffic pattern) so that the delay incurred in this queue is reasonably small. In view of this, a dynamic and flexible mechanism to adjust the token bucket's parameters may be desirable.

In some implementations, the size of the input buffer queue is intentionally set to 0, so that a packet is not delayed. In this case, a packet is cut-off and dropped if enough tokens are not available when the packet is generated. The *cut-off rate* of a source will then be the fraction of packets that are cut off due to an empty token bucket. This is an important indicator as to whether the token bucket parameters have been suitably chosen for that source. In our proposed scheme, we consider two application-classes where the input buffer queue of Class 1 applications is set to 0. Class 2 applications allow a worst-case queuing delay of 10 ms for the input buffer. Here the worst-case queuing delay is defined as Q/r where Q is the size of the input buffer.

2.1. Upper bound and lower bound for b and r

In this section, we obtain the worst-case upper bound and lower bound for r and b. We characterize a given traffic pattern using a token bucket with dynamically adjusted parameters so that there is almost no packet loss or delay in the flow. This is referred to as the *no-delay* requirement in the rest of this paper.

For a particular source, consider measurements being carried out starting at time t_0 and the measurement interval is such that there are T packet arrivals during this time. Let t_n be the arrival instant of the nth packet of size P_n in this interval. Let b_0 denote the initial number of tokens in the token bucket at time t_0. Intuitively, if the token generation rate r is too small, then irrespective of the value of b there will not be enough tokens in the bucket for all the incoming packets. Assuming that b and r are both set appropriately such that there is no loss of token due to token bucket overflow and that the basic measurement block is a measurement window of T packet transmission units, the necessary conditions to satisfy the no-delay requirement will then be

$$b_0 + r(t_n - t_0) \geq \sum_{i=1}^{n} P_i, \qquad \forall n > 0 \tag{1}$$

for

$$r \geq r_{\min} = \max_{1 \leq n \leq T} \left(\frac{\sum_{i=1}^{n} P_i - b_0}{t_n - t_0} \right) \tag{2}$$

Here, t_0 is the instant at which the measurement block begins, t_n the arrival epoch of the nth packet that arrives after t_0, and T is the total number of packets that arrive in the measurement interval. If the initial token count is set to 0, $r_{\min}$ is the average bit rate of the traffic pattern.

An adequately high value of r is desirable but the value should not be larger than necessary, as then many tokens would be lost due to token bucket overflow. The largest value of r needed is when the previous packet has used up all the tokens in the token bucket and enough new tokens have to be generated for the use of a new packet, i.e.

$$r(t_n - t_{n-1}) \leq P_n, \qquad n \geq 1 \tag{3}$$

Therefore, the maximal value of r would be

$$r_{\max} = \max_{1 \leq n \leq T} \left(\frac{P_n}{t_n - t_{n-1}} \right) r_{\max} \tag{4}$$

Note that if the initial token count is set to 0, then $r_{\max}$ would be the peak bit rate of the source.

Since the system will only inject complete packets into the network, the minimum number of tokens must be greater than the packet size. When r takes its maximum value, b can take its minimum value given by

$$b_{\min} = M \tag{5}$$

where M is the maximum packet size, as defined in the IETF Integrated Services Specification [5].

Let $b_n(r)$ denote the number of tokens accumulated before the arrival of packet n. Based on the assumption that there is no token bucket overflow,

$$b_n(r) = b_0 + r(t_n - t_0) - \sum_{i=1}^{n-1} P_i \tag{6}$$

Therefore, for $r \in [r_{\min}, r_{\max}]$, the maximum value of b we

would ever need is

$$b_{\max}(r) = \max_{1 \le n \le T} b_n(r) \tag{7}$$

Note that, $b_n(r) \ge M$ because $r \le r_{\max}$.

We therefore conclude that, for a given source with traffic pattern $\{(P_k, t_k) | 1 \le n \le T\}$ in the time interval $[t_0, t_T]$, the valid range for the token bucket rate is $[r_{\min}, r_{\max}]$. For any $r \in [r_{\min}, r_{\max}]$, the optimal token bucket size b satisfying the no-delay requirement would be between $b_{\min}$ and $b_{\max}$.

2.2. Determination of optimal value for b given r and queue size

Let $d_n = t_n - t_{n-1}$ denote the inter-arrival time between packets $n-1$ and n, where $n = 1, 2, \ldots, T$. Let $b_n(r)$ represent the number of tokens in the token bucket at time t_n^-, where t_n^- is the time instant just before the arrival of packet n. Since $b_n(r)$ cannot exceed the token bucket size b, we have

$$b_n(r) = \min(b_{n-1}(r) + rd_n - \sum_{i=J_{n-1}+1}^{J_n} P_i, b), \tag{8}$$

$$\text{if } J_n > J_{n-1}, \quad b_n > 0$$

$$b_n(r) = \min(b_{n-1}(r) + rd_n, b), \qquad \text{if } J_n = J_{n-1} \tag{9}$$

Here J_n is the index of the last packet that has been transmitted before the arrival of packet n at time t_n^-. Note that $J_n = J_{n-1}$ refers to the situation where no packet is transmitted in d_n.

If a non-zero queue size Q is allowed in the host's input buffer queue, then for a no-loss condition, Q should be large enough to accommodate the total number of bits that are still waiting for tokens. Thus, we have

$$b_n(r) + Q \ge \sum_{i=J_n+1}^{n} P_i \tag{10}$$

where $\sum_{i=J_n+1}^{n} P_i$ is the number of bits yet to be sent right after the arrival of packet n.

From Eqs. (8) and (10), we get

$$b_n(r) = \min(b_{n-1}(r) + rd_n - \sum_{i=J_{n-1}+1}^{J_n} P_i, b)$$

$$\ge \sum_{i=J_n+1}^{n} P_i - Q, b_n > 0 \tag{11}$$

The inequality is satisfied if

$$b_{n-1}(r) + rd_n - \sum_{i=J_{n-1}+1}^{J_n} P_i \ge \sum_{i=J_n+1}^{n} P_i - Q \tag{12}$$

and

$$b \ge \sum_{i=J_n+1}^{n} P_i - Q \tag{13}$$

Evaluating the inequalities for $n = T, T-1, T-2 \ldots 1$ [6], for any value of $r \in [r_{\min}, r_{\max}]$, the minimum value of b to satisfy the no-loss condition with queue size Q would be

$$b_{\text{opt}}(r) = \max_{\forall (g,h) \in \{g,h: 1 \le g < h \le T\}} \left(\sum_{i=J_g+1}^{h} P_i - r(t_h - t_g) - Q \right), b_{\text{opt}}(r) > 0 \tag{14}$$

If we need to consider a no-queue system, then from Eq. (14)

$$b_{\text{no-Q}}(r) = \max_{\forall (g,h) \in \{g,h: 1 \le g < h \le T\}} \left(\sum_{i=g}^{h} P_i - r(t_h - t_g) \right), b_{\text{no-Q}}(r) > 0 \tag{15}$$

Intuitively, $b_{\text{no-Q}}(r) > b_{\text{opt}}(r)$, as a no-queue system will reserve more resources than a queuing system implying a tradeoff between resource utilization and QoS performance. With queuing, bursty traffic would be smoothened so that the derived token bucket parameters are less demanding than those of the no-queue case, and the network resources are better utilized.

3. Proposed scheme for admission control and resource renegotiation

3.1. Admission control decision

In this paper, we assume that the Internet traffic consists of both real-time and non-real-time applications. Only the real-time applications require explicit admission control procedures, whereas the non-real-time applications continue to use the traditional best-effort service.

It should be pointed out that a measurement-based admission control scheme is best suited for links with a high level of aggregation. If the level of aggregation is low, the measured arrival rate may stay low for a long period of time even though the mean arrival rate is considerably higher. If this happens then the system may admit more new flows than it should. This may subsequently create problems when the arrival rate returns to the more normal higher levels. For this reason, for a link with a low level of aggregation, the admission control procedure could simply be based on the peak rates of the admitted flows. For links with a higher level of aggregation measurement-based admission control procedures can be of substantial benefit.

The admission decision is based on the worst-case traffic of a new flow α. This is modelled by a token bucket $\{b_\alpha, r_\alpha\}$, where b_α is the token bucket size of the new flow and r_α is its token bucket rate. In Ref. [7], it has been proved that for a flow with token bucket parameter $\{b_\alpha, r_\alpha\}$, the worst-case queuing delay (not including the transmission time of the

packet concerned) is b_α/r_α, if a minimum bandwidth of r_α is reserved for this flow. In this worst case, as long as the network can buffer the burst size of b_α and provide a sustained rate that is at least the same as r_α, the worst-case delay can be guaranteed and no packet from the source would be lost.

For a link with a total capacity C, the following conditions guide the admission decision:

Resource condition. The total resulting bandwidth usage due to the admission of the new flow may not exceed mC, where m is a selected fraction of the link capacity. Let ϱ_i denote the estimated aggregate traffic rate for class i of the existing flows and ϱ_α be the traffic rate for the new flow α (note that $\varrho_\alpha = r_\alpha$), then

$$mC \geq \sum \varrho_i + \varrho_\alpha \tag{16}$$

QoS condition. Admission of new flow should not result in violation of the delay bound at the same priority level. If we assume that flow α is from class k, then from the worst-case delay estimation, we can estimate the new delay of class k as in Ref. [4] to be

$$D'_k = \frac{\sum_{i=1}^{k-1} b_i}{C - \sum_{i=1}^{k-1} \varrho_i} + \frac{b_k + (b_\alpha)_k}{C - \sum_{i=1}^{k-1} \varrho_i} = \hat{D}_k + \frac{(b_\alpha)_k}{C - \sum_{i=1}^{k-1} \varrho_i} \tag{17}$$

where

$$\hat{D}_k = \frac{\sum_{i=1}^{k-1} b_i - b_k}{C - \sum_{i=1}^{k-1} \varrho_i}$$

is the aggregate Class k delay before admitting flow α, and can be obtained by measuring the aggregate rate for the existing traffic flow with the measurement process (described in next section). From Eq. (17), we see that the delay of Class k traffic is linearly proportional to b_α after admitting the new flow.

We use the static priority scheme for the investigation of this measurement-based admission control scheme. Therefore, the admission of new flow α of Class k will also affect the delay of Class j that has a lower priority, (i.e. $j > k$) than Class k. The implies that the admission of new flow α must not violate the delay conditions of the lower Class j flows. We have

$$D'_j = \frac{\sum_{i=1}^{k-1} b_i + b_k + (b_\alpha)_k + \sum_{i=k+1}^{j} b_i}{C - \sum_{i=1}^{k-1} \varrho_i - \varrho_k - (\varrho_\alpha)_k - \sum_{i=k+1}^{j-1} \varrho_i}$$

$$= \hat{D}_j \frac{C - \sum_{i=1}^{j-1} \varrho_i}{C - \sum_{i=1}^{j-1} \varrho_i - (\varrho_\alpha)_k} + \frac{(b_\alpha)_k}{C - \sum_{i=1}^{j-1} \varrho_i - (\varrho_\alpha)_k} \tag{18}$$

where $\hat{D}_j$ is the aggregate Class j delay before admitting flow α. Note that the new flow α will not affect the queuing delay of the flows that have a higher priority than Class k.

3.2. Parameters measurement process

For this we use a scheme similar to that used in Ref. [4] to measure residual resources. The delay performance and the aggregate traffic are measured periodically in real-time and are adopted as the estimated worst-case delay $\hat{D}_i$ and the estimated aggregate traffic $\sum \varrho_i$, respectively. We define a unit interval to be the packet transmission time and the averaging period to be S units. A delay measurement sample is obtained after each packet transmission block. Each measurement block contains T packet transmission units. Note that $T = nS$, where n is the number of averaging periods in each measurement block. At the end of each measurement block, the highest measured delay of all the transmission units is adopted as the estimated delay and the highest aggregate traffic rate of all the averaging periods is taken as the estimated aggregate traffic rate in the admission control decision.

The measurement process is summarized as follows:

1. *Measuring delay.* The measurement variable $\hat{D}_i$ tracks the estimated maximum queuing delay for Class i. We use a measurement window of T packet transmission units as our basic measurement block. The value of $\hat{D}_i$ is updated at the end of the measurement block to reflect the maximal packet delay seen in the measurement block, i.e.

$$\hat{D}_i = \max_{1 \leq k \leq T} (d_k) \tag{19}$$

if d_k is the packet queuing delay for the kth transmission unit in the measurement block.

2. *Measuring rate.* The measurement variable $\sum \varrho_i$ tracks the highest sampled aggregate rate for Class i. Each measured sample is obtained at the end of an averaging period of size S units by counting the average bit rate of S transmission units. A measurement block of size $T(= nS)$ units contains n measured samples denoted as $S_1, S_2, \ldots S_n$. At the end of a measurement block, we update $\sum \varrho_i$ using the highest average bit rate among

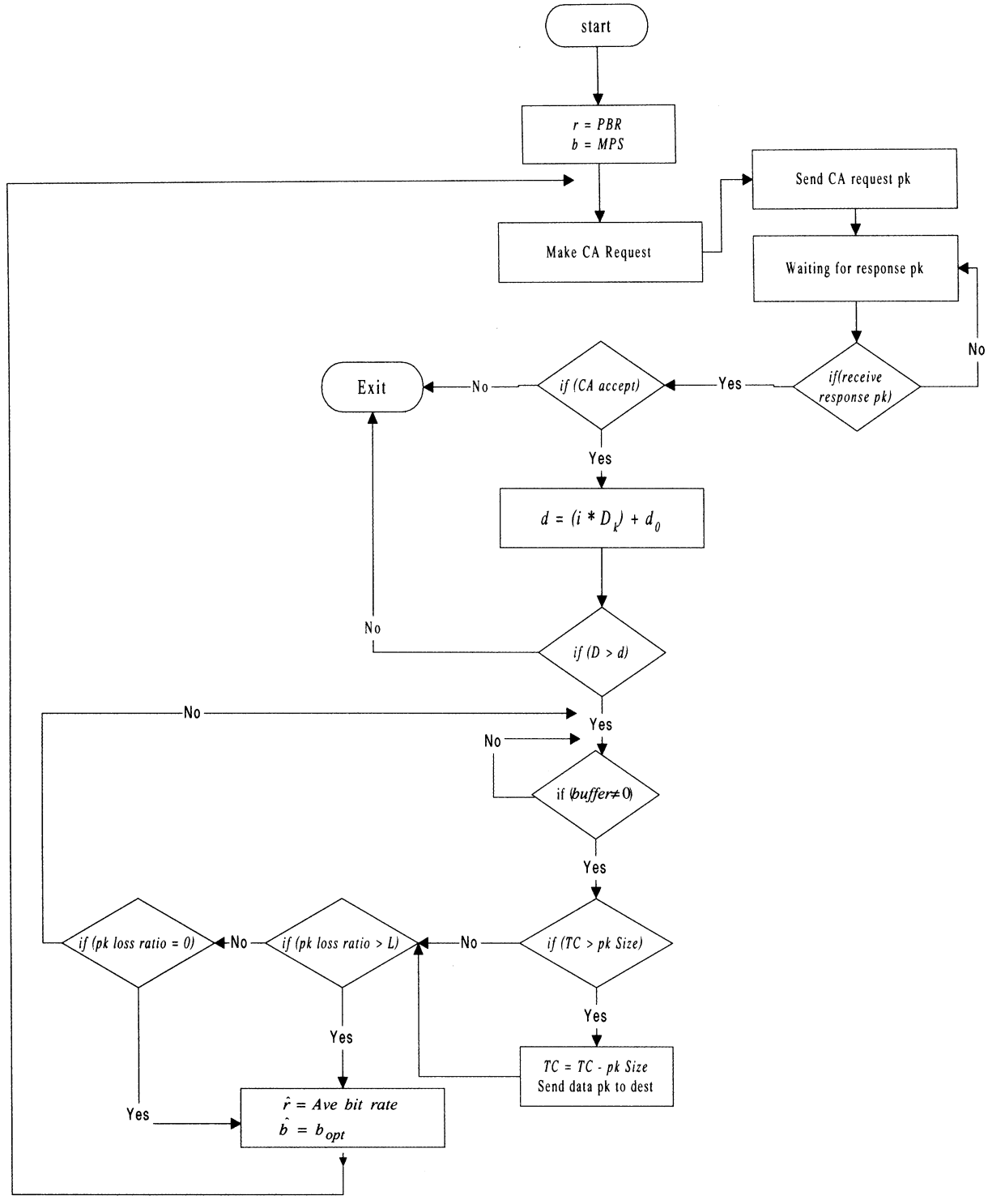

Fig. 1. Adaptive resource negotiation admission control scheme (host).

the n averaging periods in S:

$$\varrho_k = \frac{\sum_{m=1}^{S} P_m}{t_S - t_1} \tag{20}$$

$$\sum \varrho_i = \max_{1 \le k \le n} (\varrho_k) \tag{21}$$

where P_m is the packet size of the mth packet and t_s is the arriving time of the last packet in the averaging period of size S units.

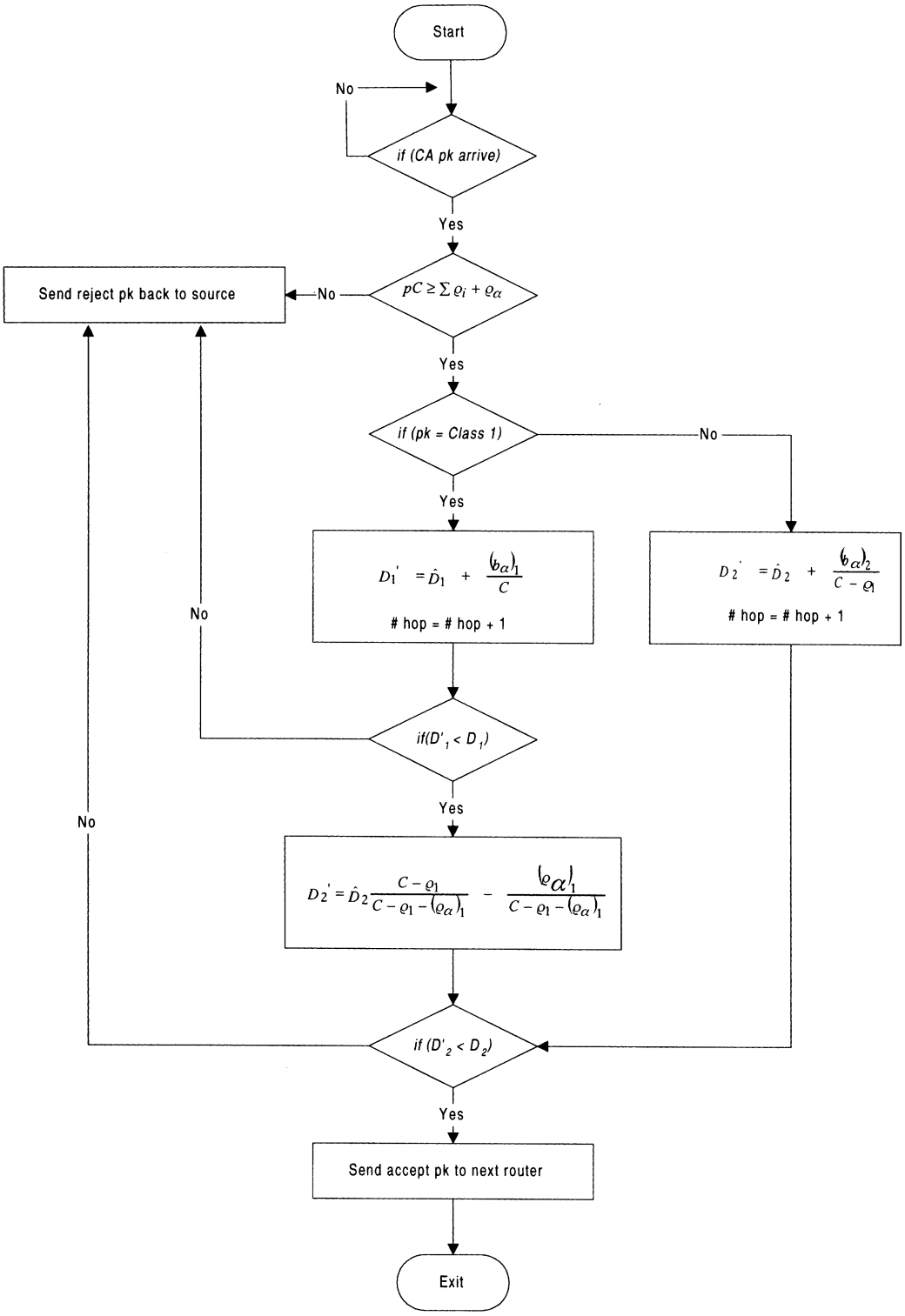

Fig. 2. Adaptive resource negotiation admission control scheme (intermediate node).

For the following special cases, we allow the estimates to be updated more quickly to improve the responsiveness of our proposed scheme.

- If a new flow is admitted in the middle of a measurement block, the delay and rate previously estimated (i.e. the ones adopted at the end of the previous measurement block) are updated *immediately* using equation $D_i' = \hat{D}_i + b_\alpha/r_\alpha$ and $\varrho_i' = \sum \varrho_i + \varrho_\alpha$.
- When a single measurement of the averaging period in

the measurement block exceeds the currently adopted estimate, the current estimate is updated *immediately*. A more conservative admission control would be to use a value somewhat larger than the sampled value. For example, in the simulation study described in Ref. [4], if a single packet delay exceeds the delay estimate currently in use, the delay performance estimate value D_i' is changed to twice the value of this delay sample. However, in our simulations, we found that it is sufficient to change the D_i' value to the exact value of this delay sample to adapt to the sudden change in this traffic information. This yields higher utilization than the scheme proposed in [4] without any observable negative effects.

3.3. Resource re-negotiation process

The resource re-negotiation process is activated either when the packet loss ratio at the source is more than the user-specified packet loss ratio L or on observing no packet loss for a certain period of time. We set the initial token rate r equal to $r_{\max}(=$ Peak Bit Rate, $P)$ and initial token size equal to $b_{\min}(=$ Maximum Packet size, $M)$. As mentioned earlier, this set of values for the token bucket parameters satisfies the no loss and no delay condition. Subsequently, when the resource re-negotiation is triggered, the value of r used is $\sum_{i=1}^{T} P_i/(t_r - t_0)$ from the actual measurements and the optimal value of b is computed from Eq. (14). These are used during the resource re-negotiation process. The optimal value of the token bucket parameters may also be adjusted depending on the user's QoS requirement and the real-time traffic conditions.

Flowcharts of our proposed adaptive resource negotiation admission control scheme for the host system and an intermediate node are shown in Figs. 1 and 2, respectively. Initially, the user specifies its QoS parameter requirements (e.g. L and D) and application parameters (e.g. P and M). The source will set the initial token rate and token size value to $r_{\max}$ and $b_{\min}$, respectively. A call admission 'request packet' will be sent from the source to the destination. This packet consists of the information on token bucket parameters and QoS parameters. When an intermediate node (e.g. router) receives the call admission packet from the source or from the previous node, the *resource condition* and *QoS condition* are considered in making the admission decision. The call admission request packet is relayed to the next router only if both conditions are satisfied: otherwise a 'reject packet' is returned to the source. The request is admitted only if all the network nodes on the path admit the flow. In that case, the 'request packet' eventually reaches the destination and the destination will send an 'accept packet' back to the source. The source will start sending data packet to the destination when it receives the 'accept packet' from the destination. Subsequently, it will detect the packet loss ratio periodically. The resource re-negotiation process will be activated whenever the packet loss ratio exceeds the user-specified packet loss ratio L or if there is no packet loss for a certain period of time. The optimal values of r and b will be calculated and used for the resource re-negotiation process. The resource re-negotiation process follows the same procedure as the call admission process described above. However, in this case even if the 'request' is rejected, the source may still continue to send data packets using its previous parameters.

4. Simulation model and results

Measurement-based admission control algorithms need to be verified through experiments on either a real network or a simulator. In our case, we used a discrete event simulator (OPNET) to run simulations and compare our results with those of given in Ref. [4]. We have tested our algorithm through simulations on a wide variety of network topologies and source models. We have considered six different source models, three network topologies and several different mixes of traffic. In each case, our scheme achieves a reasonable degree of utilization with a low delay bound violation rate. It should be noted that because of its *resource re-negotiation* feature, our proposed scheme performs better than the corresponding system in Ref. [4].

Our simulation study was carried out with three different source models, as in Ref. [4]. These are all ON–OFF processes and only differ in the distributions of the ON Times and the call holding times (CHT). The simplest of these is a 2-state Markov process with an ON–OFF model. The ON period (T_{ON}) is an exponentially distributed random variable in which packets are generated at a fixed inter-arrival time $1/Ps$, where P is the Peak Rate of the application. This means that we generate $T_{\mathrm{ON}} * P$ packets for that ON period with one packet being generated in every $1/P$ seconds. For each of these packets, the packet size is chosen to be a random variable uniformly distributed between 1000 and 2000 bits. The OFF period (T_{OFF}) is also an exponentially distributed random variable with a mean value of I milliseconds during which no packets are generated. The overall duration of each flow is also exponentially distributed with a mean value of H milliseconds.

Several studies have shown that network traffic often exhibits long-range dependency (LRD), with the implication that congested periods may be long and a slight increase in the number of active connections may result in a large increase in packet loss rate [8]. It has also been shown in Ref. [8] that traffic with a long-range dependence may have some undesirable effects if a measurement-based admission control algorithm is used. In order to investigate this and other LRD-related effects, we have included two LRD source models in our simulation studies, these are the Pareto and the Fractional ARIMA models. The first LRD model is an ON–OFF process with Pareto distributed ON and OFF times. During each ON period, a Pareto distributed number of packets, with mean N and Pareto shape parameter β, are

Table 1
Six instances of three source models

Model name	Model's parameters						
	P (Packets/s)	I (ms)	N (Packets)	β	γ	Mean (packets)	Variance (packets)
EXP1	64	325	20				
EXP2	1024	90	10				
EXP3	∞	684	9				
POO1	64	2925	20	1.2	1.1		
POO2	256	360	10	1.9	1.1		
fARIMA (0.75, 0.15, −)	∞	125				8	13

generated at the peak rate of P packets/s. The OFF durations are also Pareto distributed with a mean value of I milliseconds and the shape parameter γ. Pareto shape parameters less than 1 give data with infinite means, while a shape parameter less than 2 results in data with infinite variance. Each Pareto ON–OFF source by itself does not generate a LRD series. However, the aggregation of them does [9]. The Pareto location parameter is $mean * (shape\text{-}1)/(shape)$ as given in Ref. [8]. For all Pareto ON–OFF sources, the shape parameter for the Pareto distributed ON time (β) is selected following the observation made in Ref. [4]. In our simulation, we use γ (shape parameter for OFF time) of 1.1 for both Pareto ON–OFF sources.

The other LRD model is Fractional ARIMA. We use each number generated by the *fractional auto-regressive integrated moving average process*(*fARIMA*) [10] as the number of packets to be sent in each ON period. Inter-arrival times of packets during the ON periods are fixed. As in Ref. [4], we generate a series of 15 000 fARIMA data points at the beginning of each simulation. Each fARIMA source then picks a uniformly distributed number between 1 and 15 000 to be used as its index into that series. On reaching the end of the series, the source wraps around to the beginning. The fARIMA model takes three parameters: the auto-regressive process order with the corresponding set of weights, the degree of integration and the moving process order with the corresponding set of weights. The fARIMA model generates long-range dependent series. However, the marginal distribution of fARIMA generated series is Gaussian. We first generate a normally distributed innovation with mean N and standard deviation S packets. If the minimum of the fARIMA output is less than zero, we shift the whole series by adding the absolute value of its minimum to every number in the series. This way of obtaining non-negative series has been used in Ref. [11]. This shifting process constraints the maximum value of the generated series to be always twice its average. As in Ref. [4], for the fARIMA source, we use an auto-regressive process of order 1 (with weight 0.75) and degree of integration 0.15. The first-order auto-regressive process with weight 0.75 implies that our fARIMA traffic will also have strong short-range dependence, while maintaining stationarity [12]. The inter-arrival time between ON periods is 1/8 s. The Gaussian innovation fed to the fARIMA process has a mean of eight packets with standard deviation 13.

We considered six instances of the above three source models, as summarized in Table 1. The EXP model is an ON–OFF model with exponentially distributed ON and OFF time, and POO model is the Pareto ON–OFF model. In the table, $p = \infty$ indicates that the packets are transmitted back to back during the ON period.

We ran our simulations on three topologies. These are the ONE-LINK, TWO-LINK and FOUR-LINK topologies depicted in Figs. 3–5. In these topologies, all the sources are connected to an intermediate node by an infinite-bandwidth link. The connections between nodes in these topologies are all 100 Mbps links, with infinite buffers. As shown in Fig. 3, the ONE-LINK topology traffic flow is from Source A to Source B. In the TWO-LINK case, traffic flows between three host pairs (in source-destination order)

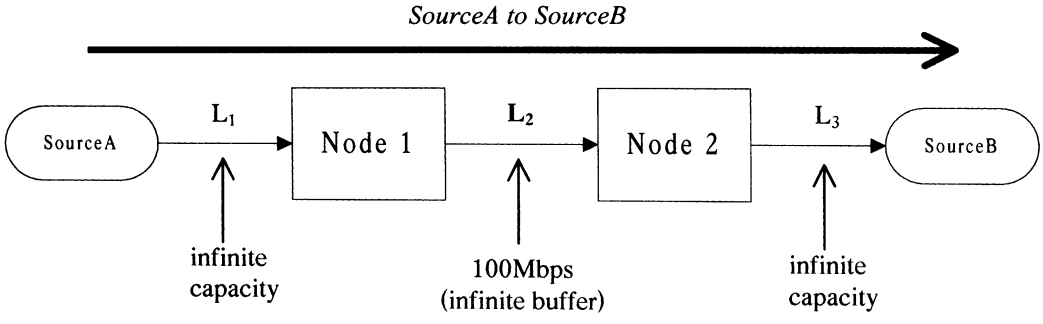

Fig. 3. ONE-LINK topology.

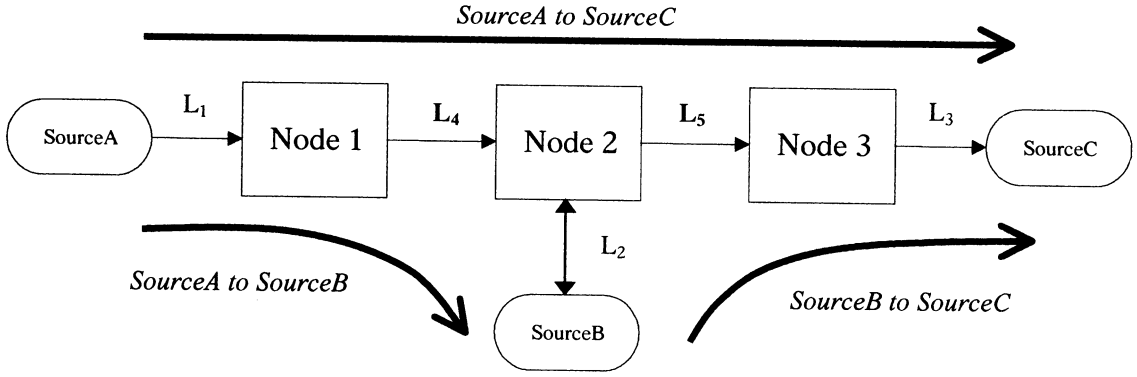

Fig. 4. TWO-LINK topology.

— Source A to Source B, Source B to Source C and Source A to Source C. Flows are assigned to one of these three host pairs with uniform probability. Fig. 5 shows the FOUR-LINK topology, which consists of six host pairs: Source A to Source C, Source B to Source D, Source C to Source E, Source A to Source D, Source B to Source E and Source D to Source E. Flows are distributed among six host pairs with uniform probability. The directed curved lines indicate the host pairs and the paths that the packets traverse.

4.1. Homogeneous sources with single-hop

These simulations are for the ONE-LINK topology of Fig. 3 assuming that the sources are homogeneous, i.e. they have the same traffic model and ask for the same kind of service (e.g. Class 1 or Class 2). Each flow also has the same QoS requirement. For each source, three kinds of simulations were run — the first has all sources requesting Class 1 service, the second has all sources requesting Class 2 service and the third where sources request the predictive service as given in Ref. [4] where delay violations may be tolerated.

Except for the LRD sources, the other simulations were run for 3000 s of simulation time with reported data taken only from the later 1500 s. The initial 1500 s were provided for the simulation to warm up [4]. The data presented are obtained by averaging the results from ten independent runs with random seed numbers. However, simulation with long-range dependent sources requesting predictive service requires longer warm-up periods. Accordingly, we ran all simulation involving LRD sources for 10 000 s of simulation time, with reported data taken only from the later 5000 s. The average holding time of all exponential sources is 300 s. The LRD sources have log-normal distributed holding times with median 300 s and shape parameter 2.5. For all sources, the inter-arrival time between flows is exponentially distributed with an average of 400 ms. Confidence interval estimates were also obtained for some of the simulation experiments to verify the reliability of the results. These were found to be sufficiently tight to ensure the reliability of the mean results shown.

The averaging period S controls the sensitivity of our rate measurements. In our simulation, we use $S = 100$ as in Ref. [4]; this is a compromise value between choosing small S for

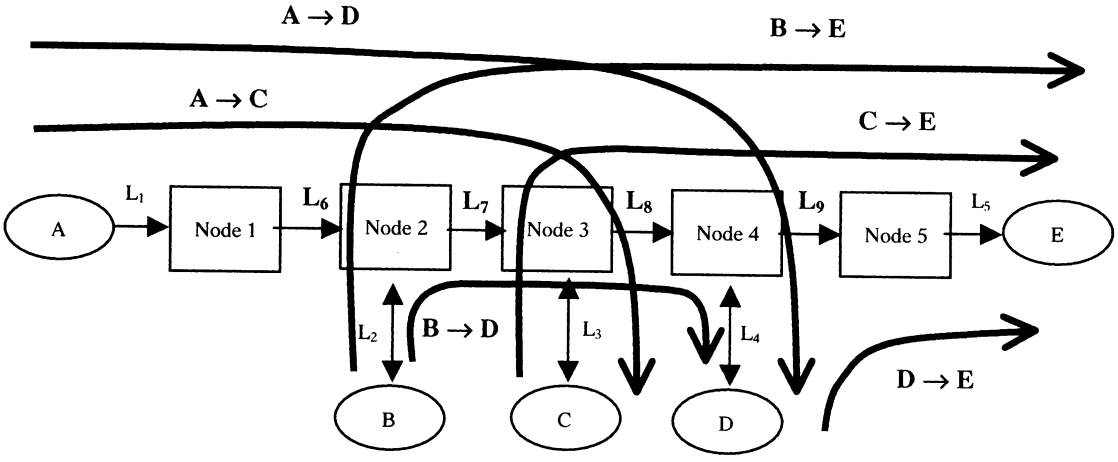

Fig. 5. FOUR-LINK topology.

Table 2
Single-hop homogeneous sources simulation results

Model's name	Utilization				Delay (ms)				Packet loss ratio			
	Class 1	Class 2	Predictive service	Reference	Class 1	Class 2	Predictive service	Reference	Class 1	Class 2	Predictive service	Reference
EXP 1	0.8012	0.8311	0.7933	0.8	1.56	1.95	1.97	3	3.75×10^{-4}	6.58×10^{-3}	0	0
EXP 2	0.7046	0.7946	0.7548	0.76	18.4	29.2	30.5	42	6.72×10^{-4}	5.29×10^{-3}	2.04×10^{-3}	2.10×10^{-3}
EXP 3	0.5728	0.6111	0.5919	0.62	13.5	31.9	22.6	33	9.89×10^{-4}	5.85×10^{-3}	9.23×10^{-5}	9.40×10^{-5}
POO 1	0.7553	0.7926	0.7329	0.74	4.77	4.97	4.22	5	9.56×10^{-4}	5.98×10^{-3}	0	0
POO 2	0.5798	0.6871	0.6329	0.64	5.79	12.6	7.66	8	7.66×10^{-4}	6.55×10^{-3}	3.96×10^{-5}	4.50×10^{-5}
tARIMA	0.7302	0.8324	0.7967	0.81	19.3	65.2	62.3	72	1.58×10^{-3}	6.28×10^{-3}	1.15×10^{-2}	1.10×10^{-2}

higher burst sensitivity and large S for better averaging. The size of T controls the adaptability of our measurement mechanism to changes in traffic load with smaller T for more adaptability and larger ones for greater stability. Following the suggestion in [13] that T/S should be greater than or equal to 10, we have used $T = 1000$ in our simulations. As mentioned earlier and in Ref. [13], we assume the Internet traffic to consist of both real-time and non real-time applications where only real-time applications require explicit admission control procedures; the other non-real-time applications will continue to use the traditional best-effort service. In our simulation, we assume that the links reserve 90% of their capacity for the use of real-time traffic and the rest is for non-real-time traffic. This means that the utilization target $m = 0.9$ is used in Eq. (16). The adaptive tuning of these parameters is considered in the next section.

For this case, Table 2 gives the packet link utilization and maximum experienced delay in the bottleneck link ($L2$). The packet loss ratio column indicates the ratio of the average number of packets that would have been dropped at the input by each flow's token bucket filter and the total number of packets sent by the flow. The 'Reference' column lists the corresponding simulation results of Ref. [4], for the purpose of comparison. Our delay bounds for Class 1 and Class 2 services are 20 and 200 ms, respectively. The objective of our scheme is *no-delay bound violation* while providing high utilization and low packet loss ratio. This can be seen in the results of Table 2. As expected, Class 2 service allows better network utilization than Class 1 at the cost of higher delays. Compared to the corresponding results of Ref. [4], the utilization of our scheme is essentially similar but with better delay performance — this is obtained at the cost of marginally higher values of packet loss. It should also be noted that our algorithm is *greedy*, always admitting a new flow when resources are available. This may discriminate against larger flows, as there may never be enough resources available for admitting them.

4.2. Homogeneous sources with multi-hop

Similar simulations have been conducted for the multi-hop topologies in Figs. 4 and 5. Table 3 shows the simulation results for the TWO-LINK topology for the source models EXP1, EXP3 and POO2. For the FOUR-LINK results of Table 4, the EXP2, POO1 and fARIMA source models were used. The results show the same trends as those observed for the single-hop case. In addition, we observe that the utilization levels of the upstream node are lower than those of the downstream ones (i.e. L_5 has a higher utilization that L_4 in Table 3 and L_9 has the highest utilization in Table 4). This shows that when flows traverse paths of different lengths, flows with longer paths would be less likely to be admitted as a flow will not be admitted if any one of the nodes along the path rejects the request. The network will then tend to admit flows that traverse relatively short paths.

Table 3
Multi-hop homogeneous sources simulation results (TWO-LINK)

Link no.	Model	Utilization				Delay (ms)			
		Class 1	Class 2	Predictive service	Reference	Class 1	Class 2	Predictive service	Refrence
L_4	EXP1	0.7061	0.7652	0.6811	0.67	2.069	3.893	2.059	2
	EXP3	0.3968	0.4577	0.4015	0.44	2.968	8.126	12.975	20
	POO2	0.5127	0.5606	0.5597	0.59	2.689	7.897	4.996	7
L_5	EXP1	0.8175	0.8174	0.7833	0.78	3.172	5.194	3.172	3
	EXP3	0.5283	0.6215	0.5566	0.58	2.969	9.568	29.735	30
	POO2	0.6701	0.7686	0.6915	0.70	3.394	8.934	8.987	17

4.3. Heterogeneous sources with single-hop

We ran three kinds of simulation with heterogeneous sources; single source model requesting multiple levels of service (i.e. Classes 1 and 2), multiple source models requesting a single class of service and multiple source models requesting multiple levels of service. In all cases, each model was given equal probability of being assigned to the next new flow generated. Table 5 shows the results obtained when flows with the same source model request both levels of service. In all cases, we were able to achieve high levels of utilization without incurring delay violations. Our delay bounds for Class 1 and Class 2 service are 20 and 200 ms, respectively. As expected, we also find that the presence of Class 2 traffic generally leads to better utilization values.

Table 6 shows the results for selected pairings of the source models for simulations with multiple sources requesting a single class of service. Table 7 similarly considers the case of multiple source models requesting multiple classes of service. In each of these cases, our scheme gives similar utilization values as in Ref. [4] without incurring delay violations; the delay performance is also given and is within the specified delay bounds.

4.4. Heterogeneous sources with multi-hop

These simulations were run with all the six source models on each of the topologies given. For this case, Table 8 shows the results obtained for some of the links for the same parameters as before. The trends observed in the results are the same as those seen earlier. In particular, we observe that the average delays are all within their given limits and have delay and utilization performance which are better than or comparable to those of Ref. [4].

Our overall results indicate that measurement-based admission control with resource re-negotiation scheme proposed by us can provide high utilization values with delays within the specified limits. This conclusion is justified through simulations for the different topologies and source models studied by us.

5. Effects of various system parameters

In this section, we present the simulation study of adaptive parameter tuning for the proposed admission control scheme and show that the choices of these parameters directly affect its performance.

Table 4
Multi-hop homogeneous sources simulation results (FOUR-LINK)

Link no.	Model	Utilization				Delay (ms)			
		Class 1	Class 2	Predictive service	Reference	Class 1	Class 2	Predictive service	Reference
L_6	EXP2	0.3831	0.4855	0.3975	0.42	2.082	4.183	4.099	6
	POO1	0.3611	0.3898	0.3168	0.31	1.556	2.290	1.548	1
	fARIMA	0.4298	0.5316	0.5203	0.54	11.077	21.078	15.270	36
L_7	EXP2	0.5370	0.7370	0.6468	0.71	2.778	12.736	12.958	31
	POO1	0.6209	0.6980	0.5902	0.66	3.095	5.580	3.095	2
	fARIMA	0.7554	0.7956	0.7673	0.77	15.600	46.500	19.690	40
L_8	EXP2	0.7226	0.7820	0.7299	0.72	3.408	14.099	12.164	24
	POO1	0.7895	0.8445	0.7665	0.75	4.665	7.915	4.642	7
	fARIMA	0.6631	0.7611	0.7598	0.74	7.644	17.042	15.990	29
L_9	EXP2	0.7246	0.7836	0.7244	0.71	5.051	15.550	13.620	31
	POO1	0.6434	0.6637	0.6434	0.59	2.439	4.426	2.449	2
	fARIMA	0.7839	0.8811	0.8104	0.80	15.884	35.448	23.225	44

Table 5
Single-hop, single source model requesting multiple levels of service

Model	Utilization		Delay (ms)		Packet loss ratio	
	Class 1 & 2	Reference	Class 1 & 2	Reference	Class 1 & 2	Reference
EXP 1	0.8171	0.77	2.104	20.007	6.20×10^4	6.44×10^3
EXP 2	0.7303	0.71	2.562	60.495	8.40×10^4	6.72×10^3
EXP 3	0.3835	0.31	2.963	83.581	8.74×10^4	2.74×10^3
POO 1	0.6865	0.70	1.873	2.165	7.03×10^4	8.82×10^3
POO 2	0.5821	0.60	2.953	10.933	4.09×10^4	7.28×10^3
fARIMA	0.7341	0.79	2.966	31.006	3.16×10^3	3.88×10^3

5.1. Effect of input buffer size (Q) on link utilization and delay

As shown in Eq. (14), the input buffer size Q is an important parameter in our proposed control scheme. The values of Q not only control the 'conservativeness' of our scheme, but also assign the different mixtures of QoS requirements to different classes of real-time applications. Since there is a tradeoff between resource utilization and QoS performance, we need to investigate the appropriate input buffer size to obtain the maximum resource utilization, without violating the delay requirement of the real-time application.

In order to study the performance of the proposed admission control algorithm, we need to run the simulations on a long time scale with different simulation seed numbers. We also need to simulate different network conditions to ensure that the simulation results reflect the general performance of the proposed scheme. For this investigation, we ran simulations with all the six source models, as described in the previous section, on the ONE-LINK topology. As usual, we ran all the simulation for 3000 s simulated time with reported data taken from the last 1500 s. The data presented in Fig. 6(a) and (b) are obtained by averaging the results of 10 different runs with random seed numbers. We use an averaging period S of 100 packet transmission times and a measurement block T of 1000 packet transmission times. The average holding time of all exponential sources is 300 s. The LRD sources have log normal distributed holding times with median 300 s and shape parameter 2.5. The other parameters are the same as in the previous section.

As shown in Fig. 6(a), the level of resource utilization increases with the input buffer size. The results show a scenario in which the sources requesting fewer resources can achieve higher level of utilization, as the resource requirement would be lower with larger input buffer. On the other hand, the queuing delay increases with increasing input buffer size. The no-queuing case produces a delay of 1.672 ms, which is able to satisfy the delay requirement for the interactive real-time application (see Section 4). For a multi-hop environment, we can approximate the total end-to-end queuing delay by multiplying the number of hops with the data obtained for one link. Confidence intervals for a confidence level of 98 and 99% were tested for some of the runs and were found to be sufficiently tight.

5.2. Effect of averaging period (S) on link utilization and delay

The averaging period S is used in getting the bandwidth measurement sample as the average bit rate over S transmission units. This controls the sensitivity of our rate measurement. With small S, burst sensitivity would be high whereas the traffic smoothening will be more for large values. Typically, the averaging period S should be much larger than a packet time and much smaller than the length of the shortest burst length. This is hard to determine in practice; a batch mean method of estimating the right value of S is given in Ref. [13].

As in the previous investigation, we ran our simulation with all the six source models on the ONE-LINK topology. All the simulation data are obtained from the later half of our 3000 s simulation time. We set the input buffer size Q for Class 1 and Class 2 traffic to be 0 and 100 kbits, respectively.

Table 6
Single-hop, multiple source models requesting single level of service

Model	Utilization				Delay (ms)	
	Class 1	Class 2	Predictive service	Reference	Class 1	Class 2
EXP1-EXP2	0.7681	0.8133	0.7601	0.75	0.823	2.628
EXP2-EXP3	0.7021	0.7091	0.7096	0.70	3.102	15.107
EXP2-POO2	0.6251	0.6415	0.6321	0.63	0.747	7.471
EXP2-fARIMA	0.8000	0.8021	0.8010	0.79	1.399	13.895
EXP3-fARIMA	0.7914	0.8407	0.8086	0.81	3.013	23.013
POO 2-fARIMA	0.7018	0.7183	0.7088	0.69	1.501	15.696

Table 7
Single-hop, multiple source models requesting multiple levels of service

Model	Utilization		Delay (ms)	
	Classes 1 and 2	Reference	Class 1	Class 2
EXP1-EXP2	0.7721	0.75	1.123	8.053
EXP1-fARIMA	0.7698	0.78	1.164	12.654
EXP1-POO2	0.5922	0.65	0.936	10.324
EXP1-POO1	0.6008	0.62	1.7011	7.082
EXP3-POO1	0.6327	0.60	1.0564	10.047
POO1-fARIMA	0.6681	0.65	1.117	12.244

While running the different sets of averaging period S in our simulations, we maintained the ratio of measurement block to averaging period at 10 ($T/S = 10$). As usual, we set the average holding time of all the sources to 300 s. In Fig. 7(a), we show the results of resource utilization as a function of the averaging period. The values are obtained from 10 different simulation runs with random seed numbers. The results demonstrate that there is an optimal size for the averaging period for which the utilization and delay performance is the best. However, this optimum value would be difficult to obtain directly without simulation studies or studies on an actual system.

Fig. 7(b) shows the relationship between the Classes 1 and 2 queuing delays as a function of the averaging period. It can be seen from the figure that the delay for Class 1 and Class 2 traffic is always within the given delay requirements. An averaging period size ranging from 100 to 300 transmission packets is acceptable as it provides nearly optimal results for resource utilization and queuing delay.

5.3. Effect of measurement block size (T) on link utilization and delay

The size of the measurement block T controls the adaptability of our measurement mechanism to drops in traffic load. Smaller T means more adaptability. Beside this, the study also shows that the value of T directly affects the conservativeness of the admission control scheme. Since the highest aggregate rate sample within the measurement block is chosen to be the estimate, a larger T implies a better aggregate rate estimate. This would result in more conservative admission control, which will therefore achieve smaller packet delays and lower resource utilization.

Basically, varying T cause two effects in our admission control algorithm. Firstly, since T is the length of the measurement block used to determine how long we keep the previous maximal packet delay and sampled utilization, increasing T makes these estimates more conservative. Thus, a larger T means fewer delay violations and lower link utilization. Secondly, T also controls the period that we continue to use our calculated estimate of the delay and utilization induced by a newly admitted flow. It should be noted that whenever a new flow is admitted, we increase the measured values to reflect the worst-case expectations and then restart the measurement window (Section 3). Thus, we are using the *calculated effects* of new flows rather than the *measured effects* until we survive an entire period of T without any new flow arrival. This means that if p is the peak flow reservation rate, and C the link capacity, we will only admit at most C/p number of flows and will not admit anymore flows until the end of a T period.

In order to examine the effect of measurement block T on resource utilization and QoS performance, we ran our simulations with all the six source models on the ONE-LINK topology. Since the averaging period for utilization measurement should allow for enough utilization samples, for all our simulations we maintain our averaging period at a constant value of 100 transmission packets. As usual, all the simulation data are obtained from the later half of our 3000 s simulation time. The input buffer size for Class 1 and Class 2 traffic is 0 and 100 kbits, respectively. The average holding time of all exponential sources is 300 s. The LRD sources have log normal distributed holding times with median 300 s and shape parameter 2.5.

In Fig. 8(a) and (b), we show the average link utilization and maximum experienced delay from 10 simulation samples with random seed numbers. We varied the measurement block T, from 500 to 2000 packet times. Notice that smaller T yields higher utilizations at higher experienced delays and larger T keeps more reliable bounds at the expense of the utilization level. Our study concludes that the measurement block T is a crucial parameter that

Table 8
Single-hop and multi-hop, all source models requesting multiple levels of service

Topology	Link number	Utilization		Delay (ms)			
		Classes 1 and 2	Reference	Class 1	Class 2	Reference [d_1]	Reference [d_2]
1-LINK	L_3	0.6038	0.66	1.6725	17.974	3	45
2-LINK	L_4	0.6683	0.72	0.7942	43.069	2	54
	L_5	0.7804	0.72	0.8236	17.617	2	41
4-LINK	L_6	0.4411	0.47	0.7823	38.109	1	36
	L_7	0.7476	0.70	0.7905	39.053	2	46
	L_8	0.7501	0.72	0.8012	40.917	2	49
	L_9	0.7555	0.75	0.7993	41.047	1	53

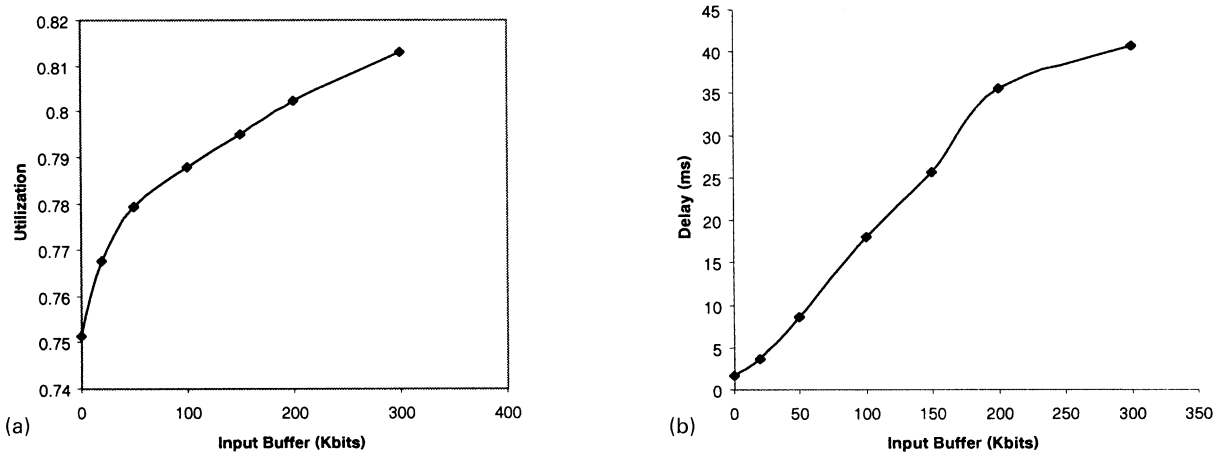

Fig. 6. (a) Link utilization as a function of input buffer. (b) Queuing delay as a function of input buffer.

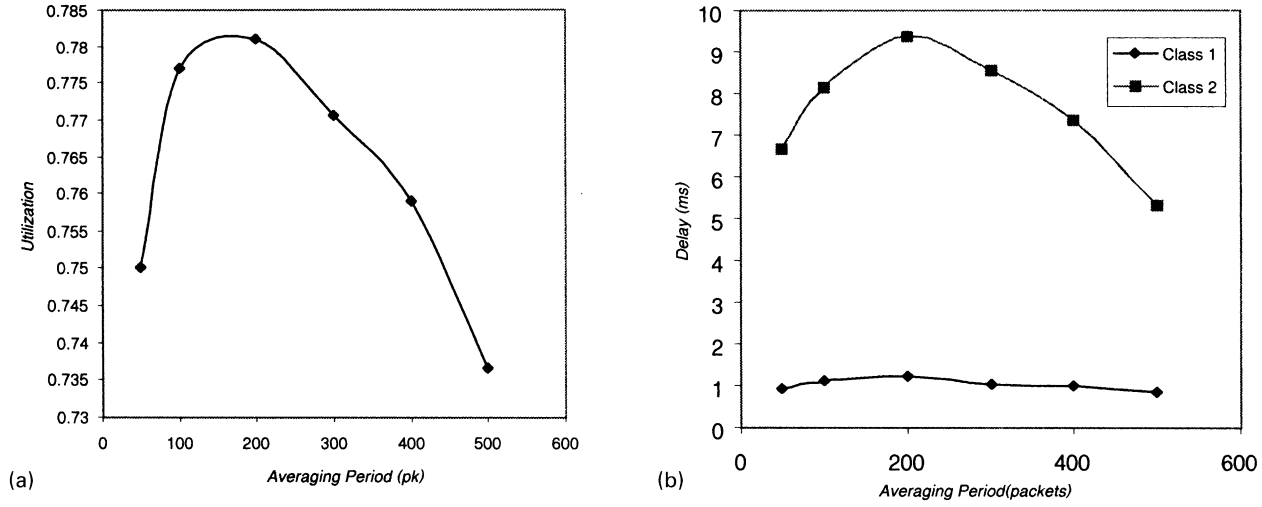

Fig. 7. (a) Link utilization as a function of averaging period. (b) Class 1 and Class 2 delay as a function of averaging period.

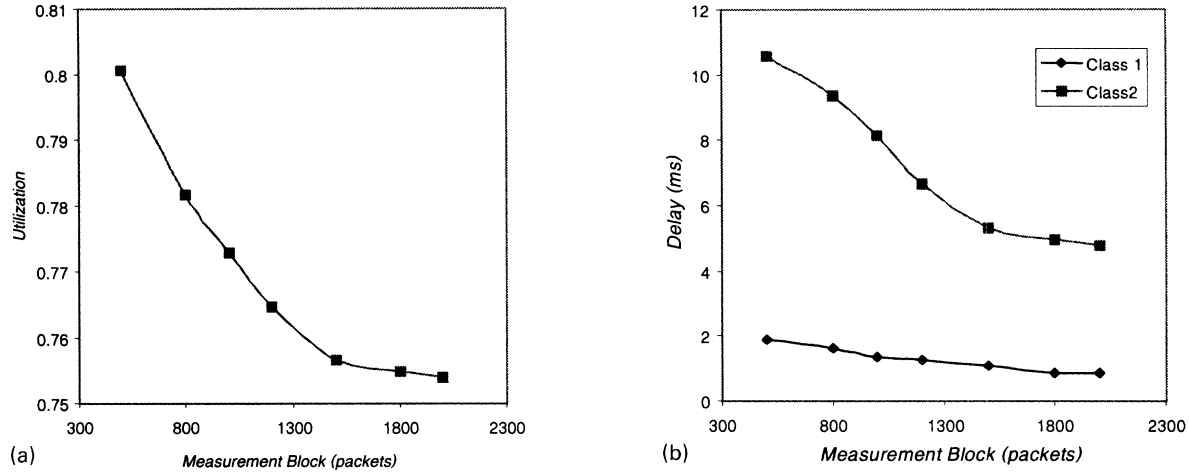

Fig. 8. (a) Utilization as a function of Measurement Block. (b) Class 1 and Class 2 delay as a function of measurement block.

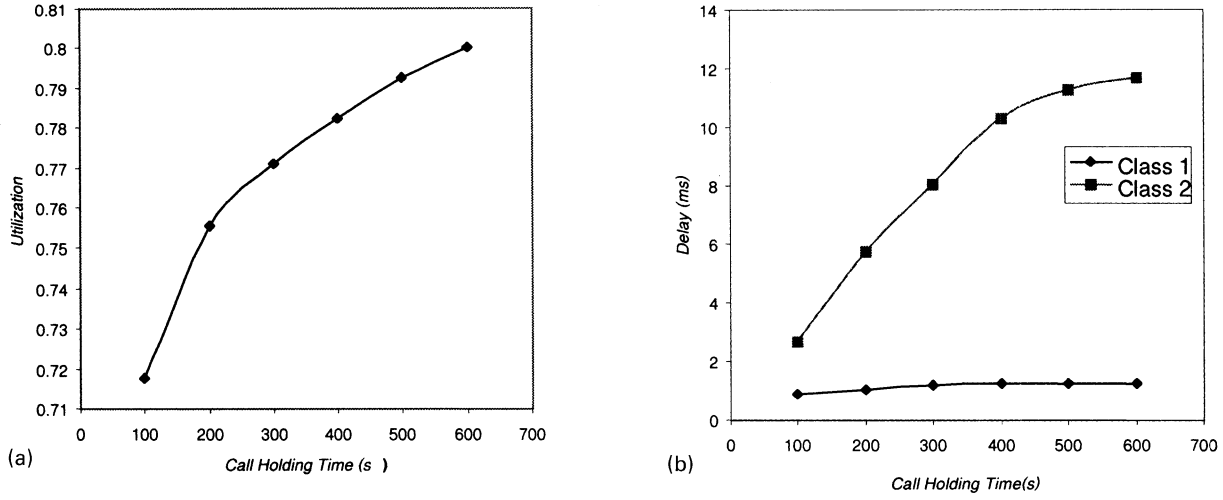

Fig. 9. (a) Utilization as a function of Call holding. (b) Class 1 and Class 2 delay as a function of CHT.

controls the tradeoff between the QoS reliability and the achievable resource utilization.

5.4. Effect of call holding time on link utilization and delay

The CHT is also commonly known as 'flow lifetime' or 'flow duration'. For measurement-based admission control, the expected CHT (i.e. flow lifetimes) of the participating flows is an important parameter that affects the QoS reliability supported by the admission control scheme. Flows with longer lifetime present a greater threat to the measurement-based admission control simply because the negative impact caused by a wrong decision will take a longer time to diminish. On the other hand, shorter average flow duration means more departure per T. If the number of departures is significant, a flow will see a much smaller number of flows during its lifetime. This is simply because the bit rate of a new flow is computed based on the peak bit rate and not the measured rate. Therefore, we should expect a lower utilization when the CHT is short. However, the lifetime of an application may not be known at admission time; even if it is known, it would still be very difficult to choose optimal system parameters because of the inherent diversity in the mixture of lifetimes of the existing flows inside the network.

Our observation on the impact of CHT on the utilization and QoS performance is based on simulations running on all six source models on the ONE-LINK topology. We adjusted the input buffer size Q for Class 1 and Class 2 traffic to 0 and 100 kbits, respectively. We use averaging period S of 100 packet transmission times and a measurement block T of 1000 packet transmission times. In Fig. 9(a), we show the average link utilization for different values of the average flow duration. We varied the average flow duration from 100 to 600 s. Fig. 9(a) shows that longer lasting flows allow higher link utilization while Fig. 9(b) shows that shorter flow durations result in lower Class 1 and Class 2 queuing delays.

In Ref. [4], the authors studied the relationship between the measurement block period T and the flow duration. They defined the ratio of measurement block period T and average lifetime (L) as *stability* (T/L = stability). A smaller T/L ratio means higher delay and higher link utilization. Thus, lowering the T/L ratio is one way to increase resource utilization.

6. Conclusions

In this paper, we have proposed a new adaptive resource re-negotiation measurement-based control scheme to handle interactive real-time applications. The admission decision is made based on the worst-case traffic of the new flow, which is modelled using a token bucket and the measured residual resources on the concerned link. Our admission algorithm tries to ensure that the QoS requirements of the new flow are met if the flow is admitted into the network and no other service commitments will be violated as a result of this new admission. One of the major contributions of our proposed admission control scheme is that our scheme offers the re-negotiation feature in the resource reservation process, which in turn allows us to achieve higher admission ratios and higher resource utilization. Our simulation studies have shown that the proposed measurement-based admission control can effectively control the incoming traffic to support reliable QoS with high resource utilization even for bursty long-range dependent traffic.

References

[1] S. Jamin, S.J. Shenker, P.B. Danzig, Comparison of measurement-based admission control algorithm for controlled-load service, IEEE INFOCOM '97 (1997) 973–980.
[2] D. Tse, M. Grossblauser, Measurement-based call admission control: analysis and simulation, IEEE INFOCOM '97 (1997) 981–989.
[3] J.Y. Qiu, E.W. Knightly, QoS control via robust envelope-based

MBAC, IEEE Workshop on Quality of Service IWQoS '98 (1998) 62–64.

[4] S. Jamin, P.B. Danzig, S.J. Shenker, L. Zhang, A measurement-based admission control algorithm for integrated services packet networks, IEEE/ACM Transactions in Networking 5 (1) (1997) 56–70.

[5] S. Shenker, J. Wroclawski, General characterization parameters for integrated service network elements, RFC 2215, September 1997.

[6] T.-Y. Tan, Quality of service control for interactive real-time applications, MEng dissertation, Nanyang Technological University, Singapore, 2000.

[7] A.K. Parekh, R.G. Gallenger, A generalized processor sharing approach to flow control in integrated services networks: the single-node case, IEEE/ACM Transactions in Networking 1 (3) (1993) 344–357.

[8] V. Paxson, S. Floyd, Wide-area traffic: the failure of Poisson modeling, ACM SIGCOMM '94 (1994) 257–268.

[9] W. Willinger, M.S. Taqqu, R. Sherman, D.V. Wilson, Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level, ACM SIGCOMM '95 (1994) 110–113.

[10] J. Haslett, A.E. Raftery, Space–time modeling with long-memory dependence: assessing Ireland's wind power resource, Application Statistic 38 (1) (1989) 1–50.

[11] A. Adas, A. Mukherjee, On resource management and QoS guarantees for long range dependent traffic, IEEE INFOCOM '95 (1995) 779–787.

[12] G.E.P. Box, G.M. Jenkins, Time Series Analysis: Forecasting and Control, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[13] S. Lee, J. Song, A measurement-based admission control algorithm using variable-sized window in ATM networks, IEEE Information, Communications and Signal Processing ICICS '97 (1997) 378–384.