A Simulation Study of an Adaptive Distributed Algorithm for Max-Min Fair Rate Control of ABR Sessions *

Santosh P. Abraham and Anurag Kumar
ECE Dept. Indian Institute of Science Bangalore, India
e-mail: aspaul, anurag@ece.iisc.ernet.in

Abstract

The successful deployment of the ABR class of traffic in ATM networks requires distributed rate allocation algorithms. These algorithms must have simple update mechanisms and be adaptive to changing available capacity, while ensuring fair allocation of the available capacity to contending sessions. We have developed a class of distributed algorithms based on the stochastic approximation approach; the theory for these has been published elsewhere by us. The switch algorithm is completely distributed, simple to implement and requires no per-VC information. In this paper we provide an overview of the algorithm, describe how avaliable capacity is estimated and how large changes in available capacity are adapted to. In order to ensure low buffers we use an available capacity estimator which incorporates queue length considerations. We use simulations to demonstrate the performance of the basic algorithm along with its various improvements.

1 Introduction

The ABR class in ATM networks is a best effort service mainly intended for the support of sessions with vague throughput and delay requirements. The main motivation for such a class comes from the desire to economically support data traffic. A reactive rate control approach with max-min fairness between sessions has been chosen for allocating the available bandwidth to ABR sessions. In order to guarantee a minimum throughput for some ABR sessions, should they demand it, a Minimum Cell Rate (MCR) facility has been introduced. A session if admitted at a certain MCR is guaranteed at least that throughput for the duration of the session. Thus MCR units of bandwidth must be reserved for such an ABR session on all the links it spans. An admission control will be essential in order to guarantee the MCR.

The design of rate allocation algorithms is complicated by the fact that the available capacity for ABR traffic (left over after servicing guaranteed QoS classes i.e., CBR

*Report based on research supported by a grant under an IISc-Nortel Memorandum of Understanding.

and VBR) is varying. These variations can be characterised as occurring over two time scales. Rapid variations occur due to intrinsic rate variations of VBR traffic, and slower variations due to the entry and exit of CBR and VBR sessions.

The successful deployment of the ABR class requires algorithms with the following features.

1. The algorithm must be distributed with simple update expressions and minimal communication of control information between various computing nodes.
2. Should be robust to short time scale variations and adapt to larger time scale variations.
3. Should ensure low losses and high utilisation of available capacity.
4. Session rates obtained should converge to their max-min fair values.

In this paper we preset a switch algorithm that has the above desirable properties by incorporating two features. We then demonstrate the performance of the algorithm via simulation. The algorithm is based on stochastic approximation, making it robust to rapid variations of available capacity, and has mechanisms for adapting to slower variations. The use of stochastic approximation for ABR rate control was proposed in [3]. The notion of max-min fairness used incorporates MCR requirements and is a natural generalisation of max-min fair idea without MCR requirements [2]. The other feature is the use of an available capacity estimate that incorporates cell loss constraints. Such an estimate is based on the theory of large deviations and can be viewed as the dual of the equivalent bandwidth notion for sources with stochastic rates.

This paper is organised as follows. In Section 2 we present a brief review of the theory max-min fair allocation with non-zero MCRs and motivate the use of stochastic approximation algorithms for computing a fair share. In Section 3 we present techniques for making the stochastic approximation based algorithm adaptive to local and remote changes in available capacity. In Section 4 we outline the algorithm to be implemented at the switches. In Section 5 we present the available capacity estimation technique that incorporates queue constraints and give a recursive estimation algorithm. Simulation results are presented and discussed in Section 6. We conclude in Section 7.

2 Max-Min Fairness with MCRs and Stochastic Approximation Algorithms for Rate Allocation - A Brief Review

In this section we briefly review a notion of Max-Min Fairness with MCRs. We then briefly review the use of stochastic approximation for computing the max-min fair share. A detailed study of the above two topics is available in [2] and [3].

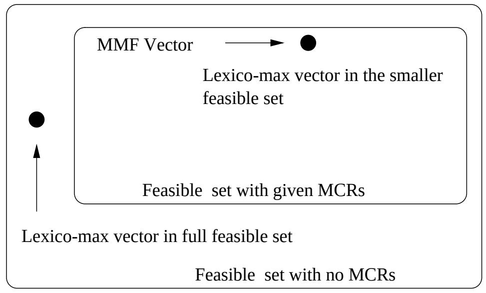


Figure 1: Relationship between the feasible rate vectors with and without MCR requirements.

In the conventional max-min fair allocation scheme, we search for a rate vector that is the *lexicographic maximum* of all the feasible rate vectors. The lexicographic ordering is obtained as follows: we consider the set of feasible vectors and deduce a new set of vectors by simply reordering each vector in the feasible set in ascending order. If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are two vectors in the new set (i.e., note that $x_1 \leq x_2 \ldots \leq x_n$, similarly for y), then x is lexicographically larger than y, if $x_l > y_1$ or if $x_1 = y_1$, then $x_2 > y_2$ and so on.

The lexicographic maximum rate vector has the following property; every session has at least one link which is fully utilised and the sessions rate is the highest among all sessions in that link. Such a link is called a *bottleneck* link. This concept has been generalised to the MCR case in [2]. In the generalised case we seek the lexicographic maximum in the reduced set of feasible vectors obtained by imposing MCR conditions on the feasible set without MCRs. This idea is illustrated in Figure 1. It has been shown in [2] that the notion of a bottleneck link also carries over to this framework.

Another important observation of the max-min fair vector is as follows. Consider the case where for the max-min fair vector every link is a bottle neck for some session. Let $\mathcal{L}$ denote the set of links, associate with each link $l \in \mathcal{L}$, a link control parameter η_l. The rate of each session s is computed as follows

$$r_s = \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j)$$

where $\mathcal{L}_s$ is the set of links spanned by session s and μ_s is the MCR of session s. If every link is a bottle neck for some session, then the vector of link control parameters $\eta = (\eta_l, l \in \mathcal{L})$ also solves

$$C_l - \sum_{s \in \mathcal{S}_l} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j) = C_l - \sum_{s \in \mathcal{S}_l} r_s = 0 \quad \forall l \in \mathcal{L}$$

Hence computing the max-min rate of a session is equivalent to solving for the root of a certain vector equation [2].

Recall that the link capacity available to ABR is stochastic, hence the above problem becomes one of finding the root of a vector valued function whose noisy values can be observed, and the root computation must be distributed. We proposed the use of *Stochastic Approximation* [6] type algorithms for this problem, and proved the convergence of a *distributed synchronous* stochastic approximation algorithm in [3]. The idea in stochastic approximation is to weight the increment at each update by a sequence of decreasing gains. The rate of the decrease of gains is such that convergence is ensured while removing the effect of noise. The link parameter update we use is of the form

$$\eta_l(k+1) = \eta_l(k) + \alpha(k)\left(C_l + \omega(k) - \sum_{s\in\mathcal{S}_l} \max(\mu_s, \min_{j\in\mathcal{L}_s} \eta_j(k))\right)$$

$C_l + \omega(k)$ denotes the stochastic available capacity and $\alpha(k)$ is the sequence of decreasing gains which has the following properties.

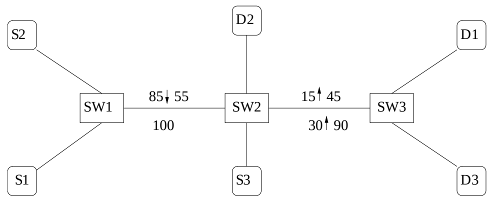
$$\sum_{k=0}^{\infty} \alpha(k) = \infty \qquad\qquad \sum_{k=0}^{\infty} \alpha(k)^2 < \infty$$

3 Adapting to Changes in Available Capacity

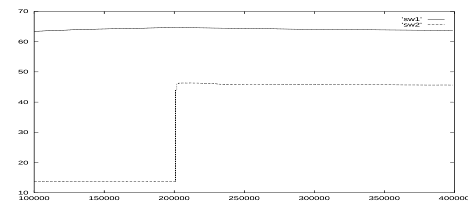
The stochastic approximation algorithm effectively handles the short time scale random changes in available capacity. However, the reduced value of the gain of the stochastic approximation algorithm causes poor response to changes that take place over long time scales. Hence, in order for the algorithms to be effective, we must incorporate into them mechanisms that detect such changes in available capacity and reset the stochastic approximation gain.

We assume that a switch is aware of the entry and exit of CBR/VBR sessions through it. When such an entry/exit occurs, the gain is reset. Thus changes in available capacity local to a switch can be handled effectively. However, note that the max-min solution is a global solution across the network. Hence a change in capacity at a given link can cause the link parameter at another link to change drastically. Thus a switch must be able to reset its gain in response to a change that occurs at a remote link.

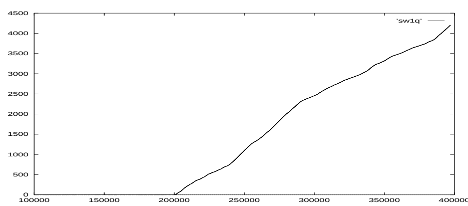
We use two examples to illustrate the effects of remote capacity changes, and outline the procedures used for detection. In the network shown in Figure 2, the link capacity at link 2 increases, this increase is assumed to be known as SW2, but not at SW1. Hence the link control at link 2 increases, causing the rate of session 1 to increase. Since SW1 does not know of the rate change at link 2 and the stochastic gain has decreased to a low value, it does not lower its control parameter quickly enough. Hence the rate of session 2 does not reduce to its max-min value and the total input rate to SW1 is greater than the output rate causing queue lengths at SW1 grow. We thus conclude that queue thresholds can be used to signal that a remote change in available capacity. In Figure 3 we show the link control parameters and the queue lengths for the same network with a queue threshold being used to signal that the stochastic approximation gain must be increased.



Simulated Network: Link capacities are indicated below the links, and the corresponding link control values are indicated above the links

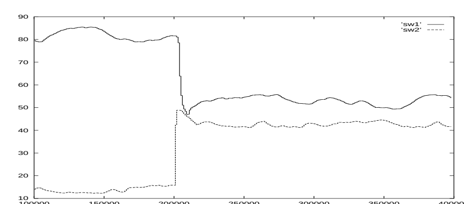


Link control parameters at SW1 and SW2

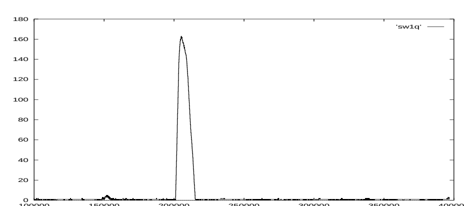


Queue length at SW1

Figure 2: Example illustrating the effect of an increase in capacity at a remote link

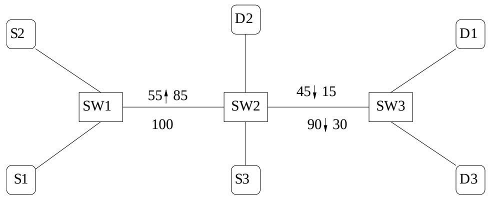


Link control parameters at SW1 and SW2

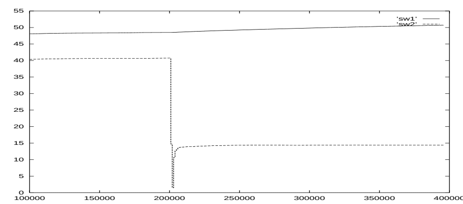


Queue length at SW1

Figure 3: Link control parameters and queue lengths when a queue threshold is used to reset the stochastic approximation gain



Simulated Network: Link capacities are indicated below the links, and the corresponding link control values are indicated above the links



Link control parameters at SW1 and SW2

Figure 4: Example illustrating the effect of an decrease in capacity at a remote link

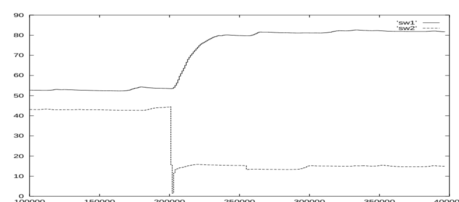


Figure 5: Link control parameter at SW1 and SW2 with low link utilisation detection incorporated

In the network shown in Figure 4, the available capacity at link 2 falls. This information is available at SW2 and it resets its stochastic approximation gain and computes the new link control parameter which is lower. The rate of session 1 drops freeing capacity for session 2 to use. However, since the drop in rate value at link 2 is not known at SW1, and the stochastic gain is low, the link control does not increase quickly enough to ensure utilisation of the available capacity. We thus incorporate a low utilisation detection mechanism at the switches. When low utilisation is detected, the stochastic gain is reset allowing the link parameter to increase ensuring higher utilisation of the freed capacity. Note that this mechanism will also detect the case when a session does not utilise its allocated rate and hence adds the desired "use-it-or-lose-it" feature to the algorithm. The low utilisation detection algorithm is given below. The algorithm updates a parameter β, based on the link utilisation using an "exponential forgetting" principle. Only when a sufficient number of low utilisation epochs have been detected, does the stochastic approximation gain increase.

Algorithm 3.1 *At each update epoch do:*

1. Compute the present link utilisation:

$$\rho = \frac{\text{Total time that the queue had at least one customer}}{\text{Update Interval}}$$

2. if($\rho >$ Utilisation Threshold)

   $\beta = \rho$

   else $\beta = h\beta + (1-h)\rho$

3. if($\beta <$ Utilisation Threshold)

   Increase the gain of the stochastic approximation algorithm.

In Figure 5, we show the operation of the algorithm with the low utilisation detection incorporated. Note that the link control parameter for link 1 increases shortly after the capacity of link 2 drops thus ensuring better utilisation of the rate freed by session 1.

4 The Switch Algorithm

In this section we present a sketch of the switch algorithm that incorporates the above features.

The switches periodically update the link control parameter for each outgoing link. At each update epoch, for each link l do the following (the following notation is used:-
η_l = link control parameter,
α_l^0 = initial value of the stochastic approximation gain,
f_l = estimate of the input rate,
C_l = estimate of the available capacity,
α_l = stochastic approximation gain used for the update,
C_l^{max} = maximum permissible value for η_l) .

1. Estimate the input rate f_l,

$$f_l = \frac{\text{number of cells that arrived in the previous inter-update interval}}{\text{length of inter-update interval}}$$

2. Estimate the available capacity C_l (methods to be described later).

3. Reduce the stochastic approximation gain α_l.

4. If the mean of the available capacity has changed increase the gain α_l.

5. If the Queue Length is greater than the preset queue threshold, increase the gain α_l.

6. If the low link utilisation has been detected, increase the gain α_l.

7. Compute the new value of the link control parameter η_l by (we use the notation $[x]_a^b = \max(a, \min(x, b)))$

$$\eta_l \leftarrow [\eta_l + \alpha_l(C_l - f_l)]_0^{C_l^{max}}$$

Note that the above algorithm is completely distributed. All the information used is available from local measurements at the switch. No explicit communication of control information between computing nodes is required. Also note that no per-VC information is maintained. We have not specified the way C_l is estimated. In the simulations presented we use this algorithm with two available capacity estimation strategies. Our emphasis shall be on an estimation technique that incorporates queue behaviour. This shall be the topic of the next section.

5 Estimating the Equivalent Available Link Capacity

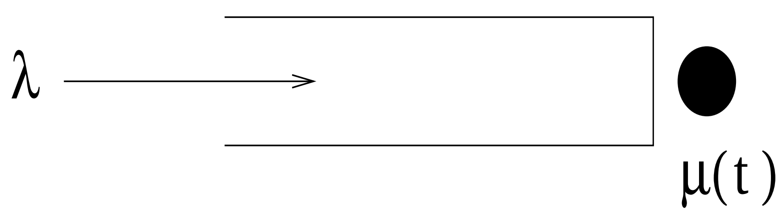


Figure 6:

Consider a queue with a constant input rate and a service rate that is a stochastic process (Figure 6). The equivalent available capacity of the queue server is that constant input rate which when fed into the queue with the stochastic service rate yields the desired overflow probability. The equivalent available capacity can be viewed as the dual of the equivalent bandwidth for sources whose rate is a stochastic process. We use the approach outlined in [5] to derive a formula for the equivalent available capacity.

Assume that we have slotted service process. Let D_m denote the number of cells that can be served the m^{th} slot. Assume that the following exists of $\theta \in \Re^+$

$$\Gamma(\theta) = \lim_{n\to\infty} \frac{1}{n} \log E \exp\left\{-\theta \sum_{m=0}^{n-1} D_m\right\}$$

It can be shown that if the input rate (A cells/slot) has the following property

$$A < \frac{\Gamma(\theta)}{-\theta}$$

then the Queue length (Q) has the following property

$$\lim_{B\to\infty} \frac{1}{B} \log P(Q > B) \leq -\theta$$

Given a buffer size B and a desired loss probability ϵ, we compute the equivalent available capacity as follows. We first compute $\theta as follows$,

$$\theta = \frac{\log_e(\epsilon)}{B}$$

The available capacity A is then given by

$$A = \frac{\Gamma(\theta)}{-\theta}$$

5.1 An Estimation Procedure

In this section we shall outline an on-line estimation procedure for computing the equivalent available capacity using measurements of the service process. We assume that for fairly large M

$$-\frac{\Gamma(\theta)}{\theta} = \lim_{n\to\infty} -\frac{1}{\theta n} \log E \exp\left\{-\theta \sum_{m=0}^{n-1} D_m\right\} \simeq -\frac{1}{\theta M} \log E \exp\left\{-\theta \sum_{m=0}^{M-1} D_m\right\}$$

Assume that the total number of customers that can be served in M non-overlapping of slots is approximately independent. Let D_m^j denote the number of customers served in the m^{th} slot of the j^{th} block. We use the following estimate

$$-\frac{1}{\theta M} \log E \exp\left\{-\theta \sum_{m=0}^{M-1} D_m\right\} \simeq -\frac{1}{\theta M} \log\left(\frac{1}{N} \sum_{j=0}^{N-1} \exp\left\{-\theta \sum_{m=0}^{M-1} D_m^j\right\}\right)$$

A simple recursive formula for the above estimate can be derived as follows. Let

$$D(j) = \frac{1}{M} \sum_{m=0}^{M-1} D_m^j$$

D(j) is the average number of customers served in the j^{th} block of M slots. Let $C(N)$ denote the estimate obtained at the Nth iteration, i.e.,

$$C(N) = -\frac{1}{\theta M} \log\left(\frac{1}{N} \sum_{j=0}^{N-1} \exp\left\{-\theta M D(j)\right\}\right) \tag{1}$$

Now consider $C(N+1)$

$$C(N+1) \;=\; -\frac{1}{\theta M} \log\left(\frac{1}{N+1} \sum_{j=0}^{N} \exp\left\{-\theta M D(j)\right\}\right)$$

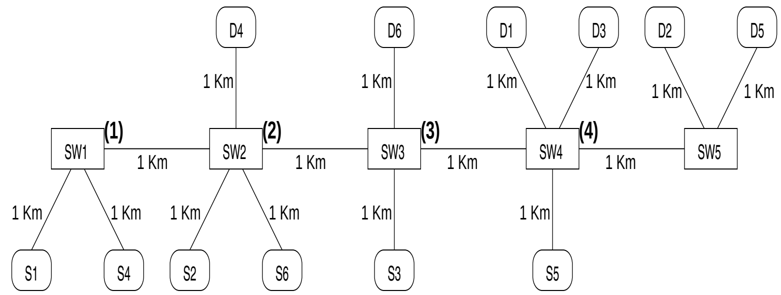


Figure 7: Network topology; showing switches, links, sessions and link lengths

$$
\begin{aligned}
&= -\frac{1}{\theta M}\log\left(\frac{1}{N+1}\exp\{-\theta M D(N)\} + \frac{N}{N+1}\left(\frac{1}{N}\sum_{j=0}^{N-1}\exp\{-\theta M D(j)\}\right)\right)\\
&= -\frac{1}{\theta M}\log\left(\frac{1}{N+1}\exp\{-\theta M D(N)\} + \frac{N}{N+1}\left(\exp\{-\theta M C(N)\}\right)\right)\\
&= C(N) - \frac{1}{\theta M}\log\left(\frac{1}{N+1}\exp\{-\theta M(D(N)-C(N)\} + \frac{N}{N+1})\right)\\
&= C(N) - \frac{1}{\theta M}\log\left(1-\frac{1}{N+1}\right) - \frac{1}{\theta M}\log\left(1+\frac{1}{N}\exp\{-\theta M(D(N)-C(N)\})\right)
\end{aligned}
$$

In the above expression the "log" terms can be replaced by suitable approximations making the estimator robust to floating point limitations of the computing nodes.

There is another way of viewing the estimate $C(N)$ given in Equation 1 that is intuitively appealing. Note that

$$\min_{j=0,\ldots,N-1} D(j) \;\leq\; C(N) \;\leq\; \frac{1}{N}\sum_{j=0} N-1 D(j) \quad \text{and}$$

$$\lim_{\theta\to\infty} C(N) = \lim_{\theta\to\infty} -\frac{1}{\theta M}\log\left(\frac{1}{N}\sum_{j=0}^{N-1}\exp\{-\theta M D(j)\}\right) = \min_{j=0,\ldots,N-1} D(j)$$

Hence C(N) is an estimate of the available capacity that is weighted towards intervals with lower service capacities. The "weighting" is controlled by the parameter θ

6 Simulation Results

We now present the results for the example LAN shown in Figure 7.

Model for Available Link Capacities: The link capacities are assumed to be random with mean values over a long time scale. The instantaneous available capacity is a truncated random walk like process obtained as follows. If C is the mean and $C(k)$ is the capacity at the k^{th} epoch then C(k+1) is obtained as follows.

$$C(k+1) = [C(k) + X(k)]_{C-5}^{C+5}$$

Where X(k) is i.i.d, and uniformly distributed between $[-1, 1]$. Note that the random walk used causes correlated variations around the mean available capacity. The change epochs of the mean value of a particular link are assumed to be known at the switch computing its link control parameter. To facilitate comparison, the link capacity sequences for all the experiments are point wise identical.

Switch Parameter Settings : The following parameters were used in the simulations.

- The interval between updates at each switch is 1ms.
- The initial value of stochastic approximation gain is 1, and this is the value that the gain is reset to.
- The gain at each switch was iteratively decreased by increasing its reciprocal by 0.1, i.e.,

$$\alpha_l(k+1) = \frac{1}{1/\alpha_l(k) + 0.1} = \frac{10\alpha_l(k)}{10 + \alpha_l(k)} \tag{2}$$

- The queue threshold was chosen to be 200 cells

Session MCRs: Session 1 and and Session 2 have MCRs' of $10Mb/s$ and $20Mb/s$ respectively. The rest of the sessions have zero MCRs.

We present two sets of results, whose simulation procedures differ in the way that the capacity available to ABR traffic is estimated. For each simulation we plot the link control parameter and the queue lengths observed at the switches. The simulated time in each experiment was 8 seconds. We assume that each switch knows when the mean of its available capacity changes, at any such epoch, the link capacity estimation is restarted.

6.1 Experiment 2: Using a long run Estimate of Available Link Capacity

In this section we present the plots of the link parameter and queue length for each link considered, when the available capacity is estimated using measurements over a long interval. The estimation procedure is outlined below.

Estimation of Available Capacity: In each measurement update interval we count the number of cells that have been served on a link. We also measure the time for which the output queue for the link had at least one cell. The estimate of the available rate is obtained by dividing the number of cells by the busy time of the queue. At each update epoch we take an average of these estimates obtained in several of the previous update intervals. The average is taken over all update intervals from the most recent mean capacity change epoch to the present update epoch. The so obtained average is multiplied by a factor of 0.95 and used as C_l in the algorithm.

Description of the Plots: In the first column of Figure 8 we plot the following for each link.

1. The instantaneous Link Capacity.
2. The exact link parameter obtained from a max-min computation if the link capacities were at 0.95 times the mean value.
3. The link parameters obtained by the algorithm.

We observe that the computed link control parameters track the ideal link parameters computed from exact knowledge of 0.95 of mean available capacity.

In the second column of Figure 8 we plot the queue length process obtained at the queue of each of the links.

6.2 Experiment 3: Using an Estimate of the Equivalent Available Link Capacity

In this section we present the plots of the link parameter and queue length for each link considered, when an estimate of the Equivalent Available Link Capacity is used. The estimation procedure is outlined below.

Estimation of Available Capacity: In each measurement update interval we count the number of cells that have been served at a link. We also measure the time for which the output queue for the link had at least one cell. The estimate of the available rate in the update interval is obtained by dividing the number of cells by the busy time of the queue. This estimate of available capacity is used as "$D(N)$" in the recursive estimation procedure outline in Section 5.1.

We have used a large value of θ for two reasons. The main reason is that the computed values of $D(N)$ may not be independent and the measurement update interval may not be long enough as required by the estimation procedure. Secondly, the derivation of the equivalent available capacity assumed that the input rate into the switch was constant. However, since the actual input process into the switch is the output of a link with stochastic capacity, we expect variations in the input rate. It is hoped that a large choice of θ will offset the effect of these variations.

Description of the Plots: In the first column of Figure 9 we plot the following for each link.

1. The instantaneous Link Capacity.
2. The exact link parameter obtained from a max-min computation if the link capacities were at 0.95 times the mean value.
3. The exact link parameter obtained from a max-min computation if the link capacities were at the mean value.
4. The link parameters obtained by the algorithm.

In the second column of Figure 9 we plot the queue length process obtained at the queue of each of the links.

6.3 Discussion of the Results

On comparing the results of Experiments 1 and 2 we observe the following:

- From the plots in Figure 9, note that the link parameter based on equivalent capacity is always less than that obtained from the mean and could be more or less than that obtained from 0.95 times the mean.
- In Experiments 1, we observe large queue lengths for links 1 and 4. However the queue lengths observed in Experiment 3, for these links, are much smaller. This shows that the choice of a fixed factor for degrading the measurements of available capacity without considering the variability in the available capacity process could yield poor behavior in terms of the queue lengths. Also note that better capacity utilisation is obtained with the with equivalent capacity estimate for links 2 and 3.
- With the use of an equivalent capacity, the utilisation of the link can be effectively governed by queue length considerations. The use of equivalent capacity provides an effective compromise between the opposing objectives of efficient link utilisation and low buffer occupancy.

7 Conclusion

In this report we have presented simulations of rate allocation algorithms for a LAN type environment that yield max-min allocation while tracking the variations in available capacity. These algorithms incur no communication over heads and are simple to implement. They use estimates of available capacity calculated from measurements.

We have presented simulation results using a long run average of measured link rate. and and estimate of the equivalent capacity. also a presented an approach based on the Our simulation results indicate that rate feedbacks based on computations made using the equivalent capacity yield better queue length performance and more efficient utilisation of the available capacity.

Future work we wish to pursue includes:

- Adapting the algorithms presented here to a WAN environment.
- A more detailed study of the equivalent capacity approach in terms of choice of the parameter θ (which depends on loss probabilities and buffer sizes).

- An investigation into techniques for obtaining better estimates of the equivalent capacity.
- An investigation into implications of equivalent capacities for admission control of ABR sessions.

References

[1] The ATM Forum Traffic Management Specification Version 4.0, April 1996

[2] Santosh P. Abraham and Anurag Kumar, "Max-Min Fair Rate Control of ABR Connections with Nonzero MCRs", Proceedings IEEE *Globecom'97*, Phoenix, Nov. 1997, pp 498-502.

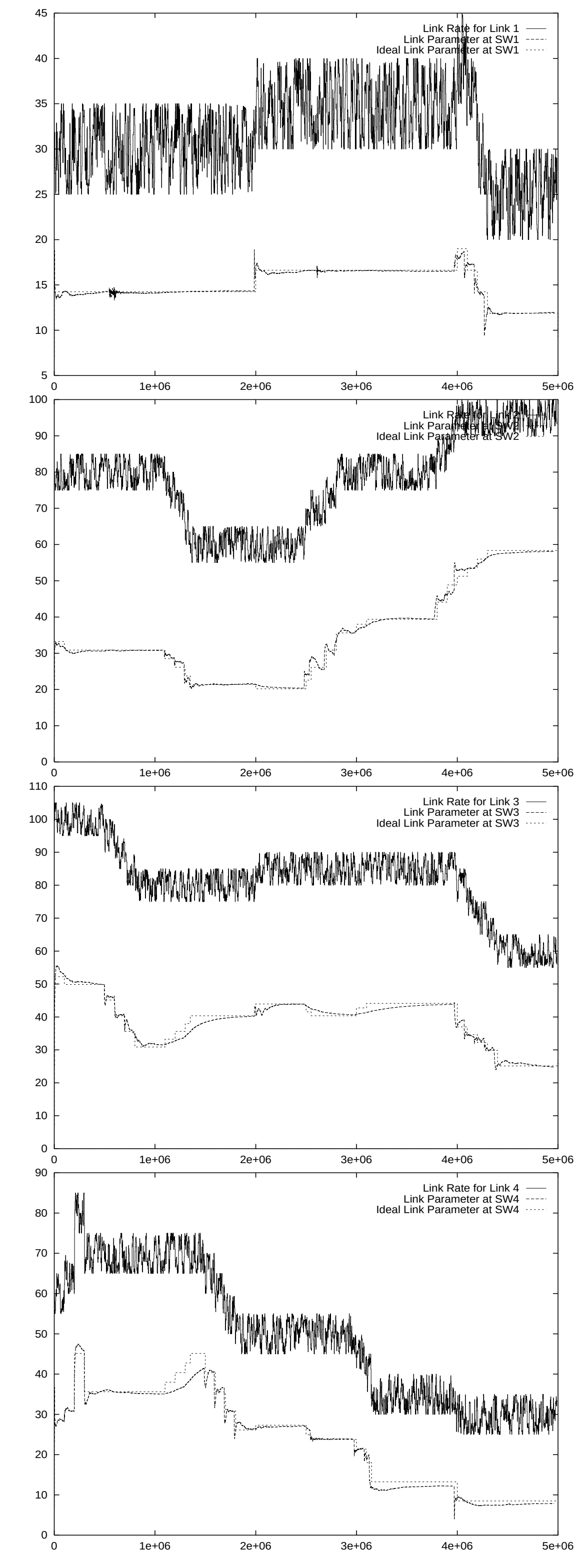
[3] Santosh P. Abraham and Anurag Kumar, "A Stochastic Approximation Approach for Max-Min Fair Adaptive Rate Control of ABR Sessions with MCRs", *IEEE Infocom'98*, San Francisco, Mar-Apr, 1998.

[4] D. Bertsekas, R. Gallager. *Data networks,* Prentice-Hall Inc., 1992.

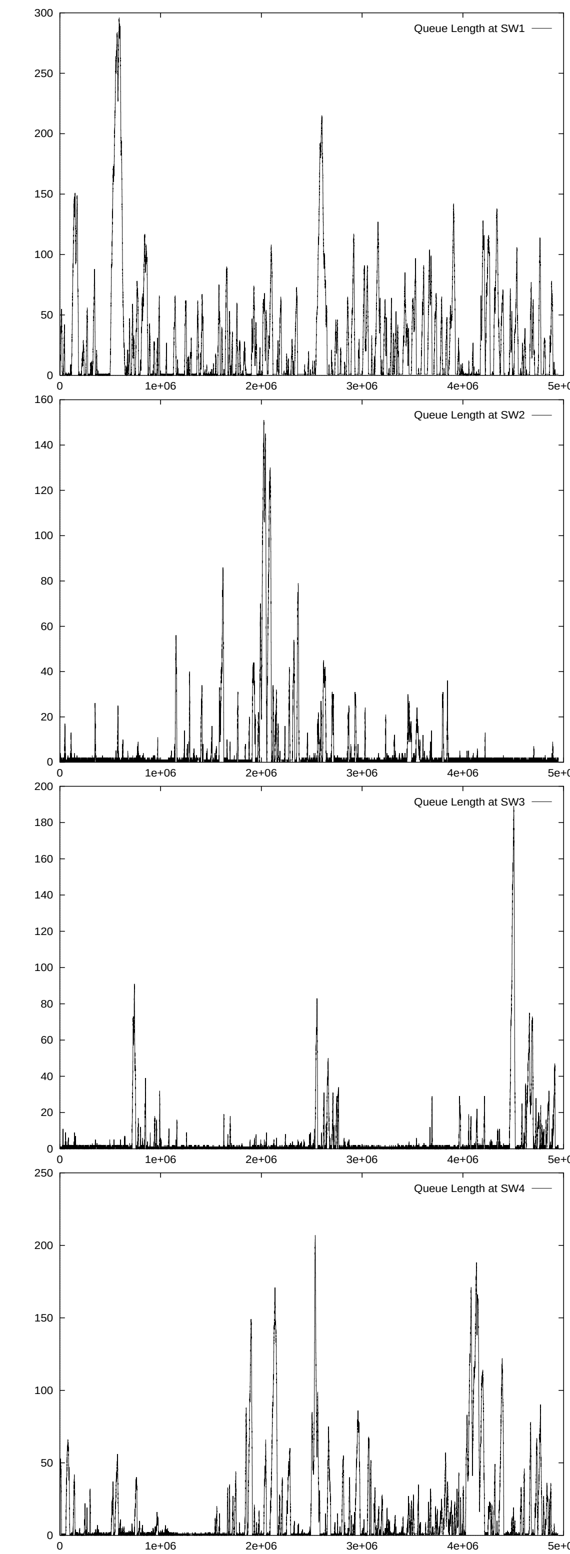
[5] George Kesidis, *ATM Network Performance*, Kluwer Academic Publishers, 1996.

[6] H.J. Kushner, D.S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems Springer-Verlag* 1978, pp 38-43.

[7] G. de Veciana, J. Walrand, "Effective Bandwidths: Call Admission, Traffic Policing and Filtering for ATM Networks", Memorandum UCB/ERL M93/47.

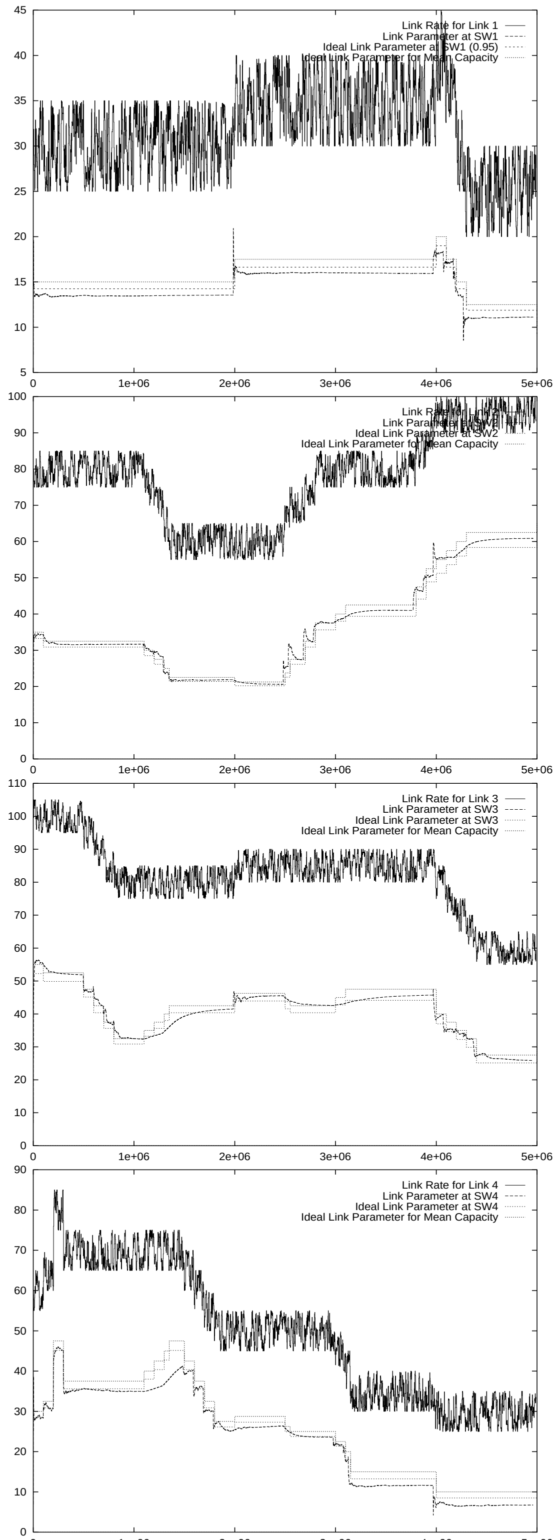


Experiment 1: Plots indicate how the link parameters at links 1,2,3 and 4 track the ideal link parameter value. Available capacity is estimated using a long run average.

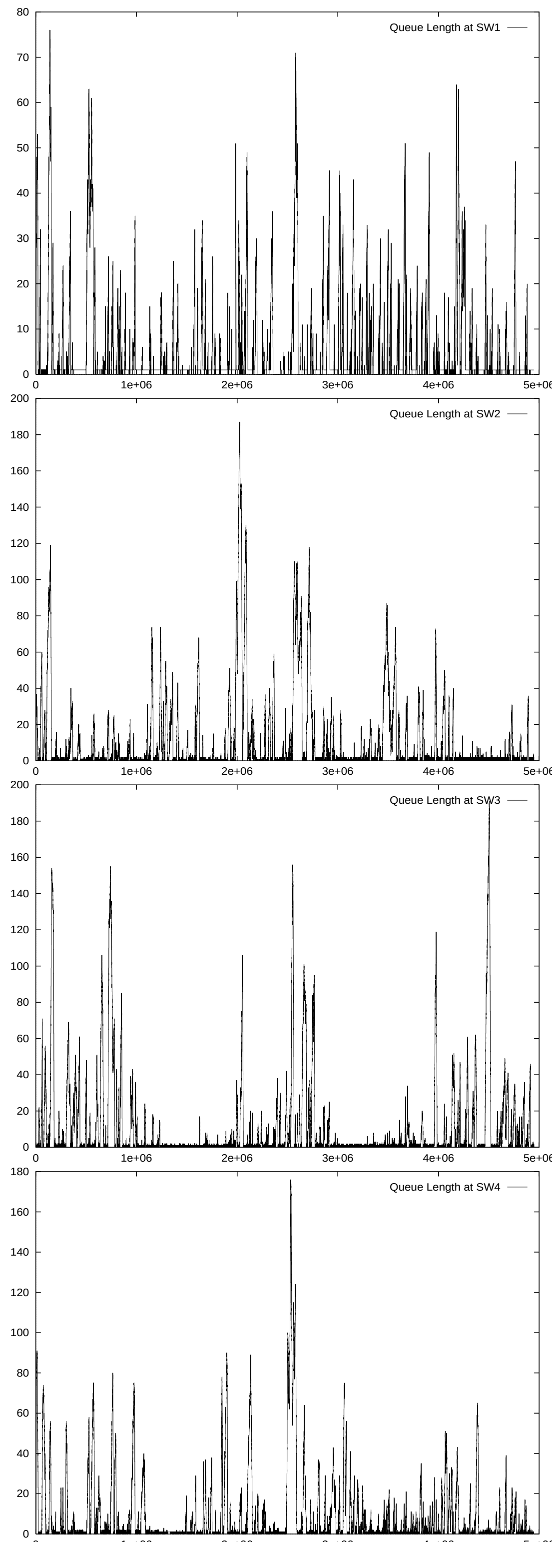


Experiment 1: Queue lengths at queues for links 1,2,3 and 4.
Available capacity is estimated using a long run average.

Figure 8:



Experiment 2: Plots of the link parameters at links 1,2,3 and 4 using an equivalent capacity estimate. Also shown are the link parameters obtained with capacity at the mean value and 0.95 times the mean value



Experiment 2: Queue lengths at queues for links 1,2,3 and 4. Rate computations were made based on the estimated Equivalent Capacity.

Figure 9: