A Stochastic Approximation Approach for Max-Min Fair Adaptive Rate Control of ABR Sessions with MCRs

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

Abstract

The ABR sessions in an ATM network share the bandwidth left over after guaranteeing service to CBR and VBR traffic. Hence the bandwidth available to ABR sessions is randomly varying. This bandwidth must be shared by the sessions in a max-min fair fashion.

Our point of departure in this paper is to formulate the problem of determining the max-min fair session rates as the problem of finding the root of a certain nonlinear vector equation; the same formulation also arises with our notion of max-min fairness with positive MCRs. This formulation allows us to use a stochastic approximation algorithm for online distributed computation of the max-min fair rates. We use the well known ordinary differential equation technique to prove convergence of the algorithm in the synchronous update case. We provide simulation results using the NIST simulator to show that the algorithm is able to track the max-min fair rates for slowly varying random available link bandwidths.

I. Introduction

A reactive rate control approach, with rate fairness between sessions has been chosen for allocating the available link rates (left over after allocation to the CBR and VBR classes) to ABR sessions. In order to guarantee a minimum throughput for some ABR sessions, should they demand it, the notion of a Minimum Cell Rate (MCR) has been introduced. Thus MCR units of bandwidth must be reserved for such a session on all the links that the session uses. An admission control will be essential in order guarantee the MCRs.

The notion of fairness proposed by the ATM forum is max-min fairness (MMF). This notion of fairness was discussed in earlier literature in the context of speech networks with variable rate coding; see [5] for a textbook treatment. A few ad-hoc proposals for extending this notion of fairness to the non-zero MCR case have been proposed by the ATM Forum [1]. These include MCR plus max-min share, and allocation proportional to the MCR.

The control algorithms for feedback control of ABR traffic will be implemented in the switches. The switches compute a suitable rate and communicate with the sources using RM (Resource Management) cells. Hence only distributed algorithms will be viable for practical implementation. The specifics of distributed feedback control algorithms for ABR traffic have not been standardised. This is a topic of current research and it is most likely that this is an area in which switch vendors will seek to differentiate their products.

Binary feedback based schemes [14] have been discussed for ABR traffic due to their simplicity in implementation. However recent research on ABR rate control focuses on explicit rate feedback algorithms. Charny [6] and Tsang [13] have proposed distributed versions of a centralised max-min allocation algorithm. Since these algorithms emulate a centralised algorithm, they need to track session bottlenecks. Distributed algorithms for max-min allocation which do not require tracking of session bottlenecks were proposed in the context of speech networks with variable rate coding in [8] and [12]. These algorithms are of a successive approximation type and are simple to implement. A recent distributed algorithm achieving max-min allocation without needing to track session bottlenecks was proposed by Fulton et al. in [7]. All these algorithms require that the available capacity remain fixed for the algorithm to converge. If a fixed available capacity is not available, then the available capacity sequence is averaged (filtered), and the average value is used in the computation of the rates. These algorithms operate on the assumption of no MCR requirements or that the computed rate is added to the MCR.

Distributed algorithms have also been designed using a control theoretic approach by Benmohammed et al. [3]. Kolarov et al. have extended the ideas in [3] to obtain algorithms that track changes in available rate quickly [10]. However these algorithms require explicit knowledge of the round trip times of individual sessions.

In this paper we examine the problem of distributed max-min fair share rate allocation to ABR sessions in a network in which the available link capacities are not constant, but are randomly varying. The notion of max-min fairness that we use was developed in [2], and is a natural generalisation of the notion with MCR=0. It was shown in [2] that the max-min allocation can be expressed in terms of the solution of a nonlinear vector equation. In the present paper, we study the use of stochastic approximation for solving this vector equation when the available capacities are random. We analytically prove convergence of the distributed stochastic approximation algorithm for the synchronous situation. We then present simulation results for an example wide-area network; the simulations incorporate the details of the ABR rate control protocol; i.e., RM cells, incremental additive increase, immediate decrease, etc.

II. Max-Min Fair Allocation Theory - A brief review

A. The Model and Notation

A session comprises a source node, cells from which traverse a fixed sequence of links to reach the destination node. The network topology, the link capacities, the sessions and their routes are all given and static. Each source has an infinite backlog, and can transfer it to the network at any specified rate [1]. For the purpose of defining the max-min fair notion we assume that the link capacities are fixed. We now describe the notation used.

We first describe some generic notation. Throughout $\mid A \mid$ denotes the size of, the set A, $A\backslash B$ denotes $A \cap B^c$, ϕ denotes the empty set and if $(x_1, x_2, \ldots, x_n)$ is a real valued vector, then $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ denotes the elements of the vector ordered in ascending order. For two vectors x and y, we say $x > y$, if

[1] Note that a maximum transfer rate from a source can be easily incorporated by augmenting the network topology with a source access link with capacity equal to the source transfer rate limit.

Research supported by a grant under an IISc-Nortel Memorandum of Understanding.

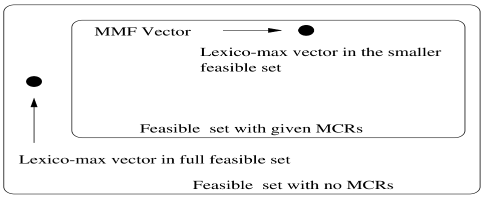

Fig. 1. Relationship between the feasible rate vectors with and without MCR requirements.

$x_i > y_i$, $\forall i$, and similarly for $<$, $\leq$, and $\geq$. $\mathcal{S}$ the set of sessions, $\mathcal{L}$ the set of links, C_l the capacity of link $l \in \mathcal{L}$, $\mathcal{C}$ denotes the ordered set $(C_l, l \in \mathcal{L})$, $\mathcal{L}_s$ the set of links used by session $s \in \mathcal{S}$, $\mathcal{S}_l$ the set of sessions through link $l \in \mathcal{L}$. Let r_i denote the rate of the ith session, $1 \leq i \leq |\ \mathcal{S}\ |$; $r = (r_1, r_2, \ldots, r_{|\mathcal{S}|})$ denotes the *rate vector*. Let μ_s denote the minimum cell rate for session $s \in \mathcal{S}$ and let $\mathcal{M}$ denote the set $\{\mu_s : s \in \mathcal{S}\}$.

For a rate vector r, and $l \in \mathcal{L}$ denote the total flow through link l by

$$f_l(r) \quad = \quad \sum_{s \in \mathcal{S}_l} r_s$$

Note that the 4-tuple $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$ characterises an instance of the bandwidth sharing problem.

B. Max-Min Fair Bandwidth Sharing with Nonzero MCR

The max-min fair sharing idea developed in [2], can be viewed as a natural generalisation of the conventional max-min fair allocation [5].

Definition II.1: A rate vector r **feasible** for the problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$ if for all $s \in \mathcal{S}$, $r_s \geq \mu_s$ and for all $l \in \mathcal{L}$, $f_l(r) = \sum_{s \in \mathcal{S}_l} r_s \leq C_l$.

Note that the set of feasible vectors is non-empty iff for all $l \in \mathcal{L}$, $\sum_{s \in \mathcal{S}_l} \mu_s \leq C_l$. We will assume that this is so, *with strict inequality*, in all the following discussions. [2]

Definition II.2: A feasible rate vector r is **max-min fair** for $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$ if it is not possible to increase the rate of a session s, while maintaining feasibility, without reducing the rate of some session p with $r_p \leq r_s$.

Definition II.3: Given a rate vector r, a link l is said to be a **bottle-neck link** for a session j if link l is saturated, i.e., $f_l(r) = C_l$, and for all the sessions $s \in \mathcal{S}_l$, such that $r_s > \mu_s$, $r_s \leq r_j$; i.e., every session in l, that is not at its MCR, has flow no more than that of session j, or equivalently $r_s \leq \max(\mu_s, r_j)$

Recalling the notation in Section II-A, we give the following definition.

Definition II.4: Let $x = (x_1, x_2, \cdots, x_n), y = (y_1, y_2, \cdots, y_n) \in \mathcal{R}^n$. Then y is defined as **lexicographically larger** than x (denoted $>_{lex}$) if $\tilde{y}_1 > \tilde{x}_1$, or if $\tilde{y}_1 = \tilde{x}_1$ then $\tilde{y}_2 > \tilde{x}_2$, etc.

The following theorem gives two equivalent characterisations of the Max-Min rate vector.

Theorem II.1: [2] If r is a feasible rate vector, then the following statements are equivalent: (i) r is Max-Min fair.
(ii) Every session $s \in \mathcal{S}$ has a bottle-neck link.
(iii) r is lexicographically the largest among all feasible rate vectors.

[2]Note that such feasibility will be ensured by an admission control procedure.

The relationship between the above definition and of Max-Min allocation is shown in Figure 1. Hence our definition with MCR is a generalisation of that without MCRs.

III. MMF Formulation as the Root of an Equation

In this section we motivate the construction of distributed algorithms for computing a MMF allocation. Consider a MMF allocation problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$. Let $r^* = (r_1^*, r_2^*, \ldots)$ denote the Max-Min fair vector. Let $\mathcal{L}_b \subset \mathcal{L}$ be the set of all links that are bottlenecks for at least one session through them. Let η_l be a link parameter associated with link $l \in \mathcal{L}$. The following result has been proved in [2].

Theorem III.1: For all $l \in \mathcal{L} \backslash \mathcal{L}_b$ let $\eta_l^* = \infty$. Then if η_l^*, $l \in \mathcal{L}_b$, satisfy

$$\sum_{s \in \mathcal{S}_l} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j^*) = C_l \ \ \forall l \in \mathcal{L}_b \tag{1}$$

then

$$r_s^* = \max(\mu_s, \min_{l \in \mathcal{L}_s} \eta_l^*) \tag{2}$$

Thus any solution to Equation (1) yields the (unique) MMF rate vector via Equation (2). We call η_l the link control parameter for link l.

Let $\underline{\eta} = (\eta_l, l \in \mathcal{L})$ denote the vector of link control parameters. Define the vector function $\underline{f}(\underline{\eta}) = (f_l(\underline{\eta}), l \in \mathcal{L})$ by

$$f_l(\underline{\eta}) = \sum_{s \in \mathcal{S}_l} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j)$$

Let $\underline{C}$ denote the vector of link capacities, for a MMF allocation problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$, if it is known that every link is bottleneck for at least one session, i.e., $\mathcal{L}_b = \mathcal{L}$, then Theorem III.1 implies that the MMF allocation can be computed by first solving for η such that

$$\underline{f}(\underline{\eta}) = \underline{C} \tag{3}$$

For each session s, we then compute a *virtual rate* as the minimum of the link control parameters along its path, i.e.,

$$\hat{r}_s = \min_{l \in \mathcal{L}_s} \eta_l$$

A session's rate is the virtual rate if the virtual rate $\hat{r}_s$ is larger than the session's MCR, otherwise it is the MCR, i.e.,

$$r_s = \max(\mu_s, \hat{r}_s)$$

IV. Stochastic Approximation Algorithms

The result in Section III shows that computing a MMF rate is simply a problem of finding the link control parameters to maximise link bandwidth utilisation. A class of successive approximation type algorithms have been proposed in the literature (see,e.g., [8] [7]) for distributed computation of the link control parameters. These algorithms operate by increasing or decreasing the link control parameters depending on whether the total input rate to the link is less than or greater than the link capacity. The link control parameter update functions for simple iterative additive and multiplicative schemes are given below; k denotes the iteration index. Recalling that $f_l(\underline{\eta})$ is the flow in link l and letting $n_l = |\ \mathcal{S}_l\ |$, the algorithm in [8] is

$$\eta_l(k+1) = \eta_l(k) + (C_l - f_l(\underline{\eta}(k)))/n_l \tag{4}$$

and the algorithm in [7] is

$$\eta_l(k+1) = \eta_l(k)(C_l / f_l(\underline{\eta}(k))) \tag{5}$$

In [8] there is no consideration of nonzero MCRs. In [7], the rate computed is added to the MCR to give the session rate. Both approaches require a fixed number C_l for the computations to converge to the max-min fair share. However, note that ABR sessions are expected to utilise bandwidth left over after servicing the CBR and VBR class. Due to the inherent bursty nature of the VBR traffic, the leftover bandwidth is of a perturbed nature. In addition, if the link utilisations are obtained by measurements, then measurement errors will have to be considered. We club the effects of perturbed available capacity and measurement error into a zero mean random process $\omega(k)$ and view the available capacity as the sum of a constant and $\omega(k)$. In [7], the authors suggest the use of an estimate of the mean available capacity obtained by filtering the "noisy" capacity sequence, however, they have not analytically proved the tracking ability of their algorithm. In this paper we resort to stochastic approximation theory to obtain such a proof.

To illustrate this approach, consider the case of a single link network with a nominal available capacity C, and with some sessions $s \in \mathcal{S}$, each with MCR μ_s. Observe that Equation(3) asks for η^* such that

$$C - \sum_{s \in \mathcal{S}} (\mu_s, \eta^*) = 0$$

This corresponds with the intersection of the solid curve and the solid line shown in Figure 2. In the iterations of Equation (4) or (5), if we replace C by $C + \omega(k)$, then the sequence $\eta(k)$ will not converge, but will jitter around η^*; see Figure 2. Thus we see that at each link we have the problem of searching

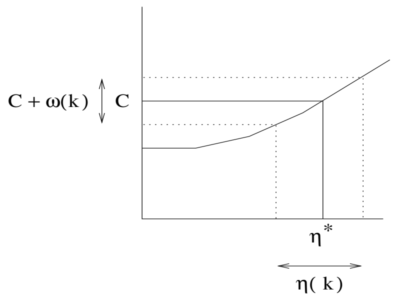

Fig. 2. Sessions in a single link. η^* is the link control parameter with the mean capacity C. When $C(k)$ is random, the corresponding $\eta(k)$ varies around η^*.

for the root of a certain function given only noisy observations of the function. Such problems have been tackled using stochastic approximation algorithms [4]. The main idea is to weight the "error estimate" at each iteration by a reducing gain sequence $\alpha(k)$. Hence, incorporating MCRs in the successive approximation algorithm, and introducing the decreasing gain sequence, yields the algorithm.

$$\eta_l(k+1) = \eta_l(k) + \alpha(k) \left(\frac{C_l(k) - \sum_{s \in \mathcal{S}_l} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j(k))}{n_l} \right) \tag{6}$$

where $C_l(k) = C_l + \omega(k)$. The sequence $\alpha(k)$ is chosen such that it has the following properties

$$\sum_{k=1}^{\infty} \alpha(k) = \infty \quad \sum_{k=1}^{\infty} \alpha(k)^2 < \infty$$

This property is satisfied by a sequence of the form

$$\alpha(k) = 1/k^x \quad 1/2 < x \leq 1$$

Such algorithms are said to be of Robbins-Monro type [11]. We now present a synchronous distributed algorithm that uses this idea. We use the notation that

$$[x]_a^b = \min(b, \max(a, x))$$

We assume that each link l has a maximum available capacity C_l^{max}. A portion of C_l^{max}, with mean $C_l < C_l^{max}$ is available for ABR traffic. At each computation of a link control parameter, the iterate computed using Equation (6) is truncated to remain between 0 and C_l^{max}.

Algorithm IV.1:
Initialisation:
$k = 0$, for each $l \in \mathcal{L}$, $0 \leq \eta_l(0) < C_l^{max}$
Do forever

1. $k \leftarrow k + 1$
2. For all $l \in \mathcal{L}$, update the link control parameter

$$\eta_l(k) = \left[\eta_l(k-1) + \alpha_l(k) \frac{C_l + \omega_l(k) - \sum_{s \in \mathcal{S}_l} r_s(k-1)}{n_l} \right]_0^{C_l^{max}}$$

3. For each session $s \in \mathcal{S}$, update the virtual rate

$$\hat{r}_s(k) = \min_{l \in \mathcal{L}_s} \eta_l(k)$$

4. Each session $s \in \mathcal{S}$ calculates its actual rate

$$r_s(k) = \max(\hat{r}_s(k), \mu_s)$$

To understand why Algorithm IV.1 is a distributed algorithm note that to compute $\eta_l(k+1)$ at link l, we only need $\eta_l(k)$ which is the previously computed link control parameter and $\sum_{s \in \mathcal{S}_l} r_s(k-1)$, which is the total flow through link l. Both these quantities are locally available information. In the following theorem we prove that the session rates converge to the max-min fair value for the case when every link is a bottleneck for some session and the equation

$$C - f(\eta^*) = 0$$

has a unique solution η^*. Extensions of this result to more general conditions and to the asynchronous situation (switches updating their link control values based on delayed previous values, and the updates not being synchronous at the various switches) will be presented in future papers.

Theorem IV.1: Consider a max-min problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$, with unique η^* such that

$$C - f(\eta^*) = 0$$

Let the noise sequence $\omega_l(k)$ be i.i.d., bounded with zero mean for all $l \in \mathcal{L}$, then the sequence of iterations given in Algorithm IV.1 yield $\eta_l(k)$ such that, almost surely,

$$\lim_{k \to \infty} r_s(k) = \lim_{k \to \infty} \max(\mu_s, \min_{l \in \mathcal{L}_s} \eta_l(k)) = r_s^*$$

where

$$r_s^* = \max(\mu_s, \min_{l \in \mathcal{L}_s} \eta_l^*)$$

Since we have assumed a zero mean, bounded i.i.d. noise sequence $\omega_l(k)$, and have bounded the values of $\eta_l(k)$ by truncation, the proof of Theorem IV.1 consists of showing that a certain ordinary differential equation is asymptotically stable [11]. The main idea involved is the observation that the asymptotic evolution of the sequence $\eta_l(k)$ is equivalent to the evolution of the solution of this differential equation. Heuristically, the differential equation is obtained as follows (see also [11]). Rewrite Step 2 of Algorithm IV.1 as

$$\eta_l(k) - \eta_l(k-1) =$$
$$\left[\eta_l(k-1) - \alpha_l(k) \frac{C_l + \omega_l(k) - \sum_{s \in \mathcal{S}_l} r_s(k-1)}{n_l} \right]_0^{C_l^{max}} - \eta_l(k-1)$$

If we view the decreasing gain $\alpha_l(k)$ as a step size in time, then the rate of change of $\eta_l(k)$ can be written as

$$(\eta_l(k) - \eta_l(k-1)) / \alpha_l(k) =$$
$$\frac{\left[\eta_l(k-1) - \alpha_l(k) \frac{C_l + \omega_l(k) - \sum_{s \in \mathcal{S}_l} r_s(k-1)}{n_l} \right]_0^{C_l^{max}} - \eta_l(k-1)}{\alpha_l(k)}$$

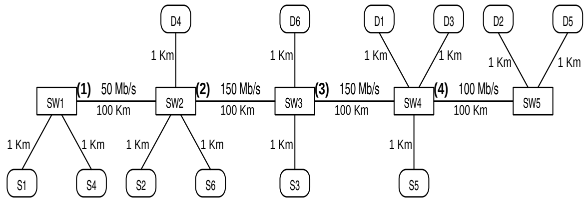

Fig. 3. Network topology; showing switches (SWi), links, sessions (Si,Di), mean link capacities and link lengths.

Session s	(S1, D1)	(S2, D2)	(S3, D3)	(S4, D4)	(S5, D5)	(S6 D6)
MCR μ_s	30	60	0	0	0	0
Max-Min	30	60	60	20	40	60
Link	1	2	3	4		
η_l^*	20	60	60	40		

TABLE I

SESSION MCRS, MAX-MIN FAIR RATES AND LINK CONTROL PARAMETERS.

As k increases to infinity, $\alpha_l(k)$ goes to zero, and the limiting rate of change is given by the following differential equation.

$$\dot{\eta}_l\ (t) = \lim_{\Delta \to 0} \frac{[\eta_l(t) + \Delta(C_l - f_l(\eta(t))]_0^{C_l} - \eta_l(t)}{\Delta} \qquad (7)$$

The proof of asymptotic stability of the differential equation (7) is given in Appendix A. The case where the noise processes are more general can be handled; these extensions will be discussed in future work.

V. SIMULATION RESULTS FOR AN EXAMPLE WIDE AREA NETWORK

In this section we present simulation results from the stochastic approximation algorithm applied to a wide area network example. The available link capacities are random with time varying means. The source rate adjustment mechanism prescribed in the [1] are simulated. There are propagation delays, and hence the link control parameter updates are asynchronous. The simulations demonstrate how the session rates track the max-min fair rates.

We have used the NIST ATM simulator [3] for the results presented here. This simulation package provides users with modules for switches, terminal equipment and links. A GUI is also provided to enable users to build a given network topology and witness the progress of the simulation. The implementation of the modules follows the guidelines provided by the ATM Forum closely. We have included the stochastic approximation algorithm, discussed above, in the switch module.

A. Simulated Network

The network being simulated is illustrated in Figure V-A. The source and destination for session i are denoted by Si and Di respectively. The WAN links are labelled $L1, \ldots, L4$. The mean available link capacity is indicated above the links. The access links are assumed to have infinite capacity. The blocks labelled SW1 to SW4 denote the switches, the link control parameter computations for link i are carried out in SWi. Table I gives the MCR, the Max-Min fair rates and the link control parameters required to achieve the Max-Min fair rates.

B. Implementation of the Algorithm

For the purpose of ABR flow control, a special cell called the RM (Resource Management) cell has been introduced for the exchange of rate information between the sources and the switches. An RM cell is sent by the source after every N data cells, where N is some chosen constant number. There are three important fields in the RM cell for the purpose of Explicit Rate control. They are the ER, MCR, and CCR fields. The MCR field holds the minimum cell rate for the session and the CCR field holds the current sending rate.

The ER (for Explicit Rate) field is set by the source to the desired peak rate. At a link, in the forward path of the RM cell (i.e., from the source to the destination), if the link control parameter is less that that the ER field value, then the ER field is changed to the link control parameter, otherwise the ER field is left unchanged. Thus after the RM cell has visited all switches along the path for the session, the ER field will hold the minimum of the link control parameters of all these switches, given that the desired peak rate is higher than this minimum. Hence this process executes Step 3 of Algorithm IV.1.

The session destination sends the RM cell back to the source along the same path taken by the cell in the forward direction. Every switch now reads the ER of the returning RM cell and stores this value. Thus the switch is informed of the virtual rate ($\hat{r}_s$ of each session through it. The MCR of a session can also be read of the RM cell. Hence all information for the execution of Step 1 of Algorithm IV.1 is available at the switch.

The choice of link control parameter update epochs is an important issue when the computations at the switches are not synchronised, and when session responses to link control updates are delayed due to propagation delays. Highly frequent updates could lead to oscillations in the computed link control parameter. In our implementation we compute a new link parameter only after receiving at least one returning RM cell from each session with an ER value less than or equal to the present link control parameter. Since we wait for returning RM cells to arrive from each session before computing, the time interval between link control parameter computations are determined by the session whose time spacing between RM cells is the largest. We note that this choice of update epochs is heuristic, however the simulation results seem to indicate that it is an effective method. The problem of choosing an optimum update interval will be taken up in our future studies.

In the simulations with stochastic link capacities, the noise sequence $\omega_l(k)$ for a link with mean available link capacity C_l is assumed to be i.i.d., uniformly distributed between $-0.25C_l$ and $0.25C_l$. The gain sequence $\alpha_l(k)$ is chosen to be

$$\alpha_l(k) = \frac{1}{1 + \frac{k}{10n_l}}$$

where n_l is the number of sessions through link l.

C. Simulation Results

Two sets of simulation results are presented here. In the first set of results, we keep the mean of the link capacities fixed. In practice this would correspond to the case when the background CBR

[3] ftp://isdn.csnl.nist.gov/atm-sim

and VBR traffic sessions are fixed. In the second set of results we vary the mean of the link capacities available for ABR traffic. This captures the realistic scenario in which there will be entry and exit of CBR/VBR sessions. Whenever the mean of the link capacity changes, we reset the gain term in the stochastic approximation algorithm to a large value. This requirement is not unreasonable, as a switch would be expected to know the entry and exit times of the CBR/VBR sessions.

C.1 Results with Fixed Mean Capacity

The mean value of the link capacities has been indicated in Figure V-A. In Figure 4, we show the time series of the session rates obtained when the link capacities are kept fixed at their mean values. The algorithm used is identical to Algorithm IV.1 with the gain term $\alpha_l(k)$ removed. This is a slightly modified version of Hayden's Algorithm [12]. Note that the session rates converge to the Max-Min values.

In Figure 5, we give the stochastic link rate sequence. The session rates obtained using stochastic approximation type updates are given in Figure 6. Note that in spite of the noisy capacity, the session rates obtained are very close to the Max-Min values.

C.2 Results with Varying Mean Capacity

In Figure 7, the variation of mean link capacity is given. The actual link capacity sequence used is given in Figure 8. The session rates obtained with this link capacity is given in in Figure 9. Note that the session rates during a period of constant mean converge to the max-min fair value corresponding to the mean link capacity in that region. We note that the session rates adapt well to the increasing means of the link capacity. However, when the means of the link capacities reduce, the convergence is poor. This is due to the fact that, there are an increased number of in-rate RM cells at a higher transmission rate, which are in transmission when the mean rate changes. The sudden fall of the available link capacity, causes these RM cells to be backlogged at the output port of the switches. Hence, the RM cells carrying the new reduced rates experience more delay before arriving at the source.

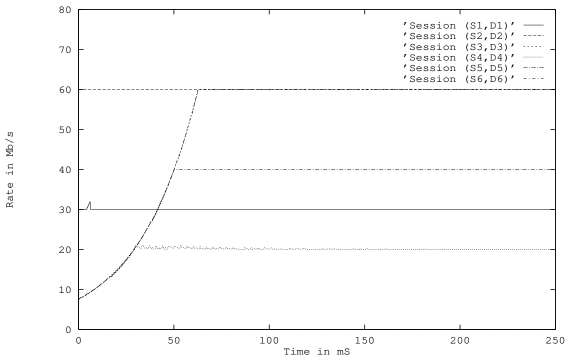

Fig. 4. Time series of session rates when link capacities are constant.

VI. Conclusion

In this paper we have discussed stochastic approximation algorithms for the problem of max-min fair allocation of rates to ABR sessions with MCR requirements. We have presented a distributed

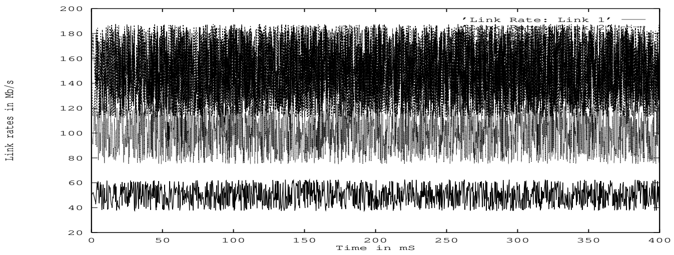

Fig. 5. Time series of the stochastic link capacity.

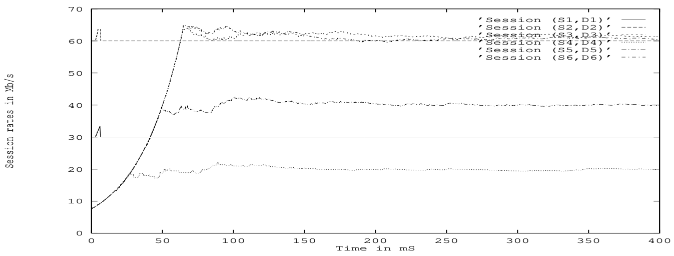

Fig. 6. Time series of session rates computed with the stochastic approximation algorithm.

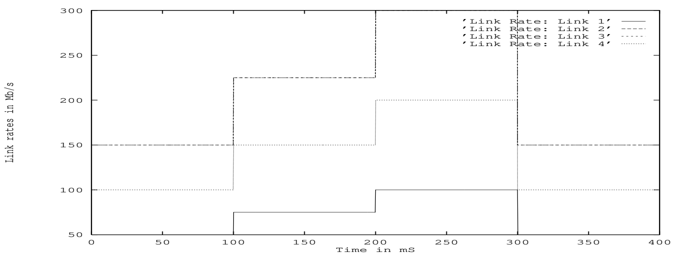

Fig. 7. Plot of the variation of the mean of available link capacities.

synchronous stochastic approximation algorithm and analytically proved that the sessions rate converge to the max-min fair share. This approach seems to hold the promise of simple implementable algorithms for max-min fair rate computation. We presented simulation results for an example wide area network, in which the random rates of the links were assumed to have varying means. The session rates were observed to converge.

In our future work, we plan to analytically examine asynchronous distributed stochastic approximation algorithms. We shall also study the performance of stochastic approximation algorithms when the estimates of link capacity and link utilisation are obtained using various measurement schemes that have mea-

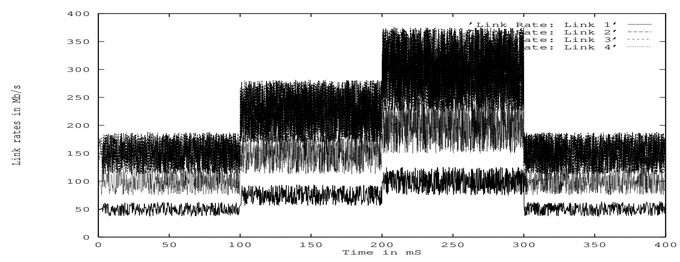

Fig. 8. Time series of stochastic link capacity with varying mean.

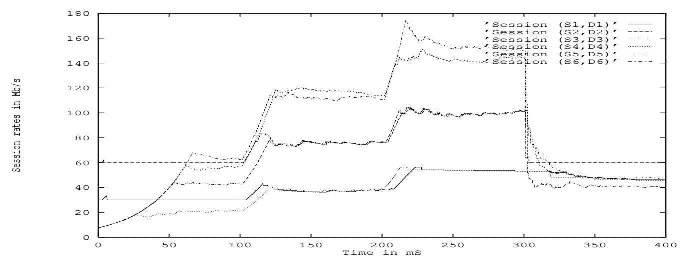

Fig. 9. Time series of session rates obtained using stochastic approximation and adaptation when link capacities are stochastic with varying mean.

surement error. The choice of the link control update interval at the switches is also an important issue for future study.

Appendix A: Proof of Theorem IV.1

Lemma A.1: Let $x(t), y(t) \in \Re^n$, consider the two differential equations

$$\dot{x}(t) = g_1(x(t), t)$$
$$\dot{y}(t) = g_2(y(t), t)$$

Assume that g_1 and g_2, are continuous in both arguments. Let, for $1 \leq l \leq n$, $g_{2l}(y(t), t)$, denote the lth component of $g_2(y(t), t)$. Assume that (i) $g_{2l}(y(t), t) = g_{2l}(y_l(t), t)$, i.e., the lth component $g_{2l}(y(t), t)$ depends only on the lth component of $y(t)$
(ii) g_2 is a non decreasing in its first argument.
(iii) For all $u \in \Re^n$, let $g_1(u, t) \geq g_2(u, t)$.
At t_0, let the initial conditions be such that $x(t_0) = y(t_0)$, then for all $t \in (t_0, \infty)$ $x(t) \geq y(t)$.

We omit the proof of the above lemma due to lack of space.

In order to prove the stability of the differential equation (7), we require to partition the sets of sessions and links. This partitioning is based on the operation of a centralised algorithm for computing the max-min allocation. Given the max-min problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$. Let us define $\mathcal{L}(1) = \mathcal{L}$ and $\mathcal{S}(1) = \mathcal{S}$. A centralised algorithm would operate as follows. Consider each link and the sessions through it in isolation. Compute the max-min allocation for each of the single link problems separately. Let η_l^1, be the link control parameter computed for link l, i.e., η_l^1 solves

$$\sum_{s \in \mathcal{S}_l} \max(\mu_s, \eta_l^1) = C_l$$

Let

$$p^1 = \min_{l \in \mathcal{L}(1)} \eta_l^1$$

For each session let its allocated rate be given by

$$r_s = \max(\mu_s, p^1) \tag{8}$$

Let $\mathcal{L}^1$ be the set of all links l, such that

$$\eta_l^1 = p^1$$

For all links in $\mathcal{L}^1$, the allocation given by Equation (8) results in full capacity utilisation, i.e., $\forall l \in \mathcal{L}^1 \ \sum_{s \in \mathcal{S}_l} r_s = C_l$. Let $\mathcal{S}^1$ be the set of all sessions that have at least one link in $\mathcal{L}^1$; this is the set of sessions that got bottlenecked at Step 1. Now we define $\mathcal{L}(2)$ as we remove the links $l \in \mathcal{L}^1$ from $\mathcal{L}(1)$ and we call the resulting set $\mathcal{L}(2)$, i.e.,$\mathcal{L}(2) = \mathcal{L}(1) \backslash \mathcal{L}^1$. Note that $\mathcal{L}(2)$ is the set of all links which have spare capacity, when the allocation is done as explained above. Further define $\mathcal{S}(2)$ as $\mathcal{S}(2) = \mathcal{S}(1) \backslash \mathcal{S}^1$. Note that $\mathcal{S}^1$ is the set of all sessions that have reached their fair allocations at Step 1. For each link in $\mathcal{L}(2)$, we reduce the capacity by an amount used up by sessions $s \in \mathcal{S}^1$. To the reduced set of links, sessions and reduced link capacities we apply the above procedure again and obtain a minimum link control parameter p^2, and the sets $\mathcal{S}^2$ and $\mathcal{L}^2$. This procedure is carried on until all sessions have been removed. Let M denote the number of steps executed before all sessions are removed. The procedure gives us a partition of $\mathcal{S}$, i.e.,$\mathcal{S}^i$, $i = 1, \ldots, M$. If $s \in \mathcal{S}^i$, we call it bottlenecked at the iteration. Note that all $s \in \mathcal{S}^i$, have at least one link $l \in \mathcal{L}^i$ as their bottleneck link (c.f., Definition II.3). Observe that

$$\mathcal{L}(i) = \cup_{j=i}^{M} \mathcal{L}^i \text{ and } \mathcal{S}(\mathrm{i}) = \cup_{\mathrm{j=i}}^{\mathrm{M}} \mathcal{S}^{\mathrm{i}}$$

$\mathcal{L}(i)$ is the set of links not yet bottlenecked just before the execution of the ith step and $\mathcal{L}^i$ is the set of links that get bottlenecked at the ith step. Similarly for $\mathcal{S}^i$ and $\mathcal{S}(i)$. In our notation for these (and subsequently defined) sets the following rule is consistently followed: the iteration index i appearing as an argument (i.e., (i)) indicates a cumulative set for iterations $\geq i$, and the index i appearing as a superscript indicates a set corresponding to iteration i. For each link $l \in \mathcal{L}$ we define the following

$$\mathcal{S}_l(i) = \mathcal{S}(i) \cap \mathcal{S}_l$$
$$\mathcal{S}_l^i = \mathcal{S}^i \cap \mathcal{S}_l$$

$\mathcal{S}_l(i)$ is the set of all sessions through link l that are not bottle necked prior to the execution of ith step and $\mathcal{S}_l^i$ is the set of sessions in link l that are bottlenecked at the ith step. Note that if $l \in \mathcal{L}^i$, then $\mathcal{S}_l^j = \phi \ \forall j > i$. Similarly for each session $s \in \mathcal{S}$ we define

$$\mathcal{L}_s(i) = \mathcal{L}(i) \cap \mathcal{L}_s$$
$$\mathcal{L}_s^i = \mathcal{L}^i \cap \mathcal{L}_s$$

$\mathcal{L}_s^i$ is the set of all links for session s that become bottlenecks at the ith iteration. Note that if $s \in \mathcal{S}^i$, then for $j < i$, $\mathcal{L}_s^j = \phi$.

From the above obtained partitions, we deduce an alternate partition on the set of sessions. Let $p^0 = 0$ and $p^{M+1} = \infty$; for $i = 1, \ldots, M$ define

$$\hat{\mathcal{S}}^i = \{s : s \in \mathcal{S}^i, \mu_s \leq p^i, \text{ or } s \in \mathcal{S}^j, \ j \leq i, p^i < \mu_s \leq p^{i+1}\} \tag{9}$$

$\hat{\mathcal{S}}^i$ is the set of all sessions which obtain their max-min allocation at the ith iteration and their rate allocation is precisely p^i, or those session which have been bottlenecked prior to the ith iteration, but have MCR lying between p^i and p^{i+1}. We define the set $\hat{\mathcal{S}}(i)$ as follows,

$$\hat{\mathcal{S}}(i) = \cup_{j=i}^{M} \hat{\mathcal{S}}^j \ \ i = 1, ..., M+1$$

For each link $l \in \mathcal{L}$ we define

$$\hat{\mathcal{S}}_l(i) = \hat{\mathcal{S}}(i) \cap \mathcal{S}_l$$

Now consider the differential equation

$$\dot{\eta}_l\ (t) = \lim_{\Delta \to 0} \frac{[\eta_l(t) + \Delta(C_l - f_l(\eta(t))]_0^{C_l^{max}} - \eta_l(t)}{\Delta} \quad (10)$$

Let the rate allocation to a session $s \in \mathcal{S}$ be given by

$$r_s(t) = \max(\mu_s, \min_{l \in \mathcal{L}_s} \eta_l(t))$$

If the vector $r^* = (r_s^*, s \in \mathcal{S})$ is the max-min fair rate vector, then we show that

$$\lim_{t \to \infty} r_s(t) = r_s^* \ \ \forall s \in \mathcal{S}$$

Lemma A.2: Consider a link $l \in \mathcal{L}$, given the (one dimensional) differential equation

$$\dot{x}\ (t) = \lim_{\Delta \to 0} \frac{\left[x(t) + \Delta(C_l - \sum_{s \in \mathcal{S}_l} \max(\mu_s, x(t)) + \epsilon(t))\right]_0^{C_l^{max}} - x(t)}{\Delta}$$

with $\epsilon(t)$ continuous and

$$\lim_{t \to \infty} \epsilon(t) = 0$$

Let the initial condition $x(t_0) \in [0, C_l^{max}]$, then

$$\lim_{t \to \infty} x(t) = x^*$$

where x^* solves

$$C_l - \sum_{s \in \mathcal{S}_l} \max(\mu_s, x^*) = 0$$

Proof: Note that due to truncation and $x(t_0) \in [0, C_l^{max}]$, $x(t) \in [0, C_l^{max}]$ for all t. If x^* is unique, we have $x^* > \min_{s \in \mathcal{S}_l} \mu_s$. Choose ϵ_1 and ϵ_2 such that (i) $0 < \epsilon_1 < \epsilon_2$, (ii) $\epsilon_2 < x^* - \min_{s \in \mathcal{S}_l} \mu_s$ Let T be large enough so that for all $t \geq T$ $\mid \epsilon(t) \mid < \epsilon_1$. Now let $0 < x(t) < x^* - \epsilon_2$ then

$$C_l - \sum_{s \in \mathcal{S}_l} \max(\mu_s, x(t)) + \epsilon(t) > C_l - \sum_{s \in \mathcal{S}_l} \max(\mu_s, x^*) + \epsilon_2 - \epsilon_1 > 0$$

Hence $\dot{x}\ (t) > 0$ and $x(t)$ increases till $x(t) \geq x^* - \epsilon_2$. Now if for $t_2 > T$, $x(t) \geq x^* - \epsilon_2$ and for some $t_3 > t_2$, $x(t_2) < x^* - \epsilon_2$, then consider $t_4 = \sup\{t_2 < t < t_3 :\ x(t) \geq x^* - \epsilon_2\}$. Note that due to continuity of $x(t)$, $x(t_4) = x^* - \epsilon_2$. But then $\dot{x}\ (t_4) > 0$, which means that there is a $\delta > 0$ such that for $t \in (t_4, t_4 + \delta)$, we have $x(t) > x^* - \epsilon_2$. This contradicts the definition t_4. Hence $x(t)$ continues to remain above $x^* - \epsilon_2$.

Arguing similarly, we can show that, given any arbitrarily small ϵ, for sufficiently large t, we have, $x(t) < x^* + \epsilon$.

Now consider the case of non-unique x^*, then any $x^* \leq \min_{s \in \mathcal{S}_l} \mu_s$, will solve the Equation (VI). Then argue similarly to show that for any given $\epsilon > 0$, for t large enough, $x(t) < \min_{s \in \mathcal{S}_l} \mu_s + \epsilon$. Hence the proof. □

Lemma A.3: Consider the partition of sets given by Equations (9). Let $1 \leq i \leq M$. If for all $s \in \hat{\mathcal{S}}^j,\ \ j < i$

$$\lim_{t \to \infty} r_s(t) = r_s^*$$

then for all $s \in \hat{\mathcal{S}}^i$,

$$\lim_{t \to \infty} r_s(t) = r_s^*$$

The proof of Lemma A.3 is divided into two parts. In the first part we consider any session $s \in \hat{\mathcal{S}}^i$ and $\mu_s \leq p^i$. For all sessions in $l \in \mathcal{L}(i)$, we construct an auxiliary differential equation whose solution is a lower bound to $\eta_l(t)$. The solutions of the auxiliary differential equation for $l \in \mathcal{L}^i$ converge to p^i. The solutions for the other links converge to a value $> p^i$. This allows us to conclude that for large enough t, $r_s(t) > p^i - \epsilon$ for any $\epsilon > 0$. We then argue that if for large enough t, $r_s(t) > p^i$, then the derivative $\dot{\eta}_l\ (t) < 0$ for all links $l \in \mathcal{L}_s^i$, thus forcing $r_s(t)$ to reduce. This fact along with the lower bounding allow us to conclude that $r_s(t)$ converges to p^i.

In the second part we consider any session $s \in \hat{\mathcal{S}}^i$ with $\mu_s > p^i$. The hypothesis of the lemma along with the first part of the proof allow us to conclude that $r_s(t)$ converges for all s such that $r_s^* \leq p^i$. We then show using arguments similar to the first part, that for large enough t, if all links $l \in \mathcal{L}_s$ have an $\eta_l(t) > \mu_s$, then the derivative $\eta_l(t) < 0$ for all links $l \in \mathcal{L}_s^j, j \leq i$. We then conclude that there will be at least one link in $\mathcal{L}_s$ in which $\eta_l(t)$ remains below or equal to μ_s for sufficiently large t.

Proof: Consider the set $\mathcal{L}(i)$. For each $l \in \mathcal{L}(i)$, define the following.

$$\epsilon_l^i(t) = \sum_{s \in \mathcal{S}_l \setminus \hat{\mathcal{S}}_l(i)} (r_s^* - r_s(t)) \quad (11)$$

$\epsilon_l^i(t)$ is the "error" in the flow of sessions that are assumed, by the hypothesis, to have flows that converge. Further define

$$\hat{C}_l^i = C_l - \sum_{s \in \mathcal{S}_l \setminus \hat{\mathcal{S}}_l(i)} r_s^* \quad (12)$$

$\hat{C}_l^i$, is the capacity of link l remaining after removing the max-min flows of all sessions whose flows are assumed to converge.

$$f_l^i(\eta(t)) = \sum_{s \in \hat{\mathcal{S}}_l(i)} r_s(t) = \sum_{s \in \hat{\mathcal{S}}_l(i)} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j(t)) \quad (13)$$

$f_l^i(\eta(t))$ is the sum of the flows in link l of all sessions whose flows have not been assumed to converge converge in the hypothesis of this theorem. With the definitions in Equations (11) (12) and (13), for each $l \in \mathcal{L}(i)$ we can rewrite the differential equation (10) as

$$\dot{\eta}_l\ (t) = \lim_{\Delta \to 0} \frac{\left[\eta_l(t) + \Delta(\hat{C}_l^i - f_l^i(\eta(t)) + \epsilon_l^i(t))\right]_0^{C_l^{max}} - \eta_l(t))}{\Delta} \quad (14)$$

Let $\eta^i(t) = (\eta_l(t), l \in \mathcal{L}(i))$, denote the vector of such $\eta_l(t)$. To obtain a lower bound on $\eta^i(t)$, we consider the auxiliary differential equation

$$\dot{x}_l\ (t) = \lim_{\Delta \to 0} \frac{\left[x_l(t) + \Delta\left(\hat{C}_l^i - \sum_{s \in \hat{\mathcal{S}}_l(i)} \max(\mu_s, x_l(t)) + \epsilon_l^i(t)\right)\right]_0^{C_l^{max}} - x_l(t)}{\Delta} \quad (15)$$

and let $x^i(t) = (x_l(t), l \in \mathcal{L}(i))$. Observe that these ODEs correspond to the isolated link problem at the ith step in the centralised algorithm. Now observe that

$$\hat{C}_l^i - \sum_{s \in \hat{\mathcal{S}}_l(i)} \max(\mu_s, u_l(t)) \leq \hat{C}_l^i - \sum_{s \in \hat{\mathcal{S}}_l(i)} \max(\mu_s, \min_{j \in \mathcal{L}_s} u_j(t))$$

Hence given identical initial conditions for Equations (14) and (15), the conditions of Lemma A.1 are satisfied with $g_1(\cdot, \cdot)$ corresponding to Equation (14) and $g_2(\cdot, \cdot)$ corresponding to Equation (15) and we have

$$\eta_l(t) \geq x_l(t) \ \ \forall t \quad (16)$$

Now using Lemma A.2 note that $x_l(t)$ converges for all $l \in \mathcal{L}(i)$ and also

$$\lim_{t \to \infty} x_l(t) = p^i \ \ \forall l \in \mathcal{L}^i \quad (17)$$

Let

$$\lim_{t \to \infty} x_l(t) = q_l > p^i \ \ \forall l \in \mathcal{L}(i) \setminus \mathcal{L}^i \quad (18)$$

From Equations (16) (17) (18) we can conclude that

$$\liminf_{t \to \infty} \eta_l(t) \quad \geq p^i \qquad \forall l \in \mathcal{L}(i) \quad (19)$$

$$\Rightarrow \liminf_{t \to \infty} r_s(t) \quad \geq \quad \max(\mu_s, p^i) \ \ \forall s \in \mathcal{S}(i) \quad (20)$$

Now consider $u \in \hat{\mathcal{S}}^i \cap \mathcal{S}^i$. This is the set of all sessions that get bottlenecked at the ith iteration of the centralised algorithm and have a max-min value of p^i. Hence $r_u^* = p^i$ $(\geq \mu_u)$. By Equation (20)

$$\liminf_{t\to\infty} r_u(t) \geq p^i \tag{21}$$

We now show that given an arbitrary $\epsilon > 0$, for large enough t, $r_u(t) < p^i + \epsilon$, i.e., $\limsup_{t\to\infty} r_u(t) \leq p^i$ Let $\overline{n} > \max_{l\in\mathcal{L}} \mid \mathcal{S}_l \mid$, choose ψ such that

$$\psi < \min\left(\min_{l\in\mathcal{L}_s} \frac{C_l^{max} - p^i}{2\overline{n}}, \min_{l\in\mathcal{L}(i)\backslash\mathcal{L}^i} \frac{q_l - p^i}{2\overline{n}}\right) \tag{22}$$

Note that

$$\begin{aligned} p^i + \overline{n}\psi &< q_l - \overline{n}\psi \ \forall l \in \mathcal{L}(i)\backslash\mathcal{L}^i \\ p^i + \overline{n}\psi &< C_l^{max} - \overline{n}\psi \ \forall l \in \mathcal{L}_u^i \end{aligned}$$

Choose $\xi_1 > 0$ and $\xi_2 > 0$ such that

$$(\overline{n} - 1)\xi_1 + \xi_2 < \overline{n}\psi$$

Given (19 note that we can choose T large enough so that for all $t > T$, we have

$$\begin{aligned} \eta_l(t) &> p^i - \xi_1 \ \forall l \in \mathcal{L}^i \\ \eta_l(t) &> q_l - \overline{n}\psi \ \forall l \in \mathcal{L}(i)\backslash\mathcal{L}^i \\ \mid \epsilon_l^i(t) \mid &< \xi_2 \end{aligned}$$

If there exists some $t_1 > T$ such that $r_u(t) \geq p^i + \overline{n}\psi$, then for all $l \in \mathcal{L}_u^i$ we have

$$\begin{aligned} &\hat{C}_l^i - f_l^i(\eta(t_1)) + \epsilon_l^i(t_1) \\ &\quad < \hat{C}_l^i - (p^i + \overline{n}\psi) - \sum_{s\in\hat{\mathcal{S}}_l(i)\backslash\{u\}} \max(\mu_s, p^i - \xi_1) + \xi_2 \\ &\quad < \hat{C}_l^i - (p^i + \overline{n}\psi) - \sum_{s\in\hat{\mathcal{S}}_l(i)\backslash\{u\}} \max(\mu_s, p^i) + (\overline{n} - 1)\xi_1 + \xi_2 \\ &\quad < 0 \end{aligned}$$

Hence note that $\dot{\eta}_l\ (t_1) < 0$ for all $l \in \mathcal{L}_u^i$. Thus $r_u(t)$ will continue to decrease till $r_u(t) < p^i + \overline{n}\psi$. Now consider $t_2 > T$ such that $r_u(t) < p^i + \overline{n}\psi$. It can now be shown (using an argument similar to the argument in Lemma A.2) that, $\forall\ t > t_2$, $r_u(t) < p^i + \xi$. Since ξ is arbitrary and positive subject to Equation (22), we have shown that for all sessions $u \in \hat{\mathcal{S}}^i \cap \mathcal{S}^i$ (i.e, those sessions that bottleneck at p^i at the ith step of the centralised algorithm), we have

$$\limsup_{t\to\infty} r_u(t) \leq p^i \tag{23}$$

Now using Equation (21) and (23) we conclude that

$$\lim_{t\to\infty} r_u(t) = p^i \tag{24}$$

Now consider those sessions with $u \in \hat{\mathcal{S}}^i$ with $\mu_u > p^i$. Let $u \in \mathcal{S}^k,\ k \leq i$ and $p^{i+1} \geq \mu_u > p^i$. Note that from the hypothesis of the lemma and equation (24), we have that the rates of all sessions with max-min rates less than or equal to p^i converge, i.e.,

$$\lim_{t\to\infty} r_s(t) = r_s^* \ \forall s \text{ s.t. } r_s^* \leq p^i.$$

Let $\tilde{\mathcal{S}}^i$ be the set of all such sessions i.e.,

$$\tilde{\mathcal{S}}^i = \{s :\ s \in \mathcal{S}\ r_s^* \leq p^i\}$$

For all $l \in \mathcal{L}(k)$, define

$$\tilde{f}_l^i(t) = \sum_{s\in\mathcal{S}_l\backslash\tilde{\mathcal{S}}^i} \max(\mu_s, \min_{j\in\mathcal{L}_s} \eta_j(t))$$

and let

$$\tilde{\epsilon}_l^i(t) = \sum_{s\in\mathcal{S}_l\cap\tilde{\mathcal{S}}^i} (r_s^* - r_s(t))$$

Note that

$$\lim_{t\to\infty} \tilde{\epsilon}_l^i(t) = 0\ \forall l \in \mathcal{L}(k)$$

Let

$$\tilde{C}_l^i = C_l - \sum_{s\in\mathcal{S}_l\cap\tilde{\mathcal{S}}^i} r_s^*$$

Now over all $l \in \mathcal{L}(k)$ we get the differential equations

$$\dot{\eta}_l\ (t) = \lim_{\Delta\to 0} \frac{\left[\eta_l(t) + \Delta(\hat{C}_l^i - \tilde{f}_l^i(t) + \tilde{\epsilon}_l^i(t))\right]_0^{C_l^{max}} - \eta_l(t))}{\Delta} \tag{25}$$

Proceeding in a manner similar to the previous case, observe that, if for large enough t, $\eta_l(t) > \mu_u$ for all $l \in \mathcal{L}_u^j,\ j \leq k$, then

$$\begin{aligned} \tilde{f}_l^i(t) + \tilde{\epsilon}_l^i(t) &> \tilde{C}_l^i \\ \Rightarrow \dot{\eta}_l\ (t) &< 0 \end{aligned}$$

Hence we can conclude that

$$\begin{aligned} \limsup_{t\to\infty} \min_{l\in\mathcal{L}_u} \eta_l(t) &\leq \mu_u \\ \Rightarrow \lim_{t\to\infty} r_u(t) &= \mu_u \end{aligned}$$

□

Proof of Theorem IV.1 The asymptotic evolution of the stochastic approximation algorithm can be shown to be identical to the evolution of the associated differential equation [11]. Hence it is sufficient to show that the trajectories of the associated ODE yield the max-min allocation. Note that by an inductive application of Lemma A.3, we can show that

$$\begin{aligned} \lim_{t\to\infty} r_s(t) &= r_s^* \ \forall s \in \mathcal{S} \\ \Rightarrow \lim_{t\to\infty} C - f(\eta(t)) &= 0 \\ \Rightarrow \lim_{t\to\infty} \eta(t) &= \eta^* \quad \text{because } \eta^* \text{ is unique} \end{aligned}$$

Note that the initial point can arbitrary as long as it is within the rectangle $\Pi_{l\in\mathcal{L}}[0, C_l^{max}]$. Hence we now apply Theorem 2.3.1 in [11] to conclude that the sequence $\eta_l(k)$ from Algorithm IV.1 converges to η^*.

□

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*, pp 498-502

[3] L. Benmohamed and S.M. Meerkov, "Feedback Control of Congestion in Store-and-Forward Networks: The Case of Multiple Congested Nodes", Control Group Report No. CGR-94-0-6, The University of Michigan, March 1994.

[4] A.Benveniste, M.Metivier and P.Priouret, *Adaptive Algorithms and Stochastic Approximations*, Springer Verlag, Berlin-Heidelberg,1990

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

[6] Anna Charny, "An algorithm for rate allocation in a packet-switching, network with feedback," Master's thesis, MIT, Cambridge, May 1994.

[7] C. Fulton, San-Qi Li and C.S. Lim, "UT: ABR Feedback Control with tracking", *Preprint.*

[8] H.P. Hayden. "Voice flow control in integrated packet networks," Master's thesis, MIT, Cambridge, October 1981.

[9] R. Jain, S. Kalyanraman, R. Viswanathan and R. Goyal, " A Sample Switch Algorithm", *ATM Forum/95-0178,* February 1995.

[10] A. Kolarov and G. Ramamurthy, "A Control Theoretic Approach to the Design of Closed Loop Rate Based Flow Control For High Speed ATM Networks", Proc. IEEE *INFOCOM'97*

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

[12] J. Mosely, "Asynchronous Distributed Flow Control Algorithms," PhD thesis, MIT, Cambridge, October 1984.

[13] Danny H. K. Tsang, Wales Kin Fai Wong, "A New Rate-Based Switch Algorithm for ABR Traffic to Achieve Max-Min Fairness with Analytical Approximation and Delay Adjustment" Proc. IEEE *Globecom'96*

[14] Nanying Yin and M.G. Hluchyj, "On Closed-Loop Rate Control for ATM Cell Relay Networks", Proc. IEEE *INFOCOM'94*, 1994, pp 99-108.