On a Distributed Stochastic Approximation Approach for Max-Min Fair Rate Control of Flows in Packet Networks *

Santosh P. Abraham† and Anurag Kumar
Dept. of Electrical Communication Engg.
Indian Institute of Science, Bangalore 560 012, INDIA
e-mail: sabraham@qualcomm.com, anurag@ece.iisc.ernet.in

December 13, 2004

Abstract

We consider a distributed stochastic approximation algorithm that computes max-min fair rate allocations to several *elastic* flows sharing a network (an elastic flow is one that can adapt its sending rate to the rate that the network can provide it). The flows are assumed to traverse a fixed sequence of links in the network. The available capacities at the network links are modeled as stochastic processes. Each session can request a *minimum rate guarantee*, hence we work with a notion of max-min fairness with minimum rates. A major part of this paper is the proof that the rate allocation computed by the stochastic approximation iterations converges to max-min rate.

1 Introduction

This paper is about a distributed flow control algorithm for the following scenario. Several *elastic* sessions s ($\in \mathcal{S}$) share a network comprising links l ($\in \mathcal{L}$) with *stochastic* available capacities. Elastic sessions essentially comprise file transfers which can adapt their sending rates to the rate that the network can provide them (unlike, for example, real-time voice transfers). Each session follows a fixed route, and may share the links on that route with other sessions. The problem is to compute the rates ($r_s, s \in \mathcal{S}$) at which the sessions can transmit, so as to achieve a *fair* sharing of the available bandwidth. We constrain the form of the algorithm to capture the following requirements.

- Should be based on distributed and asynchronous computations at the various nodes: The fair shares to be allocated to the sessions depend on the network topology, the link capacities and the

*Submitted to Automatica. This paper is based on research supported by a grant from Nortel Networks.

†Corresponding author. Currently with Qualcomm Inc. , San Diego, CA, USA

routes taken by the sessions. In practice, no central entity knows the actual routes taken by the sessions since routes are themselves discovered in a distributed manner. Thus, a central entity cannot compute the fair rate allocations. Hence, of necessity, the computation of the fair rates has to be performed in a distributed (and asynchronous) manner. In practice, we cannot assume a globally synchronising clock at which the nodes carry out their computations.

- Should be able to work in the face of communication delays between the nodes: Owing to the finite speed of propagation of signals in the network links, there are delays in control information flow between the network nodes, and between the network nodes and the session sources.
- Should be computationally simple: A network node in a high speed network would carry hundreds to thousands of sessions. An algorithm that does not require per-session state to be maintained, and per session computations to be done, would certainly be preferable.

In addition, the rates allocated to the sessions are constrained to remain above a certain minimum value for each session ($\mu_s, s \in \mathcal{S}$).

The fair rate allocation we consider is *max-min* fairness (MMF). The max-min fair paradigm captures the dual goals of "equitable" sharing and maximal utilisation. In a max-min fair allocation no session is allowed to increase its rate if this increase will require another current session with a smaller or equal rate to further reduce its rate. In communication networks max-min fair allocation was proposed in the context of speech transmission with variable rate coding. A good textbook treatment is available in [8]. In order to accommodate the minimum rate requirements of the sessions, we use a generalized notion of max-min fairness which preserves the "lexicographical maximum" property of max-min fairness (see [1] for details).

The design of distributed algorithms for max-min fair rate allocation has received much attention in existing literature. Distributed max-min fair sharing algorithms were discussed in earlier literature in the context of speech transmission with variable rate coding (Hayden [13], Mosely [26]). Hayden's algorithm consisted of a simple additive successive approximation update at each node. Mosely [26] extended Hayden's algorithm to accommodate asynchronous updates and network delays. A similar class of algorithms (ERICA [15], UT [11]) have been proposed in more recent literature in the context of the Available Bit Rate (ABR) service in Asynchronous Transfer Mode (ATM) networks; these approaches use multiplicative successive approximation updates at each node.

In other literature [10, 16, 28] several authors have presented methods for implementing a centralized max-min fair sharing algorithm [8] in a distributed manner. Another approach has been the use of control theoretic methods. The pioneering paper in this direction is by Benmohamed and Meerkov [6]. Kolarov and Ramamurthy [19] extended the idea of Benmohamed et al. [6] by using dual controllers in order to provide good steady state and good transient performance. In [7], Benmohamed and Wang extend the control theoretic formulation of [6] to the case where per-flow queueing is available and apply the control method to each flow's queue.

However, in all the above literature, changes in the capacity available to the elastic sessions is assumed to be constant over long time scales. However, in multimedia networks where the total capacity is shared among applications that have short time scale variations in transmission rates (such as encoded video), the available capacity for elastic traffic is subject to such rapid fluctuation. Our model departs from the traditional approaches in that we have incorporated these short time scale variations owing to the variations of the link capacity utilisation by the higher priority (real-time) sessions. We do this by modelling the available capacity at each link as a random process. However, this processes at the various links are assumed to be stationary and ergodic. The desired max-min fair allocation of rates to sessions is calculated with respect to some statistic of the capacity process at each link; e.g., a fraction of the mean.

Our presentation in this paper is divided into three steps. We begin by reviewing an extension of the conventional max-min fair notion (see [8] and [12]) to accommodate minimum rate requirements. This extension to max-min fair allocation was also independently proposed in the ABR context by [19] and [14]. We then motivate and propose the stochastic approximation based distributed algorithm. The remaining paper is devoted to a proof that the rate allocations computed by the algorithm indeed converge to their respective max-min fair values. The proof involves two major steps. The first step is to show that the stochastic approximation iterations converge to the solution of a certain differential equation; this can be shown via standard arguments, and we only sketch the approach. The second step is to show that the solution of this differential equation corresponds to the desired max-min fair solution. This proof is presented in detail and forms the main contribution of this paper.

In this paper we limit ourselves to the presentation of the theoretical aspects of the algorithm. In a companion paper [5] (and in other references therein) we have studied certain implementation aspects of the algorithm, and have used simulations to study how the algorithms performs on an example network.

This paper is organised as follows. In Section 1.1 we set down some notation that will be used throughout the paper. In Section 2 we review the theory of max-min fair allocation with non zero minimum rate requirements. In Section 3, we motivate the stochastic approximation approach. In Sections 4 and 5, we present the stochastic approximation algorithm and prove the convergence of the computed rates via an ODE approach. We close with some final remarks in Section 6. The proofs of some Lemmas have been relegated to the Appendix.

1.1 The Network Model and Notation

In this section we present the notation used

- If A is a set, then $\mid A \mid$ denotes the size of, or the number of elements in A. For sets A and B, $A \backslash B$ denotes $A \cap B^c$. ϕ denotes the empty set.
- 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.

The following is a list of specific symbols that we have used

$\mathcal{S}$ the set of sessions

$\mathcal{L}$ the set of links

$C_l(t)$ the stationary and ergodic stochastic process of the capacity of the link $l \in \mathcal{L}$

C_l a statistic of the process $C_l(t)$, $l \in \mathcal{L}$; for example, a scaled mean, e.g., $C_l = 0.95E[C_l(t)]$. More sophisticated statistics can be used; for example the Effective Service Capacity which is derived from a target tail behaviour of the link buffer (see [5]). *The max-min fair allocation is sought with respect to this statistic at each link.*

$\mathcal{C}$ 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 $l \in \mathcal{L}$

n_l the number of sessions through link $l \in \mathcal{L}$

r_i the rate of the ith session, $1 \leq i \leq \mid \mathcal{S} \mid$; $r = (r_1, r_2, \ldots, r_{|\mathcal{S}|})$ denotes the *rate vector*

μ_s the minimum rate for session $s \in \mathcal{S}$

$\mathcal{M}$ the set $\{\mu_s : s \in \mathcal{S}\}$

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

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

Note that the 4-tuple $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$ characterises an instance of the bandwidth sharing problem. Thus we will say, for example, that the rate vector r is feasible for $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$, or that r is the max-min fair rate vector for $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$, etc.

2 Max-Min Fair Bandwidth Sharing with Nonzero MCRs- A Brief Review

In this section we present a review of the theory of max-min allocation for the case where sessions have non-zero minimum rate requirements. We adopt the generalisation of the notion of MMF rate allocation that is defined in [12].

Definition 2.1 *We call a rate vector r* **feasible** *for the problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$ if*

$$\text{for all } s \in \mathcal{S}, \quad r_s \geq \mu_s, \text{ and for all } l \in \mathcal{L}, \quad f_l(r) = \sum_{s \in \mathcal{S}_l} r_s \leq C_l.$$

Note that the set of feasible vectors is non-empty iff $\forall l \in \mathcal{L}$

$$\sum_{s \in \mathcal{S}_l} \mu_s \leq C_l$$

We will assume that this is so, *with strict inequality*, in all the following discussions. Note that such feasibility will be ensured by an admission control procedure for connections using the best effort service.

Definition 2.2 *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 2.3 *Given a rate vector r, a link l is said to be a* **bottle-neck link** *for a session j if*

(i) link l is saturated, i.e., $f_l(r) = C_l$, and

(ii) 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

$$r_s \leq \max(\mu_s, r_j)$$

The following theorem gives two equivalent characterisations of the max-min rate vector.

Theorem 2.1 *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.

For a proof see [1]. We note here that with max-min fair flow rates while every session has a bottleneck link, *not every link is a bottleneck for some session.*

2.1 Max-Min Fair Allocation as the Solution of a Vector Equation

We now show that the max-min fair rates of a problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$ can be obtained from a solution of a certain vector equation. This perspective motivates our approach to the design of distributed algorithms for computing the max-min fair share. We associate with each link $l \in \mathcal{L}$ a number η_l which we call the link control parameter (LCP). The rate r_s of any session s through link l, is bounded by $\max(\mu_s, \eta_l)$. Obtaining the max-min fair rate vector can then be equivalently stated as a problem of obtaining all the LCP's. We now show that a desired (not necessarily unique) vector of link control parameters is a solution of a certain vector equation.

Theorem 2.2 *For the max-min fair bandwidth sharing problem $(\mathcal{L}, \mathcal{C}, \mathcal{S}, \mathcal{M})$, if a set of LCP's $(\eta_l, l \in \mathcal{L})$ and set of links $\tilde{\mathcal{L}}$ satisfy*

$$\begin{aligned} \min_{j \in \mathcal{L}_s} \eta_j &= \min_{j \in \mathcal{L}_s \backslash \tilde{\mathcal{L}}} \eta_j \quad \text{for all } s \in \mathcal{S} \\ \sum_{s \in \mathcal{S}_l} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j) &= C_l \quad \text{for all } l \in \mathcal{L} \backslash \tilde{\mathcal{L}} \end{aligned}$$

then the rate vector $(r_s, s \in \mathcal{S})$ defined as

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

is max-min fair

Proof: From a centralized algorithm (see [1]) it is clear that such a vector $(\eta_l,\ l \in \mathcal{L})$ exists. By Theorem 2.1 it is sufficient to show that with $r_s, s \in \mathcal{S}$ as defined in the theorem, every session $s \in \mathcal{S}$ has a bottleneck link. Consider any $s \in \mathcal{S}$. Let $l_s \in \mathcal{L}_s \backslash \tilde{\mathcal{L}}$ [1] be such that

$$\eta_{l_s} = \min_{j \in \mathcal{L}_s} \eta_j$$

The link l_s is saturated, by hypothesis (ii) of the theorem. Also

$$r_s = \max(\mu_s, \eta_{l_s})$$

It follows that, $\forall q \in \mathcal{S}_{l_s}$,

$$r_q = \max(\mu_q, \min_{j \in \mathcal{L}_q} \eta_j) \leq \max(\mu_q, \eta_{l_s}) \leq \max(\mu_q, r_s)$$

Hence by Definition 2.3, l_s is a bottleneck link for $s \in \mathcal{S}$.
□

Consider the case in which $\tilde{\mathcal{L}}$ is empty, i.e., every link is a bottleneck for at least one session. Define a vector function $\underline{f}(\underline{\eta}) = (f_l(\underline{\eta}),\ l \in \mathcal{L})$ with $f_l(\underline{\eta}) = \sum_{s \in \mathcal{S}_l} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j) \quad \forall l \in \mathcal{L}$. For each value of $\underline{\eta}$, $f_l(\underline{\eta})$ is just the total flow in link l. Then by Theorem 2.2, the max-min allocation can be obtained by solving

$$\underline{f}(\underline{\eta}) = \underline{C} \tag{1}$$

For each value of $\underline{\eta}$, $f_l(\underline{\eta})$ is just the total flow in link l. In this study we restrict ourselves to networks where the solution to equation 1 is unique.

3 Motivating the Use of Stochastic Approximation

It is clear from the discussion in Section 2 that the max-min fair solution is a global solution across a network. Hence, computing the max-min fair solution requires information of all links and sessions in the network. An algorithm that requires complete information of the network is clearly a centralized algorithm (see Section 5.1), and would not be of much value for implementation. However, Theorem 2.2 can be used to motivate a distributed framework in which instances of the distributed algorithm run independently at each link, using only information local to the link, such as link capacity and total flow through the link.

Note, from Section 2.1, that the solution of a vector equation framework essentially expresses two requirements:

(i) Given the LCP's, $\eta_l, l \in \mathcal{L}$, the rate of a session is computed by $r_s = \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j)$

(ii) The LCP's are to be chosen so that the total rate into a link (which is a bottleneck for some session) is equal to the available capacity of the link.

[1] $\mathcal{L}_s \backslash \tilde{\mathcal{L}}$ is not empty because every session has at least one bottleneck link.

We assume that the best-effort service control mechanism in the network includes special control packets corresponding to each session (an example is the Resource Management (RM) cells in ATM networks). These control packets have data fields that can be used for conveying information between the end-points and the network. Control packets are emitted by the source of each session, can be examined and modified by the switches, and then returned back to the source by being "turned around" by the destination end-point of the session.

The first requirement mentioned above is then easily achieved in the following way: when emitting a control packet, a source places a large value in the appropriate field (e.g., the maximum rate at which the source wishes to emit); as the control packet travels along the route of the session, at each link the minimum of the value in the field and the links's LCP is taken and placed back in the field. When the packet reaches the destination end-point the rate field contains the value $\min_{j \in \mathcal{L}_s} \eta_j$ (for the case of session s).

The second requirement is achieved by each link iteratively adjusting its LCP based on a comparison of the total rate into the link and the link capacity. Theorem 2.2 can be used to motivate a stochastic approximation approach as follows.

Let us first consider a *single* link with N sessions and constant capacity C. The max-min fair allocation is obtained by solving the following equation for η

$$C - \sum_{s=1}^{N} \max(\mu_s, \eta) = 0$$

Since $C > \sum_{s=1}^{N} \mu_s$, the above equation has a unique solution η^*. The session rates are then given by: for $1 \leq s \leq N$,

$$r_s = \max(\mu_s, \eta^*)$$

An iteration proposed by Hayden in [13] for solving this problem is

$$\eta(k+1) = \eta(k) + \frac{1}{N}\left(C - \sum_{s=1}^{N} \max(\mu_s, \eta(k))\right)$$

and it is easily shown that $\eta(k) \to \eta^*$.

In a more general network, the above algorithm can be independently implemented at every link. Thus we would have at a link l

$$\eta_l(k+1) = \eta_l(k) + \frac{1}{N_l}\left(C_l - \sum_{s \in \mathcal{S}_l} r_s(k)\right) \tag{2}$$

The session rates $r_s(k)$ in the above distributed solution can be shown to converge as long as

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

i.e., the sources update their rates in lock step with the links. Simply applying the above algorithm to a real world network would not yield the desired results due to the following reasons:

1. **Stochastic Available Capacity:** Until now we have assumed that the available capacity (for elastic sessions) at link l is a fixed number C_l, and that we seek the MMF rate vector for this problem. As discussed in the Introduction, however, the point of departure of our work is that available capacity at each link is not a fixed number but is random, since the bandwidth requirement of the stream sessions at each link is random. Hence C_l needs to be some statistic of the random available capacity. A naive choice for C_l would be the mean of the stochastic available capacity. It is clear from elementary queueing theory that when the total input rate for a stochastic server is equal to the mean of the stochastic service rate, then queue lengths increase to infinity. A simple alternative would be to choose C_l to be a fraction (say 0.95) of the mean. The choice of such a scaling factor is not clear as the queue length process depends on the higher moments of the stochastic service process. Another approach is to choose C_l to be the total *input* rate that ensures that queue lengths are constrained. We have proposed a large deviations theory based formulation for obtaining such a value in [5] and have called this the Equivalent Service Capacity (ESC).

2. **Asynchronous Updates**: It is clear that in a pactical network it would not be possible to synchronise the computations at the nodes. Also, sessions have different (and possibly random) round trip times, so it would not be possible to guarantee that sessions update their rates in lock step with LCP calculation. Even if we had a fixed link capacities, a naive application of Hayden algorithm without accounting for the asynchrony in the network would lead to non-convergent and potentially unstable iterations.

   Note that if we view $\left(C_l - \sum_{s \in \mathcal{S}_l} r_s(k)\right)$ as a cost function that we seek to minimise at each link using the distributed algorithm, then in a practical network the above cost function takes the form

$$C_l + \omega_l(k) - \sum_{s \in \mathcal{S}_l} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j(k - \tau_{lj}^s(k))))$$

   whre the parameter $\tau_{lj}^s(k)$ is the random delay that a session experiences in obtaining the LCP information from a link and $\omega_l(k)$ denote the random variations in the available link capacity. We are thus faced with the problem of designing a distributed algorithm where the ”cost function” in the individual iterations are computed based on random quantities such as the random link rates. We propose the use of a stochastic approximation algorithm based on a Robbins-Munro type iteration[21]. Based on recent work by Borkar [9], we note that this above approach also effectively addresses the problem of random delays in updating the session rates. The main idea is to use the simple the iteration given in 2 with a decreasing gain instead of the fixed gain of $1/n_l$.

Relationship to the control of elastic traffic in the Internet: While the original motivation for the work presented in this paper was the ABR service in ATM networks, there are interesting relationships with recent research on TCP (Transmission Control Protocol) controlled bandwidth sharing in the Internet. In an internet, bandwidth sharing is achieved by TCP's end-to-end adaptive window mechanism. Recent research has focussed on formulating the fair bandwidth sharing problem as one of maximising the total session utility (each session's utility is a function of rate provided to the session) subject to network capacity constraints on the session rates. It has been argued in [18] and in [23] that TCP achieves "proportional" fairness. In some simple situations (simple topologies or a single bottleneck for each route) proportional fairness is equivalent to max-min fairness. Further, TCP can be viewed as a whole class of adaptive window based algorithms, and the bandwidth sharing that is achieved depends on the network feedback and source adaptation; see [25]. With this point of view, it can be seen that with appropriate feedback from routers, TCP can actually achieve max-min fairness. New versions of TCP permit routers to mark a special bit in each passing packet in the forward path, and these bits are then fed back to the session sources by their corresponding sinks. Thus a session's source can determine the probability of bit marking in the forward path of the session. Now suppose that the link computations in the network proceed exactly as proposed in our paper. Each link computes its LCP η_l, and marks packets passing through it with probability $e^{-\alpha\eta_l}$, for some $\alpha > 0$. For large α it follows that the total marking probability along route s is approximately $e^{-\alpha \min_{l\in\mathcal{L}_s} \eta_l}$ (see also [24]). Thus, knowing α, each source on route s effectively learns $\min_{l\in\mathcal{L}_s} \eta_l$; it can then measure the round trip time along the path and adjust its window accordingly (see also [17]).

In this context, the relationsip between our algorithm and the one in [24] is also interesting. In [24] the solution of the dual of the utility maximisation problem reduces to the problem of matching the flows into a link to the link's available capacity, just as in our solution of the max-min fair sharing problem. If this leads to the saturation of a link then that link's congestion price (the dual variable; see [24]) is positive, else the congestion price is 0. It follows from the discussion in the previous paragraph, that $e^{-\alpha\eta_l}$ can be viewed as a congestion "price", which is zero for an unsaturated link, where η_l can be taken to be ∞. In [24] a constant gain algorithm is proposed, whereas in our present paper and in [5] we study a decreasing gain algorithm. The decreasing gain algorithms have the added advantage of being able to provide provable convergence even with asynchronous updates and arbitrary delays. To take care of large changes in the available capacities we have proposed a gain resetting approach in [3].

4 Asynchronous Distributed Stochastic Approximation Algorithms

In this section we shall present the distributed stochastic approximation algorithm and analytically prove that the session rates obtained converge to the max-min fair value. The algorithm considered has the same flavor as Hayden's algorithm, i.e., an additive increment/decrement is performed at each iteration, albeit with decreasing gain. However, to precisely define the algorithm so that the asynchronous operation is correctly captured and the frame work corresponds rigorously to [9] we develop the following notation and state some required assumptions.

4.1 Some Key Assumptions

To capture the effect of asynchrony between nodes we assume the existence of a global clock. At each tick of the global clock, one of the links is selected and its link control parameter is updated. The link selection process is random and denoted by the stochastic process $\psi(k) \in \mathcal{L},\ k = 1, \ldots, \infty$. Note that the existence of a global clock is an artifice introduced for the convenience in indexing the updates at each node. No relation between the ticks of this global clock and wall clock time is assumed except for causality, i.e. increasing order of global clock ticks index events that occurred in non decreasing order of wall clock time.

Assumption 4.1 *The sampling process ψ_k is assumed to have the following property*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} I\{\psi_i = l\} \geq \Delta \textit{ for some } \Delta > 0$$

At a link l, the rate of a session s, just prior to update $k+1$, (denoted by $r_{sl}(k)$) is given by

$$r_{sl}(k) = \max(\mu_s, \min_{j\in\mathcal{L}_s} \eta_j(k - \tau^s_{lj}(k)))$$

The delay in feedback is accounted for by the parameter $\tau^s_{lj}(k)$,i.e., $k - \tau^s_{lj}(k)$ gives the clock tick at which the LCP of link j used to compute $r_{sl}(k)$ was computed. Let the rate at the session source at clock tick k be given by

$$r_s(k) = \max(\mu_s, \min_{j\in\mathcal{L}_s} \eta_j(k - \tau^s_j(k)))$$

$k - \tau^s_j(k)$ gives the clock tick at which the LCP of link j used to compute $r_s(k)$ was computed.

Since the "global-clock" is an indexing artifice, note that the case of links updating simultaneously in wall clock time in the real network is easily taken care of by considering the updates to take place at different "global clock" ticks.

Assumption 4.2 *There exists $D < \infty$ such that for all $s \in \mathcal{S}$, for all $l, j \in \mathcal{L}_s$, and for all $k = 1, \ldots, \infty$, $\tau^s_{lj}(k) \in \{0, 1, \ldots, D-1\}$.*

The capacity of link l at clock tick $k+1$ is denoted by $C_l(k)$.

Assumption 4.3 *$C_l(k)$ can be written as*

$$C_l(k) = C_l + \omega(k) + \beta(k)$$

where C_l is a constant and

$$\lim_{k\to\infty} \beta(k) = 0$$

Let $\mathcal{F}_k$ denote the following σ-field.

$$\mathcal{F}_k = \sigma\{\eta(m)\ m \leq k,\ \tau^s_{lj}(m) m < k,\ C_l(m)\ m < k\ \ \psi_m,\ m \leq k\}$$

We assume that $\omega(k)$ is bounded and

$$E(\omega(k) \mid \mathcal{F}_k) = 0$$

The sequence $\omega(k)$ captures the random variations in the capacity. The term $\beta(k)$ has been introduced to keep the discussion general enough to handle the case where a converging estimate of the capacity is used, e.g., a long term average of the instantaneous measurements may be used. The sequence $\beta(k)$ can also be stochastic.

Along the lines of [9], with each link $l \in \mathcal{L}$, we associate a sequence of gains $a(l, i),\ l \in \mathcal{L},\ i = 0, 1, \ldots, \infty$ with the following properties.

Assumption 4.4

$$\sum_{i=0}^{\infty} a(l,i) = \infty$$

Assumption 4.5

$$\sum_{i=0}^{\infty} a^2(l,i) < \infty$$

Assumption 4.6 *For* $0 < x \leq 1$

$$\lim_{n\to\infty} \frac{\sum_{k=0}^{xn} a(j,k)}{\sum_{k=0}^{n} a(j,k)} = 1 \ \ \forall l \in \mathcal{L}$$

Assumption 4.7

$$\lim_{n\to\infty} \frac{\sum_{k=0}^{n} a(i,k)}{\sum_{k=0}^{n} a(j,k)} = a_{ij}$$

At each link l, update we use the next member of the gain sequence $a(l,i)$. Hence if link l is updated at tick $k+1$, the gain parameter used (denoted by $\overline{a}(l,k)$) is given by

$$\overline{a}(l,k) = a(l, \sum_{i=0}^{k} I\{\psi_i = l\})$$

Note that $\sum_{i=0}^{k} I\{\psi_i = l\}$ is the number of updates at link l that have taken place up to and including the tick $k+1$.

4.2 Link Control Parameter Update Expression

The link parameter update expressions at each link is as follows (let $[x]_a^b = min(b, max(x,a))$).

$$\eta_l(k+1) = \left[\eta_l(k) + \overline{a}(l,k)\left(C_l(k) - \sum_{s\in\mathcal{S}_l} \max(\mu_s, \min_{j\in\mathcal{L}_s} \eta_j(k - \tau_{lj}^s(k)))\right) I\{\psi_k = l\}\right]_0^{C_l^{max}} \quad (3)$$

We now state the key result of this paper. However, in order to complete the rigorous proof of convergence for the seesion rates we need the following assumption on the sessions and network

Assumption 4.8 *The vector of LCPs η^* for which the max-min fair rate is obtained is unique, i.e., there exists exactly one $\eta^* = (\eta_l^*, l \in \mathcal{L})$ such that*

$$r_s^* = min_{l\in\mathcal{L}_s} \eta_l^*$$

Theorem 4.1 *Given Assumptions 4.1 4.2 4.3 4.4 4.5 4.7 and 4.8, the sequence of session rates computed by update Equation 3 converges to the max-min fair rates, i.e.,*

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

1. As in the case of proofs of stochastic approximation algorithms [21], the first part of the proof is to show that the asymptotic evolution of the sequence of link control parameters $\eta_l(k)$ is equivalent to the solution of a certain ordinary differential equation (ODE). We denote the solution of this ODE by $\eta_l(t)$ (by an abuse of notation) [2]. The detailed proof of the existence of the ODE and the equivalence of the evolution of the ODE and the algorithm iterations uses standard technical arguments available in the literature [22] [9]. We will only motivate and display the form of the differential equation below; details of this part of the proof are available in [4]. The Assumptions 4.1 - 4.7 can also be used to show that the asymptotic evolution is not affected by the asynchrony and delays in the updates; see [9] for a detailed proof. The essential idea is the following. By Assumption 4.2 the oldest iterate used in a given iteration is no more than D steps old. The "gain" in the stochastic approximation iteration can be viewed as a time increment over which the ODE is being integrated. Since the gains (integration step sizes) are decreasing, and the oldest iterate being used in an iterate is at most D steps old, the iterates are using previous values that less and less older in time. Hence with decreasing gains the update delays cease to matter.

2. The second part consists of showing that the differential equation has a steady state solution, and the rates corresponding to the steady state solution of the differential equation are the max-min rates. This part of the proof is specific to the MMF problem with minimum rate guarantees. The proof is interesting as it depends on the way the centralized MMF computation evolves. We will present the details of this proof in this paper.

We now motivate the form of the ODE. Assume a completely synchronous framework. There are L links whose LCPs are updated at the instants $v_1 = v_2 = \cdots = v_L < v_{L+1} = v_{L+2} = \cdots = v_{2L} < \cdots$. Let $\underline{\eta}(k)$ denote the link control parameter vector, and let $f_l(\underline{\eta}(k))$ denote the total flow through link l, i.e.,

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

Let, for $1 \leq l \leq L, k \geq 1$, the stochastic approximation gains be given by $a(l,k) = \frac{1}{k}$. Then the synchronous update at link l would be written as follows. Suppose the mth update of η_l takes place at v_k. Then

$$\begin{aligned}
\eta_l(k) &= \left[\eta_l(k-1) + \frac{1}{m}(\hat{C}_l(k-1) + \omega_l(k-1) - f_l(\underline{\eta}(k-1))\right]_0^{C_l^{max}} \\
\eta_l(k+1) &= \eta_l(k) \\
\cdots &= \cdots \\
\eta_l(k+(L-1)) &= \eta_l(k+(L-2))
\end{aligned}$$

[2] Throughout this paper use $\eta_l(.)$ with the arguement *u,v,t* to denote the solution of the ODE and $\eta_l(.)$ with argument *k,l,m,n* for the discrete iterations.

¿From the above steps we obtain

$$\eta_l(k+(L-1))-\eta_l(k-1) = \left[\eta_l(k-1)+\frac{1}{m}(\hat{C}_l(k-1)+\omega_l(k-1)-f_l(\underline{\eta}(k-1)))\right]_0^{C_l^{max}} - \eta_l(k-1)$$

Viewing the decreasing gains $a(l,k)$ as steps in "time", the rate of change of $\eta_l(k)$ can be written as

$$\begin{aligned}
&\frac{\eta_l(k+(L-1))-\eta_l(k-1)}{\frac{L}{m}} \\
&= \frac{\left[\eta_l(k-1)+\frac{1}{m}(\hat{C}_l(k-1)+\omega_l(k-1)-f_l(\underline{\eta}(k-1)))\right]_0^{C_l^{max}} - \eta_l(k-1)}{\frac{L}{m}} \\
&= \frac{1}{\frac{L}{m}}\left(\left[\eta_l(k-1)+\frac{L}{m}\frac{1}{L}(C_l+\beta_l(k-1)+\omega_l(k-1)-f_l(\underline{\eta}(k-1)))\right]_0^{C_l^{max}} - \eta_l(k-1)\right)
\end{aligned}$$

It can be shown that as $k \to \infty$, the sequences $\omega_l(k)$ and $\beta_l(k)$ have diminishing effect on the sequence $\eta_l(k)$. It can also be shown that the propagation delays and the asynchrony have a diminishing effect (see [9]). It can then be shown to follow that as k increases to infinity, the limiting behaviour of the sequence is given by the following set of differential equations (see also [21] and [22]). For $l \in \mathcal{L}$

$$\overset{\bullet}{\eta}_l(t) = \lim_{\Delta \to 0} \frac{\left[\eta_l(t)+\Delta\frac{1}{L}(C_l - f_l(\underline{\eta}(t))\right]_0^{C_l^{max}} - \eta_l(t)}{\Delta} \tag{4}$$

To formally state the differential equation, in general, we need to define the following parameters $\gamma_l^*, l \in \mathcal{L}$. With reference to the parameters $a_{ij}, i, j \in \mathcal{L}$, defined in Assumption 4.7, for $l \in \mathcal{L}$, define

$$\gamma_l^* = \frac{1}{\sum_{j\in\mathcal{L}} a_{jl}}$$

Notice that for the simple case discussed above, $a_{jl} = 1, \forall j, l$, and hence $\gamma_l^* = \frac{1}{L}$.

Finally, the differential equation can be shown to be of the following form. For each $l \in \mathcal{L}$

$$\overset{\bullet}{\eta}_l(t) = \lim_{\Delta \downarrow 0} \frac{[\eta_l(t)+\Delta\gamma_l^*(C_l - f_l(\eta(t)))]_0^{C_l^{max}} - \eta_l(t)}{\Delta} \quad \forall l \in \mathcal{L} \tag{5}$$

In the next section we shall prove that the session rates obtained from the differential Equation 5 are the desired max-min fair rates for the bandwidth sharing problem.

5 Steady State Solution of the Differential Equation

Let the $\eta^* = (\eta_l^*, l \in \mathcal{L})$ denotes the unique (by Assumption 4.8) link control parameter vector that correspond to r^*; i.e.,

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

The main result we shall prove in this section is that the differential Equation 5 has a steady state solution, and the session rates obtained from this steady state solution are max-min fair. We state the result precisely in the following theorem.

Theorem 5.1 *Consider the differential Equation 5. Let the rate allocation to a session $s \in \mathcal{S}$ be given by $r_s(t) = \max(\mu_s, \min_{l \in \mathcal{L}_s} \eta_l(t))$. Then the following hold. For all $s \in \mathcal{S}$,*

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

and, for all $l \in \mathcal{L}$,

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

In order to prove Theorem 5.1 we require a special partition of the set of sessions $\mathcal{S}$. The details of this partition are given in the following subsection.

5.1 A Partition on the Set of Sessions

We first present a centralized algorithm for computing the max-min allocation. The centralized algorithm yields a certain partition on the set of links and sessions. We deduce an alternate set of partitions required for the proof of Theorem 5.1 from the partition obtained by the centralized algorithm. For a proof that the rate allocation yielded by the centralized algorithm is the max-min solution see [1].

Algorithm 5.1

The iterations are indexed by $k, k \geq 1$. At the end of the kth iteration, the following variables are defined

$r(k)$**:** rate vector after k^{th} iteration

$\mathcal{S}(k)$**:** the set of unbottlenecked sessions

$\mathcal{L}(k)$**:** the set of unsaturated links

$f_l(r(k))$**:** the total flow in link l when $r(k)$ is the rate vector

$F_l(k)$**:** the total flow in link l due to the bottlenecked sessions

$n_l(k)$**:** the number of unbottlenecked sessions through link l, (i.e., $n_l(k) = \mid \mathcal{S}_l \cap \mathcal{S}(k) \mid$)

$\eta_l(k)$**:** result obtained by distributing the residual capacity of link l (after removing flow due to bottlenecked sessions) among the unbottlenecked sessions on that link; for links with no unbottlenecked sessions, $\eta_l(k) = \eta_l(k-1)$.

Initialisation:
$k = 0, \mathcal{S}(0) = \mathcal{S}, \mathcal{L}(0) = \mathcal{L}$, and $\forall l \in \mathcal{L}, n_l(0) = \mid \mathcal{S}_l \mid, \gamma_l(0) = 0, \eta_l(0) = 0, F_l(0) = 0.$

While $\mathcal{S}(k)$ is not empty, do steps 1 to 7

1. $k \leftarrow k + 1$
2. Calculate $\eta_l(k)$

   if $l \in \mathcal{L}(k-1)$ and $\mathcal{S}_l \cap \mathcal{S}(k-1) \neq \phi$, then compute $\eta_l(k)$ by solving

$$C_l - F_l(k-1) - \sum_{s \in \mathcal{S}_l \cap \mathcal{S}(k-1)} \max(\eta_l(k), \mu_s) = 0$$

   Otherwise

$$\eta_l(k) = \eta_l(k-1)$$

3. Compute the rate of each unbottlenecked session $s \in \mathcal{S}(k-1)$

$$r_s(k) = \max(\mu_s, \min_{l \in \mathcal{L}(k-1)} \eta_l(k))$$

   For the sessions $s \in \mathcal{S} \backslash \mathcal{S}(k-1)$, i.e., the bottlenecked sessions

$$r_s(k) = r_s(k-1)$$

4. Calculate the new total flow through each link $l \in \mathcal{L}$.

$$f_l(\eta(k)) = \sum_{s \in \mathcal{S}_l} \max(\mu_s, , \min_{l \in \mathcal{L}(k-1)} \eta_l(k)) = \sum_{s \in \mathcal{S}_l} r_s(k)$$

5. Find the new set of unsaturated links.

$$\mathcal{L}(k) = \{l : f_l(r(k)) < C_l\}$$

6. Find the new set of unbottlenecked sessions; these are the sessions all of whose links are in $\mathcal{L}(k)$.

$$\mathcal{S}(k) = \{s : \mathcal{L}_s \subseteq \mathcal{L}(k)\}$$

7. Find the flow in each link $l \in \mathcal{L}$ due to the bottlenecked sessions.

$$F_l(k) = \sum_{s \in \mathcal{S}_l \backslash \mathcal{S}(k)} r_s(k)$$

□

For the centralized Algorithm 5.1, let M be the number of iterations until the termination of the algorithm, i.e., when every session has at least one bottleneck link. The following sequence of sets are obtained from the execution of the algorithm.

$\mathcal{S}(k)$: the set of sessions that are not bottlenecked at any link after iteration $k = 0, 1, \ldots, M$. Note that $\mathcal{S}(0) = \mathcal{S}$ and $\mathcal{S}(M) = \phi$ and $\mathcal{S}(0) \supset \mathcal{S}(1) \ldots \supset \mathcal{S}(M)$.

$\mathcal{L}(k)$: the set of links that are not bottlenecks to any session for any links just after iteration $k = 0, 1, \ldots, M$. Note that $\mathcal{L}(0) = \mathcal{L}$ and $\mathcal{L}(M)$ is the set of all links that are not bottlenecks for any session; further $\mathcal{L}(0) \supset \mathcal{L}(1) \ldots \supset \mathcal{L}(M)$.

In the execution of the centralized algorithm, at every iteration one or more links become bottlenecks for the sessions through them. We now consider the link control parameters computed at the links when they become bottlenecks. Let $p^0 = 0$, $p^{M+1} = \infty$ and for $k = 1, \ldots, M$, let p^k denote the minimum of the link control parameters computed at iteration k of the centralized algorithm. Note that every link at which p^k is computed at the k^{th} iteration becomes a bottleneck link at the k^{th} iteration. p^k is obtained as follows. For every link $l \in \mathcal{L}(k-1)$, at the k^{th} iteration we solve for x_l such that

$$C_l - F_l(k-1) - \sum_{s \in \mathcal{S}_l \cap \mathcal{S}(k-1)} \max(x_l, \mu_s) = 0$$

Recall that $F_l(k-1)$ is the flow of all the sessions that have been bottlenecked at iterations prior to the k^{th} iteration of the centralized algorithm ($F_l(0) = 0$). Compute p^k by

$$p^k = \min_{l \in \mathcal{L}(k-1)} x_l$$

Define the following sequence of subsets of $\mathcal{L}$ and $\mathcal{S}$

$\mathcal{L}^k$: the set of links at which p^k is the obtained value of $\eta_l(k)$ at the k^{th} iteration of the centralized algorithm, i.e.,

$$\mathcal{L}^k = \{l : \eta_l(k) = \min_{j \in \mathcal{L}(k-1)} \eta_j(k) = p^k\}$$

Note that every link in $\mathcal{L}^k$ becomes a bottleneck link at the k^{th} iteration and thus

$$\mathcal{L}^k = \mathcal{L}(k-1) \backslash \mathcal{L}(k)$$

$\mathcal{S}^k$: the set of sessions that are bottlenecked at the k^{th} iteration.

It is important to note that for all $s \in \mathcal{S}^k$, if $\mu_s \leq p^k$, then $r_s = p^k$, otherwise $r_s = \mu_s$.

Using the notation developed above for the centralized algorithm, we define the following alternate partition. The proof of convergence that follows uses an induction argument on this partition.

$$\hat{\mathcal{S}}^i = \{s \in \mathcal{S} : p^i \leq r_s^* < p^{i+1}\}$$
$$\hat{\mathcal{S}}^{M+1} = \{s \in \mathcal{S} : \ \mu_s > p^M\}$$

We also define, for $m \geq 0$,

$$\hat{\mathcal{S}}(m) = \mathcal{S} \setminus \cup_{i=1}^{m} \hat{\mathcal{S}}^i$$

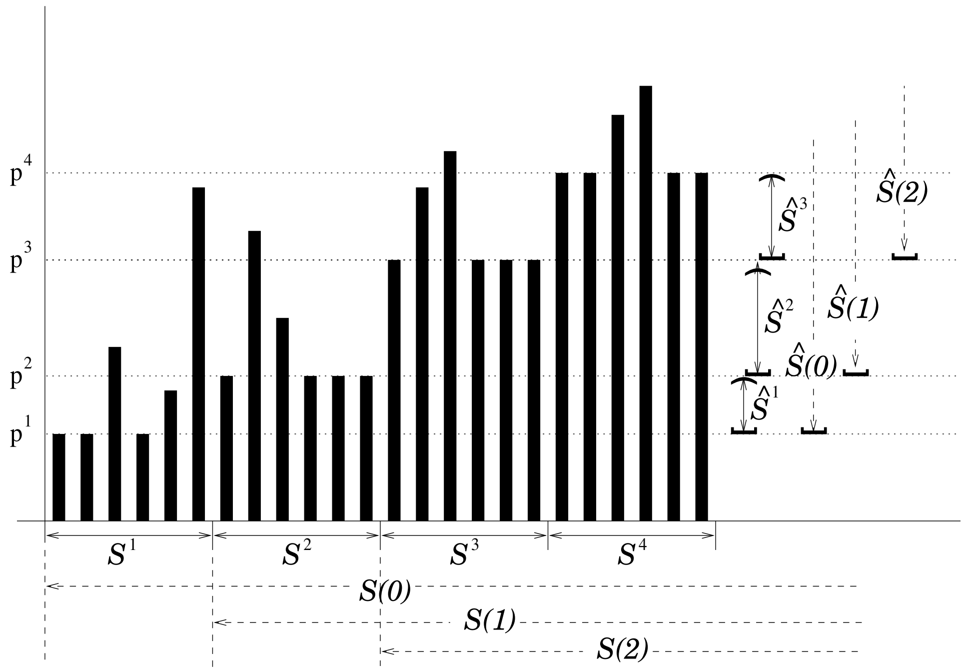

Figure 1: The relationship between the partition of sets obtained from the centralized algorithm and the alternate partition. Each vertical bar corresponds to a session. The sessions are ordered from left to right according to the sequence in which they get bottlenecked. The height of each session's bar is its max-min fair rate. In the alternate partition, the sessions are partitioned according their max-min fair rates; a session belongs to the "highest" interval $\hat{\mathcal{S}}^k$ in which its max-min fair rate lies. The sets $\mathcal{S}(k)$ and $\hat{\mathcal{S}}(k)$ are also shown.

$$\hat{\mathcal{S}}_l(m) \ = \ \mathcal{S}_l \setminus \cup_{i=1}^{m} \hat{\mathcal{S}}^i$$

It is clear, from the italicised comment following the definition of S^k, that without MCR requirements (i.e., $\mu_s = 0, \forall s$), $\hat{\mathcal{S}}^i = \mathcal{S}^i, \ i = 1, \ldots, M$. In general, the relationship between the partition $\hat{\mathcal{S}}^i, i = 1, \ldots, M$ and the partition $\mathcal{S}^i, i = 1, \ldots, M$ is depicted in Figure 1. The "non-hat" partition is according to the sequence in which sessions gets bottlenecked during the centralized algorithm, whereas the "hat" partition is based upon an ordering of the max-min fair rates. Notice that a session that is bottlenecked at iteration 1, but at its MCR $> p^2$, does not belong to $\hat{\mathcal{S}}^1$; for example, in the figure, the third session from the left got bottlenecked at step 1 of the centralized algorithm but this session is in $\hat{\mathcal{S}}^2$.

5.1.1 Proving Theorem 5.1

The proof will follow by induction on the session partition $\hat{\mathcal{S}}^i$. We begin by stating two lemmas, namely Lemmas 5.1 and 5.2. Lemma 5.1 is about the steady state solution of the differential equation when the network consists of a single link only. Lemma 5.2 is a general result on the ordering of the solutions of differential equations. These two lemmas will be used in Lemma 5.3 to show that the link control parameters are asymptotically lower bounded. The lower bounding subsequently leads to one step of the induction, i.e., the rates of the sessions in a given partition $\hat{\mathcal{S}}^i$ converge to the max-min fair value if the rates of the sessions in the partitions $\hat{\mathcal{S}}^j,\ j < i$, converge to the max-min fair value. The proof will follow by an inductive application of Lemma 5.3.

Lemma 5.1 *Consider a single link network with link capacity C, and let $\mathcal{U}$ denote the set of sessions. Given the (one dimensional) differential equation*

$$\overset{\bullet}{x}(t) = \lim_{\Delta \downarrow 0} \frac{[x(t) + \Delta(C - \sum_{s\in\mathcal{U}} \max(\mu_s, x(t)) + \epsilon(t))]_0^{C^{max}} - x(t)}{\Delta}$$

with $\epsilon(t)$ continuous and

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

Let the initial condition $x(t_0) \in [0, C^{max}]$. Then

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

where x^* solves

$$C - \sum_{s\in\mathcal{U}} \max(\mu_s, x^*) = 0$$

Proof: See the Appendix.

Lemma 5.2 *Let $x(t), y(t) \in \Re^n$. Consider the two differential equations*

$$\overset{\bullet}{x}(t) = g_1(x(t), t) \quad (6)$$

$$\overset{\bullet}{y}(t) = g_2(y(t), t) \quad (7)$$

The functions g_1 and g_2 are continuous in both arguments. For $1 \le l \le n$, $g_{2l}(y(t), t)$ denotes the lth component of $g_2(y(t), t)$. In addition, the following hold.

(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) $g_2(\cdot, \cdot)$ *is nondecreasing in its first argument.*

(iii) For all $u \in \Re^n$, $g_1(u, t) \ge g_2(u, t)$.

Under the above conditions, if $x(t_0) = y(t_0)$, then for all $t \in (t_0, \infty)$ $x(t) \ge y(t)$.

Proof: See the Appendix.

Lemma 5.3 *Consider $i, 1 < i \le M$. If for all $s \in \hat{\mathcal{S}}^j, j < i$,*

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

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

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

Proof: The proof of Lemma 5.3 is divided into three parts. In Part 1 we consider the links $l \in \mathcal{L}(i-1)$. For all links in $l \in \mathcal{L}(i-1)$, 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. In Part 2 we consider the following subset of $\hat{\mathcal{S}}^i$.

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

$\tilde{\mathcal{S}}^i$ is the set of all sessions that get bottlenecked at the i^{th} iteration of the centralized algorithm with max-min fair rate p^i. The bottleneck links of the sessions in $\tilde{\mathcal{S}}^i$ are in $\mathcal{L}^i$. We show that given the lower bounding argument on the link control parameters, the rate of the sessions $s \in \tilde{\mathcal{S}}^i$ converge to the max-min fair share. We then show that the rates of the rest of the sessions in $\hat{\mathcal{S}}^i$, i.e., $s \in \hat{\mathcal{S}}^i \backslash \tilde{\mathcal{S}}^i$ also converge to their max-min fair share.

Part 1: Consider the set $\mathcal{L}(i-1)$. For each $l \in \mathcal{L}(i-1)$, define the following.

$$\epsilon_l^i(t) = \sum_{s \in \mathcal{S}_l \backslash \hat{\mathcal{S}}_l(i-1)} (r_s^* - r_s(t)) \tag{8}$$

Note that $\mathcal{S}_l \setminus \hat{\mathcal{S}}_l(i-1) = \mathcal{S}_l \cap \left(\cup_{j=1}^{i-1} \hat{\mathcal{S}}^j\right)$. Thus $\epsilon_l^i(t)$ is the "error" in the flow of sessions that are assumed, by the hypothesis, to have flows that converge. Further define

$$\hat{C}_l^i = C_l - \sum_{s \in \mathcal{S}_l \backslash \hat{\mathcal{S}}_l(i-1)} r_s^* \tag{9}$$

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

$$f_l^i(\eta(t)) = \sum_{s \in \hat{\mathcal{S}}_l(i-1)} r_s(t) = \sum_{s \in \hat{\mathcal{S}}_l(i-1)} \max(\mu_s, \min_{j \in \mathcal{L}_s} \eta_j(t)) \tag{10}$$

$f_l^i(\eta(t))$ is the sum of the flows in link l of all sessions whose flows have not been assumed to converge in the hypothesis of this theorem. With the definitions in Equations (8), (9) and (10), for each $l \in \mathcal{L}(i-1)$ we can rewrite the differential equation (4) as

$$\overset{\bullet}{\eta}_l(t) = \lim_{\Delta \downarrow 0^+} \frac{\left[\eta_l(t) + \Delta\gamma^*(\hat{C}_l^i - f_l^i(\eta(t)) + \epsilon_l^i(t))\right]_0^{C_l^{max}} - \eta_l(t))}{\Delta} \tag{11}$$

Let $\eta^i(t) = (\eta_l(t), l \in \mathcal{L}(i-1))$. To obtain a lower bound on $\eta^i(t)$, we consider the auxiliary differential equation

$$\overset{\bullet}{x}_l(t) = \lim_{\Delta \downarrow 0^+} \frac{\left[x_l(t) + \Delta\gamma^*\left(\hat{C}_l^i - \sum_{s \in \hat{\mathcal{S}}_l(i-1)} \max(\mu_s, x_l(t)) + \epsilon_l^i(t)\right)\right]_0^{C_l^{max}} - x_l(t)}{\Delta} \tag{12}$$

and let $x^i(t) = (x_l(t), l \in \mathcal{L}(i-1))$. Observe that this ODE (12) corresponds to a single link problem in which the link capacity is obtained by subtracting the flows of all the sessions whose rates are assumed to have converged and whose session set includes only those sessions whose flows are not yet known to converge. It is thus a candidate for the application of Lemma 5.1. Also observe that

$$\hat{C}_l^i - \sum_{s \in \hat{\mathcal{S}}_l(i-1)} \max(\mu_s, u_l(t)) \leq \hat{C}_l^i - \sum_{s \in \hat{\mathcal{S}}_l(i-1)} \max(\mu_s, \min_{j \in \mathcal{L}_s} u_j(t))$$

Hence for a link l, given identical initial conditions for Equations (11) and (12), the conditions of Lemma 5.2 are satisfied with $g_1(\cdot,\cdot)$ corresponding to Equation (11) and $g_2(\cdot,\cdot)$ corresponding to Equation (12) and we have, for all t,

$$\eta_l(t) \geq x_l(t) \ \ \forall t \tag{13}$$

Recall that $\mathcal{L}(i-1) = \mathcal{L}^i \cup \mathcal{L}(i)$. Using Lemma 5.1, note that $x_l(t)$ converges for all $l \in \mathcal{L}(i-1)$, and also

$$\lim_{t\to\infty} x_l(t) = p^i \ \ \forall l \in \mathcal{L}^i \tag{14}$$

Let

$$\lim_{t\to\infty} x_l(t) = q_l > p^i \ \ \forall l \in \mathcal{L}(i) \tag{15}$$

¿From Equations (13), (14), and (15) we can conclude that

$$\liminf_{t\to\infty} \eta_l(t) \quad \geq p^i \qquad \forall l \in \mathcal{L}(i-1) \tag{16}$$

and hence

$$\liminf_{t\to\infty} r_s(t) \quad \geq \quad \max(\mu_s, p^i) \ \ \forall s \in \mathcal{S}(i-1) \tag{17}$$

Part 2: Now consider $u \in \tilde{\mathcal{S}}^i$. This is the set of all sessions that get bottlenecked at the i^{th} iteration of the centralized algorithm and have a max-min value of p^i. Hence $r_u^* = p^i$ $(\geq \mu_u)$. By Equation (17)

$$\liminf_{t\to\infty} r_u(t) \geq p^i \tag{18}$$

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 $\psi > 0$ such that

$$\psi < \min\left(\min_{l \in \mathcal{L}_s} \frac{C_l^{max} - p^i}{2\overline{n}}, \min_{l \in \mathcal{L}(i-1)\setminus\mathcal{L}^i} \frac{q_l - p^i}{2\overline{n}}\right) \tag{19}$$

Note that

$$\begin{aligned} p^i + \overline{n}\psi \quad &< \quad q_l - \overline{n}\psi \ \ \forall l \in \mathcal{L}(i-1) \setminus \mathcal{L}^i \\ p^i + \overline{n}\psi \quad &< \quad C_l^{max} - \overline{n}\psi \ \ \forall l \in \mathcal{L}_u^i \end{aligned}$$

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

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

Given (16) note that we can choose T large enough so that for all $t > T$, we have

$$\begin{aligned} \eta_l(t) &> p^i - \xi_1 \;\; \forall l \in \mathcal{L}^i \\ \eta_l(t) &> q_l - \overline{n}\psi \;\; \forall l \in \mathcal{L}(i-1) \setminus \mathcal{L}^i \\ | \epsilon_l^i(t) | &< \xi_2 \end{aligned}$$

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

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

Hence note that $\overset{\bullet}{\eta}_l(t_1) < 0$ for all $l \in \mathcal{L}_u \cap \mathcal{L}^i$. 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 5.1) that, $\forall\, t > t_2$, $r_u(t) < p^i + \overline{n}\psi$. Since ψ is positive and can be arbitrarily small, we have shown that for all sessions $u \in \tilde{\mathcal{S}}^i$:

$$\limsup_{t\to\infty} r_u(t) \leq p^i \tag{20}$$

Now using Equation (18) and (20) we conclude that

$$\lim_{t\to\infty} r_u(t) = p^i \tag{21}$$

Part 3: Now consider those sessions with $u \in \hat{\mathcal{S}}^i \backslash \tilde{\mathcal{S}}^i$ These sessions can have bottlenecks at multiple levels. The potential bottle neck links lie in links in $\mathcal{L}^1, \ldots, \mathcal{L}^i$. Since the session rate is at least its MCR, note that

$$\liminf_{k\to\infty} r_s(k) \geq \mu_s \tag{22}$$

Hence for these sessions it is sufficient to show that

$$\limsup_{k\to\infty} r_s(k) \leq \mu_s \tag{23}$$

Let

$$\tilde{\mathcal{L}}_s = \mathcal{L}_s \cap \cup_{j=1}^i \mathcal{L}^j$$

$\tilde{\mathcal{L}}_s$ contains all the potential bottlenecks for the session s. In Case 1, note that we considered $\mathcal{L}_s \cap \mathcal{L}^i$, since all the bottlenecks for the sessions considered lay in there. Now we use $\tilde{\mathcal{L}}_s$ instead of $\mathcal{L}_s \cap \mathcal{L}^i$. In Case 1, we also required for every link in $l \in \mathcal{L}(i)$ that $\liminf_{t\to\infty} \eta_l(t) > p^i$. Similarly for the present case we require to show that for every $l \in \mathcal{L}_s \cap \mathcal{L}(i)$, $\liminf_{t\to\infty} \eta_l(t) \geq \mu_s$, this is where the following lemma is required.

Lemma 5.4 *Consider $l \in \mathcal{L}(m)$, let $s \in \mathcal{S}_l \cap \mathcal{S}^j, j < m+1$, and $p^m < \mu_s < p^{m+1}$. Assume for all $u \in \mathcal{S}_l$ with $r_u^* < \mu_s$, $r_u(k) \to r_u^*$, then*

$$\liminf_{t\to\infty} \eta_l(t) > \mu_s$$

Proof: The argument required is similar to that in Part 1 of the above proof and is given in the Appendix.

Continuing the Proof of Lemma 5.3: Now we shall show that for $u \in \hat{\mathcal{S}}^i \backslash \tilde{\mathcal{S}}^i$, given an arbitrary $\epsilon > 0$, for large enough t, $r_u(t) < \mu_u + \epsilon$.

Let u_1, u_2, be an ordering of $\hat{\mathcal{S}}^i \backslash \tilde{\mathcal{S}}^i$ in ascending order of MCRs, i.e. $\mu_{u_1} \leq \mu_{u_2} \dots$. Let $r_{u_m}(t) \to r^*_{um} = \mu^*_{um}$ for $m = 1, \dots, j-1$. Then for all $l \in \mathcal{L}(i)$, using Lemma 5.4, we have

$$q_l = \liminf_{t \to \infty} \eta_l(t) > \mu_{u_j}$$

Choose ψ such that

$$\liminf_{t \to \infty} \eta_l(t) > \mu_{u_j} + (\overline{n} + 1)\psi$$

Consider $l \in \tilde{\mathcal{L}}_s$, for $s \in \mathcal{S}_l$, and $s \neq u_j$, choose T large enough so that for $t > T$, we have

$$r_s(t) > r^*_s - \psi$$

Note that this is possible due to the hypothesis of the lemma, Part 1, and the fact that every session rate is at least its MCR. Hence for all $l \in \mathcal{L}^s$,

$$C_l - \sum_{s \in \mathcal{S}_l} r_s(t) \;=\; C_l - r_{u_j}(k) - \sum_{s \in \mathcal{S}_l, s \neq u_j} r_u(k) \;<\; C_l - (\mu_{s_j} + (\overline{n}+1)\psi) - \sum_{s \in \mathcal{S}_l, s \neq u_j} (r^*_s - \psi) \;<\; -\psi$$

Hence there exists some $\epsilon > 0$ such that for all $l \in \tilde{\mathcal{L}}_s$ $\dot{\eta}_l\,(t) < \epsilon$. Thus $r_{u_j}(t)$ will continue to decrease till $r_{u_j}(t) < \mu_{u_j} + \overline{n}\psi$. Now consider $t_2 > T$ such that $r_{u_j}(t) < \mu_{u_j} + \overline{n}\psi$. It can now be shown (using an argument similar to the argument in Lemma 5.1) that, $\forall\ t > t_2$, $r_{u_j}(t) < \mu_{u_j} + \overline{n}\psi$. Since ψ is positive and can be arbitrarily small, we have shown that for u_j. sessions $u \in \tilde{\mathcal{S}}^i$:

$$\limsup_{t \to \infty} r_{u_j}(t) \leq \mu_{u_j}$$

Now apply the above argument inductively on $u_1, u_2, \dots.$ to conclude that

$$\limsup_{t \to \infty} r_u(t) \leq \mu_u \quad \forall u \in \hat{\mathcal{S}}^i \backslash \tilde{\mathcal{S}}^i$$

□

Proof of Theorem 5.1: Note that by an inductive application of Lemma 5.3 we can show that

$$\lim_{t \to \infty} r_s(t) = r^*_s \;\; \forall s \in \mathcal{S} \tag{24}$$

By Assumption 4.8 and Equation 24 .

$$\lim_{t \to \infty} \eta_l(t) = \eta^*_l$$

Proof of Theorem 4.1: The sequence $\eta_l(k), l \in \mathcal{L}$, generated by the update Equation 3 asymptotically imitates the evolution of the differential Equation 5. By Theorem 5.1, we note that the steady state solution of Equation 5 yields the max-min fair rate. Hence the update Equation 3 yields a rate sequence that converges to the max-min fair rate.

6 Final Remarks

In this paper we have presented an asynchronous distributed algorithm for max-min fair bandwidth allocation to elastic sessions in a packet network. The algorithm is robust to short term variations in the available capacity. We have use a distributed stochastic approximation iteration. The main contribution of this paper was to show that the limit mean ODE for the stochastic approximation iteration has the max-min fair solution as its stable point. The algorithm involves an extremely simple iterative step and does not require explicit inter-node communication. Also, the algorithm does not require any per flow rate computation, since the LCP computation is common to all sessions in a link.

The simple form of the iteration at each node ensures that the proposed algorithm would be very simple to implement. However, the decreasing gains of the stochastic approximation algorithm cause poor adaptation to sudden long term changes in available capacity, for example due to the entry and exit of real-time sessions (such as voice and video calls). When such changes occur, the gain of the algorithm should be reset to a large value. A few simple techniques for detecting such changes have been proposed and studied via simulations in [5].

An important issue to note is that the convergence of the algorithm is dependent, in part, on the bounded delay between the calculation of a link control parameter at a link and its effect at the source. In a practical implementation, a node in which the available capacity has dropped suddenly (in its random evolution) may experience a surge in the queue length of packets and may lead to dropped packets. In such cases a mere count of arriving packets at a node may not be an accurate measure of the sending rate of the sources. To ameliorate this problem there are several solutions.

1. Using a conservative estimate of the available link capacity can reduce the possibility of large queue build ups. A straight forward choice would be some fraction of the mean of the link capacity process. Another choice of the link capacity would the effective service capacity (ESC) discussed in [5] where the capacity is calculated (based on large deviation theory) such that the buffer length remains below a given target queue with very high probability. In addition all sources could be required to start at a very low rate. This would reduce the possibility of large queue build ups during the progress of the algorithm.

2. In the context of ATM networks with ABR traffic, the RM cells may be forwarded through nodes with a higher priority than the undelying data. Since the RM cells are introduced at the source at fixed packet count intervals, the rate of arrival of the RM cells can be used to determine the rate of data sources. Also the ATM RM cells themselves contain the source sending rates and the rates may be read of the RM cells

3. Similarly, if the packets of a flow contain sequence numbers and the packet sizes are fixed, then the sequence numbers may be used to determine the sending rate of the source.

A large number of decreasing gain sequences would satisfy the requirements of the formulation. However, note that the choice of the decrease step for the gain in the stochastic approximation algorithm would significantly impact the transient performance of the algorithm. Further study into the choice of the gain decrease method needs to be carried out. Investigation into the use of measurable processes such as the queue length and aggregated input flow rate in determining the gain could yield sequences that improve the convergence rate of the algorithm.

References

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

[2] 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.

[3] Santosh P. Abraham and Anurag Kumar, "A Simulation Study of Adaptive Distributed Algorithms for Max-Min Fair Rate Control of ABR Sessions," *Canadian Conference on Broadband Research (CCBR '98)*, Ottawa, Canada, June 1998.

[4] Santosh P. Abraham, *Asynchronous Distributed Rate Control Algorithms for Best-Effort Sessions in Integrated Services Networks with Minimum Rate Guarantees*, Ph.D. Thesis, Dept. of Electrical Communication Engg., Indian Institute of Science, Bangalore, September 1998.

[5] Santosh P.Abraham and Anurag Kumar, "A New Approach for Asynchronous Distributed Rate Control of Elastic Sessions in Integrated Packet Networks," *IEEE/ACM Transactions on Networking*, February 2001.

[6] L. Benmohamed and S.M. Meerkov, "Feedback Control of Congestion in Store-and-Forward Networks: The Case of a Single Congested Node", *IEEE/ACM Trans. on Networking*, vol. 1, No. 6, 1993, pp693-708.

[7] L. Benmohamed and Y.T. Wang, A Control-Theoretic ABR Explicit Rate Algorithm for ATM Switches with Per-VC Queueing *IEEE Infocom'98*, San Francisco, 1998.

[8] D. Bertsekas and R.G. Gallager, *Data Networks*, Prentice Hall, 1987.

[9] V. Borkar, "Asynchronous Stochastic Approximation" *SIAM J. Control and Optim.*, vol. 36, No. 3, May 1998, pp. 840-851

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

[11] C. Fulton, San-Qi Li and C.S. Lim, "UT: ABR Feedback Control with tracking", *Proc. IEEE INFOCOM'97*, April 1994, pp. 806-815.

[12] E. M. Gafni and D.P. Bertsekas, "Dynamic Control of Session Input Rates in Communication Networks," *IEEE Trans. Automatic Control*, Vol AC-29, No. 11, November 1984.

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

[14] Y.T. Hou, H.H.-Y. Tzeng, V.P. Kumar, "On Fair Rate Allocation Policies with Minimum Cell Rate Guarantees for ABR Service in ATM Networks" *Proc. ITC'15*, Washington, D.C., June 1997.

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

[16] L. Kalampoukas, A. Varma, K.K. Ramakrishnan, "An Efficient Rate Allocation Algorithm for Packet-Switched Networks Providing Max-Min Fairness" *Preprint*

[17] Aditya Karnik and Anurag Kumar, "Performance of TCP Congestion Control with Rate Feedback: Rate Adaptive TCP (RATCP)," in *Proc. IEEE Globecom 2000*, 2000.

[18] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Operations Research*, vol. 49, 1998.

[19] A. Kolarov, G. Ramamurthy, "A Control Theoretic Approach to the Design of Closed Loop Rate Based Flow Control For High Speed ATM Networks" *Proc INFOCOM'97*, April 1997, pp. 293-301.

[20] Anurag Kumar,"On the Average Idle Time and Queue Length Estimates in an M/M/1 Queue", *Operations Research Letters*, Vol 12, pp. 153-157, Sept. 1992.

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

[22] H.J. Kushner, G.G. Yin, *Stochastic Approximation Algorithms and Applications* Springer-Verlag 1997.

[23] Steven Low, Larry Peterson, and Limin Wang, "Understanding TCP vegas: A Duality Model," in *Proc. SIGMETRICS 2001*, 2001, paper obtained from http://www.cs.princeton.edu/nsg/publications.html.

[24] Steven H. Low and David E. Lapsley, "Optimization Flow Control-I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, December 1999.

[25] Jeonghoon Mo and Jean Walrand, "Fair End-to-End Window-Based Congestion Control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, October 2000.

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

[27] J. Neveu, *Discrete Parameter Martingales North-Holland Publishing*, 1975.

[28] 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*

Appendix

Proof of Lemma 5.1: Note that due to truncation and $x(t_0) \in [0, C^{max}]$, $x(t) \in [0, C^{max}]$ for all t. If x^* is unique, we have $x^* > \min_{s\in\mathcal{U}} \mu_s$. Choose ϵ_1 and ϵ_2 such that (i) $0 < \epsilon_1 < \epsilon_2$, (ii) $\epsilon_2 < x^* - \min_{s\in\mathcal{U}} \mu_s$. 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

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

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 $x^* \leq \min_{s\in\mathcal{U}} \mu_s$, will solve the Equation (5.1). Then argue similarly to show that for any given $\epsilon > 0$, for t large enough,
$x(t) < \min_{s\in\mathcal{U}} \mu_s + \epsilon$. Hence the proof.
□

Proof of Lemma 5.2 With $x(t_0) = y(t_0)$, suppose that there exists $t_1 > t_0$ such that

$$x(t_1) \not\geq y(t_1) \tag{25}$$

This implies that there exists l such that

$$x_l(t_1) < y_l(t_1) \tag{26}$$

Let $t_2 = \sup\{t_0 < t < t_1 :\ x_l(t) \geq y_l(t)\}$. Then by the continuity of $x_l(t)$ and $y_l(t)$, we have

$$x_l(t_2) = y_l(t_2) \tag{27}$$

Let

$$z(t) = x(t) - y(t) \tag{28}$$

$$\Rightarrow \dot{z}\ (t) = \dot{x}\ (t) - \dot{y}\ (t) = g_1(x(t), t) - g_2(y(t), t) \tag{29}$$

Now

$$z_l(t_1) < 0 \ \text{ and } \ z_l(t_2) = 0 \tag{30}$$

Use the mean value theorem and the fact that $z_l(t)$ is continuously differentiable to conclude that there exists $t_3 \in (t_2, t_1)$ where

$$\dot{z}_l\ (t_3) = \frac{z_l(t_1)}{(t_1 - t_2)} < 0 \tag{31}$$

$$\Rightarrow g_{1l}(x(t_3), t_3) < g_{2l}(y(t_3), t_3) \tag{32}$$

Hence

$$g_{2l}(x_l(t_3)) = g_{2l}(x(t_3)) \leq g_{1l}(x(t_3), t_3) < g_{2l}(y(t_3), t_3) = g_{2l}(y_l(t_3), t_3) \tag{33}$$

Where have used the three given properties *(i), (ii)* and *(iii)* for $g_1(\cdot, \cdot)$ and $g_2(\cdot, \cdot)$. Note that

$$x_l(t_3) \;\; < \;\; y_l(t_3) \tag{34}$$

$$\Rightarrow g_{2l}(x_l(t_3), t_3) \;\; \geq \;\; g_{2l}(y_l(t_3), t_3) \tag{35}$$

which contradicts Equation 33.

□

Proof of Lemma 5.4: First note that for $l \in \mathcal{L}(m)$ and $\mu_s < p^{m+1}$, if x is such that

$$\sum_{u \in \mathcal{S}_l, r_u^* < \mu_s} r_u^* + \sum_{u \in \mathcal{S}_l, r_u^* \geq \mu_s} \max(\mu_s, x) \geq C_l \tag{36}$$

then $x > \mu_s$. We shall show that $\liminf_{k \to \infty} \eta_l(t)$ has this property. Let

$$\epsilon_l(t) \;\; = \;\; \sum_{u \in \mathcal{S}_l, r_u^* < \mu_s} r_u^*$$

$$\hat{C}_l \;\; = \;\; C_l - \sum_{u \in \mathcal{S}_l} r_u^*$$

For link l the differential equation can be written as

$$\overset{\bullet}{\eta}_l(t) = \lim_{\Delta \downarrow 0^+} \frac{\left[\eta_l(t) + \Delta\gamma^*(\hat{C}_l - \sum_{u \in \mathcal{S}_l, r_u^* \geq \mu_s} \max(\mu_s, \min_{j \in \mathcal{L}_u} \eta_j(t)) + \epsilon_l(t))\right]_0^{C_l^{max}} - \eta_l(t))}{\Delta} \tag{37}$$

To obtain a lower bound on $\eta_l(t)$, we consider the auxiliary differential equation

$$\overset{\bullet}{x}_l(t) = \lim_{\Delta \downarrow 0^+} \frac{\left[x_l(t) + \Delta\gamma^* \left(\hat{C}_l - \sum_{u \in \mathcal{S}_l, r_u^* \geq \mu_s} \max(\mu_s, x_l(t)) + \epsilon_l(t))\right)\right]_0^{C_l^{max}} - x_l(t)}{\Delta} \tag{38}$$

Also observe that

$$\hat{C}_l - \sum_{u \in \mathcal{S}_l, r_u^* \geq \mu_s} \max(\mu_s, y_l(t)) \leq \hat{C}_l - \sum_{u \in \mathcal{S}_l, r_u^* \geq \mu_s} \max(\mu_s, \min_{j \in \mathcal{L}_s} y_j(t))$$

Hence for link l, given identical initial conditions for Equations (37) and (38), the conditions of Lemma 5.2 are satisfied with $g_1(\cdot, \cdot)$ corresponding to Equation (11) and $g_2(\cdot, \cdot)$ corresponding to Equation (12) and we have

$$\eta_l(t) \geq x_l(t) \;\; \forall t \tag{39}$$

Now using Lemma 5.1 note that $x_l(t)$ converges and by Equation 36

$$\lim_{t \to \infty} x_l(t) > \mu_s$$

Hence

$$\liminf_{t \to \infty} \eta_l(t) \;\; \geq \;\; \lim_{t \to \infty} x_l(t) \;\; > \;\; \mu_s \qquad \square$$

□
