

## A PROPORTIONALLY-FAIR ALGORITHM FOR LOSS-FREE RATE ALLOCATION TO ELASTIC USERS\*

P. GOUDARZI<sup>1\*\*</sup>, F. SHEIKHOESLAM<sup>2</sup> AND H. SAIDI<sup>2</sup>

<sup>1</sup>Iran Telecom Research Center, Tehran, I. R. of Iran  
Email: pgoudarzi@itrc.ac.ir

<sup>2</sup>Dept. of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, I. R. of Iran

**Abstract**– Proportional fairness criterion, first proposed by Kelly, has outstanding properties in allocating fair rates to network users. For example, it resembles the Jacobson's AIMD method in rate allocation to users, and there exists a well-established stability analysis relating to the stability of the rate allocation algorithm. Kelly's algorithm uses a form of the *scaled gradient ascent* projection method for converging to the equilibrium point. The structure of Kelly's algorithm is such that in some instants of time, the aggregate flow which is passing through a link may exceed the link capacity. In other words, the algorithm is not loss-free. In this paper, we have proposed a novel *time-varying scaled gradient ascent* projection method that, under some assumptions about the link penalty function, the rate allocation algorithm is loss-free in some network topologies. Also, it is shown by simulation that in a general network topology, the proposed algorithm does not have any loss event in comparison with the Kelly's method.

**Keywords** – Proportional fairness, time-varying scaled gradient ascent, penalty function, loss-free rate allocation, elastic traffic, best-effort

### 1. INTRODUCTION

Loss is an undesired phenomenon in any rate allocation algorithm. In the TCP/IP algorithm in the current internet, some bandwidth is consumed in reaction to loss events, which reduce bandwidth efficiency.

Designing loss-free algorithms or algorithms that have the capability of reducing the packet loss is of crucial importance in developing future high speed networks that can guarantee the users' Quality of Service (QoS) requirements.

In this paper, our goal is to provide a simple framework based on deterministic fluid models [1] that can achieve loss-free fair rate allocation to a number of competing elastic users, and in fact, complements the work of Kelly [2] in some aspects.

Stability analysis of dynamic systems is one of the important steps to design a controller or an algorithm [3-5]. We start with the nonlinear programming formulation of a rate allocation problem suggested in [6] from which a penalty function formulation is derived in [2]. In [2], it has been shown that a congestion controller can be designed such that the equilibrium point of the congestion controller is stable and converges to the unique solution of the penalty function form of the nonlinear optimization problem.

In previous works we encounter the concept of *utility function*, which was first defined by S. Shenker [6]. As was discussed by Shenker in [6], the user's *utility function* is a measure of his (her) satisfaction from the allocated rate

Various notions of fairness can be defined in terms of appropriate utility functions [7-9]. While the well-known *max-min* fairness [10] cannot be defined in terms of a single utility function, it can still be defined in terms of a sequence of utility functions [7].

\*Received by the editors January 14, 2002; final revised form September 22, 2005.

\*\*Corresponding author

This paper is organized as follows. In Section 2, we briefly review the method used by Kelly and his colleagues in establishing *proportional fairness* between a set of elastic users and describe why this method is not loss-free. In Section 3, we explain the proposed loss-free algorithm in mathematical terms. Section 4 is devoted to simulation and we compare the proposed method with that of Kelly for a sample and general network topology. Section 5 concludes the paper and motivates interested readers to future research on this active research area.

## 2. RELATED WORKS

Consider a network with a set  $J$  of resources or links and a set  $R$  of users. Let  $c_j$  denote the finite capacity of link  $j \in J$ . Each user has a fixed traffic route  $R_r$ , which is a nonempty subset of  $J$ . We define a zero-one matrix  $A$ , where  $A_{r,j} = 1$  if link  $j$  is in user  $r$ 's traffic route  $R_r$  and  $A_{r,j} = 0$  otherwise.

The user's *utility function* is a measure of satisfaction from the allocated rate. In other words, when the throughput of user  $r$  is  $x_r$ , user  $r$  receives utility  $U_r(x_r)$ . For example, suppose that a user is transferring a file. The per-transfer delay is inversely proportional to the rate he/she receives. Hence, the utility of the user may be measured as a function of the rate. We assume that the utility  $U_r(x_r)$  is an increasing, strictly concave and continuously differentiable function of  $x_r$  over the range  $x_r \geq 0$ . Furthermore, we assume that the utilities are additive so that the aggregate utility of rate allocation  $X = (x_r, r \in R)$  is:  $\sum_{r \in R} U_r(x_r)$ . This is a reasonable assumption since these utilities are those of independent network users. Let  $U = (U_r(\cdot), r \in R)$ ,  $C = (c_j, j \in J)$  and  $\Omega = (\omega_r, r \in R)$  in which the  $\omega_r$  is a positive and constant weighting parameter associated with the user utility function. In the current paper, as in [2], we assume that the user utility is logarithmic and the objective is to solve the following optimization problem:

NETWORK(A,C;  $\Omega$ ) :

$$\begin{array}{ll} \text{Max} & \sum_{r \in R} \omega_r \log(x_r) \\ \text{Subject to} & A^T X \leq C \\ \text{Over} & X \geq 0 \end{array} \quad (1)$$

The Kelly's discrete time algorithm for solving NETWORK(A,C;  $\Omega$ ) is as follows[2]:

$$x_r[t+1] = x_r[t] + k (\omega_r - x_r[t] \sum_{j \in R_r} \mu_j(t)) \quad (2)$$

$$\text{where: } \mu_j(t) = p_j(\sum_{s: j \in R_s} x_s[t]) \quad (3)$$

Where parameter  $k_r$  controls the speed of convergence in Eq. (2) and  $p_j(y)$  is the amount that link 'j' penalizes its input aggregate traffic 'y' and is a nonnegative, continuous and increasing function of its argument. " $x_s[t]$ " in relation (3) represents a typical user's traffic traversing the link "j".

One of the interpretations that is expressed for the above system is that Eqs. (2-3) try to equalize  $\omega_r$  with  $x_r[t]$ .  $\sum_{j \in R_r} \mu_j(t) \cdot p_j(\sum_{s: j \in R_s} x_s[t])$  is in fact a form of penalty function [10] that is used for the constrained optimization problem NETWORK(A,C;  $\Omega$ ).

By Eqs. (2-3) we can see that the unique equilibrium  $x_r^*$  is the solution of the following equation:

$$\omega_r = x_r^* \sum_{j \in R_r} p_j(\sum_{s: j \in R_s} x_s^*) \quad , r \in R \quad (4)$$

Although, Kelly and his colleagues have proved the stability of their algorithm by using a Lyapunov approach, there is no formal or mathematical discussion about the existence of an interval for 'k' parameter in which users can ensure the convergence of their rates in the discrete-time version of the Kelly algorithm (Eq. (2)).

Furthermore, the algorithm structure implies the probability that in some instants of time, the aggregate rate of some users traversing a link exceeds the link capacity [11] i.e. the algorithm is not loss-free. To clarify this, consider the following simple example:

Suppose that the network is composed of a single user whose traffic traverses a single link with finite capacity  $c$ . By considering the first two iterations of Eq. (2), if  $k$  is selected such that  $k > c/\omega$  and by assuming  $x[0]=0$  it can be seen that:

$$x[1] = x[0] + k (\omega - x_r[0] \sum_{j \in R_r} \mu_j(0)) = k\omega > c$$

The rate through the link remains greater than the link capacity until the next iteration of Eq. (2) takes place and this leads to the loss event in the link.

This means that we may have loss in rate allocation and the QoS requirements of rate allocation in future communication networks motivate us in designing a loss-free rate allocation mechanism.

As it is shown in the mentioned example, blind selection of 'k' parameter as is the case in the Kelly's method, can lead to the loss in the rate allocation. Therefore, as it can be verified in the later sections, we try to choose this parameter more intelligently based on some feedback from the core network in each iteration in order to overcome the mentioned shortcoming.

### 3. LOSS-FREE ALGORITHM

As in [11], the main purpose of this type of algorithm is to find a rate allocation strategy in which the aggregate rate of users traversing a link never exceeds the link capacity. One of the algorithms which is used in this paper in order to achieve the loss-free property is using an appropriate and intelligent method for varying 'k' parameter in the Kelly's method with respect to time and based on the most recent knowledge about the user's rate.

It is shown that by using a time-varying version of Eqs. (2-3), in each step of the algorithm, the parameter 'k' changes as a function of  $x_r[t]$  and other recent samples of the users' rates. This selection of the parameter 'k' forces the aggregate rate through each link, to remain less than or equal to the link capacity which leads to a loss-free rate allocation strategy in a fluid-flow based network traffic point of view.

In the following three subsections we will investigate three different scenarios which are the cases of *one user-one link*, *multi user-one link*, and finally the general case of *multi user-multi link* and discuss the probability of the existence of some methods for changing the gain parameter 'k<sub>r</sub>' in the Kelly's algorithm that lead to a loss-free rate allocation in each scenario.

The 1<sup>st</sup> scenario leads to an exact solution for parameter 'k' in each iteration of the relation (2). In the 2<sup>nd</sup> scenario, by designing appropriate network layer protocols or under certain conditions, it is shown that the exact  $k_r$  can be found for each user in each iteration.

In the 3<sup>rd</sup> and general scenario, it is shown in Section 4 by simulation that by adopting an appropriate method, the  $k_r$  parameter can converge to an exact value in a short period of time and the resulting rate assignment benefits from a better convergence rate and is loss-free.

#### a) Single-user and single-link scenario:

In this case, Eq. (2) changes to the following form:

$$x_{n+1} = x_n + k_n (\omega - x_n p(x_n)) \quad (5)$$

in which  $p(\cdot)$  is convex with positive 1<sup>st</sup> and 2<sup>nd</sup> derivatives and  $x_n$  is the rate of the user at iteration 'n'.

We further assume that  $p(\cdot)$  has the simple form of  $p(x) = \tau \cdot x^\gamma$ , in which  $\gamma > 2$  and  $\tau$  is a positive constant.

As it has been shown in [2], another appropriate form for the penalty function  $p(\cdot)$  is:

$$p_j(y) = (y - c_j + \varepsilon_1)^+ / \varepsilon_1^2, \quad j \in J$$

Where,  $\varepsilon_1$  is a very small positive constant and  $c_j$  is the link capacity. But it can be verified in Fig. 1 that, by selecting the form  $p_j(y) = \sigma \cdot (y/c_j)^\gamma$  for the penalty function and the proper selection of  $\sigma$  and  $\gamma$  parameters, the proposed penalty function converges to the ideal one.

In Fig. 1 we have considered a link with normalized capacity 10, and selected parameter  $\sigma = 10000$ . For three different values of  $\gamma$  parameter, the resulting penalty function shapes have been compared with that of Kelly in which we have used  $\varepsilon_1 = 0.0001$  as a good approximation of the ideal penalty function. As it can be verified, by increasing the  $\gamma$  parameter, we can better approximate the ideal penalty function which can be obtained by moving  $\varepsilon_1 > 0$  in the above equation towards zero.

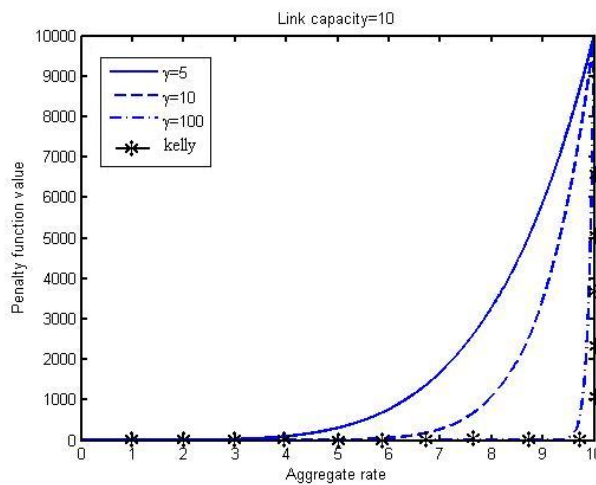


Fig. 1 Comparison of different penalty functions with that of Kelly

As will be shown selecting the simpler form  $p(x) = \tau \cdot x^\gamma$  for the penalty function, simplifies the mathematical computations.

Assume that,  $k_n$  in the rate allocation (5) is selected such that the following equality holds:

$$x_{n+1} p_{n+1} = x_n p_n + \rho (\omega - x_n p_n) \tag{6}$$

where:

$$0 < \rho \leq 1 \quad \text{and} \quad p_n = p(x_n)$$

Then, it is shown in theorem 1 that the resulting rate allocation is loss-free i.e. the allocated rate to the user never exceeds the link capacity (or equilibrium  $x^*$  in the *single user-single link* case as has been discussed by Kelly in [2]).

**Theorem 1:** Consider the following equation:

$$x_{n+1} = x_n + k_n (\omega - x_n p_n)$$

if there exists a real  $k_n$  which causes Eq. (6) to be valid, and further assuming that:

$$0 < \rho \leq 1 \quad \text{and} \quad x_0 p_0 < \omega$$

Then:

- a)  $k_n > 0$
- b)  $x_{n+1} > x_n$

c)  $\omega = x^* p(x^*)$  and  $\lim_{n \rightarrow \infty} x_n = x^*$

**Proof:** From (6), we can write

$$\omega - x_{n+1} p_{n+1} = (1-\rho)(\omega - x_n p_n) \tag{7}$$

By induction we have

$$\omega - x_{n+1} p_{n+1} = (1-\rho)^n (\omega - x_0 p_0) \tag{8}$$

For part (a) from relation (8) and  $0 < \rho \leq 1$  and  $x_0 p_0 < \omega$  we conclude that  $\omega - x_n p_n > 0$ .

The function  $p(\cdot)$  is convex and strictly increasing, and  $\omega - x_n p_n > 0$  and  $0 < \rho \leq 1$ . If we rewrite Eq. (6) in terms of Eq. (5) we have

$$[x_n + k_n (\omega - x_n p(x_n))] p([x_n + k_n (\omega - x_n p(x_n))]) = x_n p_n + \rho (\omega - x_n p_n)$$

From the theorem assumptions, the LHS (Left Hand Side) of the above equation is equal to its RHS (Right Hand Side) only if:  $k_n > 0$ .

For part (b) it suffices to use the relations  $k_n > 0$  and  $\omega - x_n p_n > 0$ .

For part (c), from relation (8), we conclude that:

$$\lim_{n \rightarrow \infty} (\omega - x_n p_n) = 0 \tag{9}$$

By considering Eq. (4), we conclude that:

$$\omega = x^* p(x^*) \quad \text{and} \quad \lim_{n \rightarrow \infty} x_n = x^*$$

We have assumed that the real solution of Eq. (6) with respect to  $k_n$  can be found mathematically, but this assumption is true only for some forms of  $p(\cdot)$ . For example, as we said before, if we consider the form:

$$p(x) = \tau x^\gamma = \sigma (x/c)^\gamma, \quad \gamma > 2 \quad \text{and} \quad \tau, \sigma > 0 \tag{10}$$

then the real solution of Eq. (6) with respect to  $k_n$  can be obtained. It is necessary to say that the form of (10) is not the only form which leads to an obtainable real solution for  $k_n$  and finding the set of functions that lead to such solutions can be considered as an open area. If we assume the form (10) for  $p(\cdot)$  and solve Eq. (6) for  $k_n$ , we can see that

$$k_n = \{ [(1-\rho) x_n^{\gamma+1} + \rho \omega / \tau]^{1/(\gamma+1)} - x_n \} / (\omega - \tau x_n^{\gamma+1}), \quad \forall n. \tag{11}$$

Furthermore, using L'Hospital's rule we have:

$$\lim_{x_n \rightarrow (\omega/\tau)^{1/\gamma+1}} k_n = \rho / [\tau (\gamma+1) (\omega/\tau)^{\gamma/(\gamma+1)}].$$

It can be concluded from this simple case that, in this scenario, by selecting  $k_n$  according to relation (11), a loss-free rate allocation can be achieved.

**b) Multi-user and single-link scenario**

As you can see in Fig. 2, in this case, we assume that 'm' users are traversing a single link and Eq. (5) changes to

$$x_{i,n+1} = x_{i,n} + k_{i,n} (\omega_i - x_{i,n} p(\sum_{j=1}^m x_{j,n})) \tag{12}$$

$$i = 1, \dots, m \quad n = 0, 1, 2, \dots$$

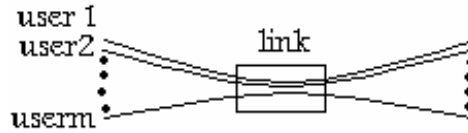


Fig. 2. The scenario in which ‘m’ users are traversing one link

The function  $p(\cdot)$  has the same properties which was discussed in part (3-a) and we have:

$$x_{i,n+1} p_{n+1} = x_{i,n} p_n + \rho_i (\omega_i - x_{i,n} p_n) , i = 1, \dots, m \tag{13}$$

$$\text{where: } p_n = p\left(\sum_{j=1}^m x_{j,n}\right) \quad \text{and} \quad 0 < \rho_i \leq 1$$

We further assume that:  $x_{i,0} p_0 \leq \omega_i , \forall i$ .

As before, our goal is finding a time-varying  $k_{i,n}$  in Eq. (12), such that the resulting  $x_{i,n}$  satisfies Eq. (13) for every  $i$ .

As before, we can write

$$\omega_r - x_{r,n+1} p_{n+1} = (1 - \rho_r) (\omega_r - x_{r,n} p_n) \tag{14}$$

By induction:

$$= (1 - \rho_r)^n (\omega_r - x_{r,0} p_0) , r = 1, 2, \dots, m$$

For simplicity, if we assume,  $k_{i,n} = k_n , \forall i$ , then the following holds:

**Corollary:** If we assume,  $k_{i,n} = k_n , \forall i$ , then Eq. (13) has at least one positive solution  $k_n > 0$  for each  $i$ .

**Proof:** If we replace  $x_{i,n+1}$  from Eq. (12) into Eq. (13), for each  $i$  we have

$$(x_{i,n} + k_n (\omega_i - x_{i,n} p_n)) p\left(\sum_{j=1}^m (x_{j,n} + k_n (\omega_j - x_{j,n} p_n))\right) = x_{i,n} p_n + \rho_i (\omega_i - x_{i,n} p_n) \tag{15}$$

Consider the LHS of the above equation, as we know:  $\omega_i - x_{i,n} p_n > 0 \forall i, n$  and if we pay attention to the properties of  $p(\cdot)$  function (described before), we conclude that the LHS of (15) is a continuous strictly increasing function of  $k_n$ .

On the other hand, we can verify that for  $k_n = 0$ , the LHS of (15) is less than or equal to the RHS of (15) and as  $k_n$  goes to infinity, the LHS of (15) is greater than or equal to the RHS of (15), as LHS is a continuous and increasing function of  $k_n$ , and there exists exactly one positive solution for  $k_n$  in (15).

One of the consequences of the above corollary is the fact that Eq. (6) always has a positive solution for  $k_n$  for any form of penalty function (as far as penalty function satisfies the mentioned conditions).

In reality, the assumption  $k_{i,n} = k_n , \forall i$  may not be true in a distributed network because each user ‘i’ updates its ‘ $k_i$ ’ parameter based on feedback information received from the core network. These feedbacks are user dependent (as can be verified in Eq. (11)) and may be different for different users. Thus we must try to find other strategies to solve (13) for  $k_{i,n}$ . In the sequel, we assume that the link penalty function is in the form (10).

Equation (13) is equivalent to

$$x_{i,n+1} p\left(\sum_{j=1}^m x_{j,n+1}\right) = x_{i,n} p_n + \rho_i (\omega_i - x_{i,n} p_n) , i = 1, \dots, m \tag{16}$$

As we can see

$$\frac{x_{j,n+1}}{x_{i,n+1}} = \frac{x_{j,n} p_n + \rho_j (\omega_j - x_{j,n} p_n)}{x_{i,n} p_n + \rho_i (\omega_i - x_{i,n} p_n)} \quad \forall i, j, n \quad (17)$$

And we can write

$$p\left(\sum_{j=1}^m x_{j,n+1}\right) = p\left(x_{i,n+1} + \frac{x_{i,n+1}}{x_{i,n} p_n + \rho_i (\omega_i - x_{i,n} p_n)} \sum_{\substack{j=1 \\ j \neq i}}^m [x_{j,n} p_n + \rho_j (\omega_j - x_{j,n} p_n)]\right) \quad (18)$$

It is clear from (13) and (18) that if by an appropriate network layer protocol, the information  $\sum_{\substack{j=1 \\ j \neq i}}^m \rho_j \omega_j$  and  $\sum_{\substack{j=1 \\ j \neq i}}^m x_{j,n}$  and  $\gamma$  and  $\tau$  parameters in Eq. (10) are sent to user 'i' from the common link, he (she) can compute its  $k_{i,n}$  coefficient independently as it is discussed in the scenario (3-a).

**c) Multi-user and multi-link scenario**

In the case of a network with general topology, Eq. (6) changes to

$$x_{i,n+1} \lambda_{i,n+1} = x_{i,n} \lambda_{i,n} + \rho_i (\omega_i - x_{i,n} \lambda_{i,n}) \quad (19)$$

where

$$\lambda_{i,n} = \sum_{j \in R_i} p_{j,n}$$

As before, the stability and convergence of (19) can be verified by the following equation:

$$\omega_i - x_{i,n+1} \lambda_{i,n+1} = (1 - \rho_i) (\omega_i - x_{i,n} \lambda_{i,n}) = (1 - \rho_i)^n (\omega_i - x_{i,0} \lambda_{i,0}) \quad (20)$$

In the simple case in which each user's traffic traverses one link we can say:

$$\lambda_{i,n+1} = \sum_{j \in R_i} p_{j,n+1} \left( x_{i,n+1} + \frac{x_{i,n+1} \sum_{\substack{\ell=1 \\ \ell \neq i}}^m [x_{\ell,n} p_{j,n} + \rho_\ell (\omega_\ell - x_{\ell,n} p_{j,n})]}{x_{i,n} p_{j,n} + \rho_i (\omega_i - x_{i,n} p_{j,n})} \right) \quad (21)$$

If we define

$$\xi_{j,n,i} = 1 + \frac{\sum_{\substack{\ell=1 \\ \ell \neq i}}^m [x_{\ell,n} p_{j,n} + \rho_\ell (\omega_\ell - x_{\ell,n} p_{j,n})]}{x_{i,n} p_{j,n} + \rho_i (\omega_i - x_{i,n} p_{j,n})} \quad (22)$$

We can write

$$\lambda_{i,n+1} = \sum_{j \in R_i} p_{j,n+1} = \sum_{j \in R_i} p_j (\xi_{j,n,i} \cdot x_{i,n+1}) \quad (23)$$

If we assume that for all of the links, the penalty function  $p(\cdot)$  is in the form  $p_j(y) = \tau_j \cdot y^\gamma$  in which  $\gamma > 2$  and  $\tau_j > 0$  are constant values, then

$$\lambda_{i,n+1} = \sum_{j \in R_i} \tau_j x_{i,n+1}^\gamma \xi_{j,n,i}^\gamma = x_{i,n+1}^\gamma \sum_{j \in R_i} \tau_j \xi_{j,n,i}^\gamma \quad (24)$$

The term  $\sum_{j \in R_i} \tau_j \xi_{j,n,i}^\gamma$  is independent of  $x_{i,n+1}$  and if we assume that the network can send this value to the end user through the nodes in the user's traffic route, according to relation (20), the user can compute

its corresponding  $k_{i,n} > 0$  and uses this parameter in Kelly's algorithm. This results in a loss-free rate allocation algorithm.

Another special case that we can apply Eqs. (21-24) to a general network topology is the case that we can express the network topology in a hierarchical manner as you can see in Fig. 3. In this figure, the gray areas represent users' subnetworks and the dashed area represents the network backbone. If the network congestion occurs in the backbone, those users that are in a subnetwork and traverse the same route in the backbone, have the same  $\lambda_{i,n}$  parameters that represent the congestion level in the user's traffic route.

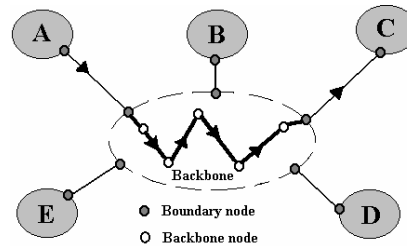


Fig. 3. A hierarchical network topology

In the general case, solving Eq. (19) with respect to 'k' is very difficult, but as it is shown in Section 4, some iterative methods can be used for converging towards the exact solution of 'k'.

As you will see in the next section, we have used an iterative method by which we can converge towards the exact 'k' parameter in a general network topology.

Although we may lose the loss-free property in some occasions because of inaccurate estimation of 'k' in the initial steps of the iterative method, it is shown by simulation that the proposed iterative method has no loss with respect to Kelly's method, at least in the simulated scenarios. The stability and convergence property of the revised method cannot be proved by Eq. (20), therefore these properties must be proved using theorem 2.

**Theorem 2:** Consider the following continuous-time system:

$$\frac{d}{dt} x_r(t) = k_r(X) (\omega_r - x_r(t) \sum_{j \in R_r} \mu_j(t)) \tag{25}$$

where

$$\mu_j(t) = p_j \left( \sum_{s: j \in R_s} x_s(t) \right)$$

We further assume that:

$$k_r(X) > 0 \quad \forall r \in R \tag{26}$$

Then the function:

$$V(X) = \sum_{r \in R} \omega_r \log x_r - \sum_{j \in J} \int_0^{\sum_{s: j \in R_s} x_s} p_j(y) dy \tag{27}$$

is a Lyapunov function for the above system and the vector X maximizing V(X) is a stable point of the system, to which all trajectories converge [2].

**Proof:** See the Appendix.

#### 4. SIMULATION

In this section, we use the sample networks in [11] and [12] to compare the proposed algorithm with that of Kelly. The first network consists of three users with two links as you can see in Fig. 4, and the second and more general topology consists of 20 users and 28 links (Fig. 5).



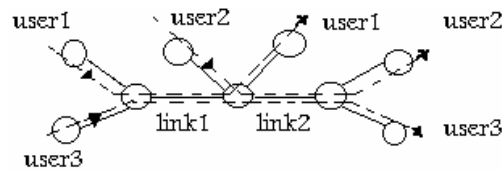


Fig. 4. The first topology with 3 users and 2 links

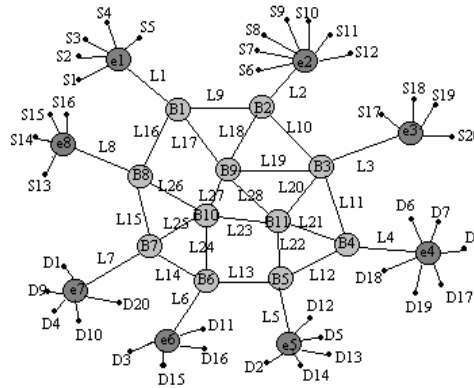


Fig. 5. The second topology with 20 users and 28 links

As we have used the macroscopic fluid-flow traffic model [1] for our purpose and considering the behavior of network queues and packet-level processing of network traffic (microscopic model) are not important for us, the MATLAB® package seems to be a good candidate for performing simulations. With this package, we have simulated the networks and compared the proposed method with that of Kelly.

The strategy for finding ‘k’ is based on the following iteration:

$$k_{i,n+1} = k_{i,n} - \varepsilon \tanh \{x_{i,n} \lambda_{i,n} - [(1 - \rho_i) \cdot x_{i,n-1} \lambda_{i,n-1} + \rho_i \omega_i]\} \quad , \forall i \in R \quad (28)$$

In which,  $\varepsilon > 0$  is some small constant.

The idea behind iteration (28) is that each user tries to follow the exact solution to (19) with one sampling delay and based on the most recent traffic information.

If  $k_{i,n}$  in relation (28) converges for every ‘i’, the resulting  $k_i^*$  is the exact solution to Eq. (19) for every ‘i’.

The important point which must be mentioned here is that, each user for computing his or her corresponding  $k_{i,n}$  parameter, only needs the  $\lambda_{i,n}$  which might be available to him from the network by designing an appropriate network layer protocol. This results in distributed solution [13] for  $k_{i,n}$ . The simulation consists of two parts:

**a) Lack of loss in rate allocation of the Kelly method**

In this part, the ‘k’ parameter for the Kelly method and the initial ‘k’ in the proposed method are selected small enough for the users. So, there is not any loss event in the network.

For topology of Fig.4, we have selected ‘k’ of the Kelly method and the initial ‘k’ of the proposed method in the iteration (28) to be 0.0009.

The normalized link capacity for all links is selected to be 30. We have also used  $\rho_r = 0.01, \forall r \in R$  and  $\varepsilon=0.01$  in relation (28).

We have selected  $\sigma=1$  and  $\gamma=600$  in equation (10).

The users’ utility functions are in logarithmic form with the following parameters:

$$\omega(1) = \omega(2) = 0.5 \text{ and } \omega(3) = 0.3$$

It can be verified in Figs. 6-7 that the rate allocated to users 2 and 3 have a higher convergence rate in the proposed method.

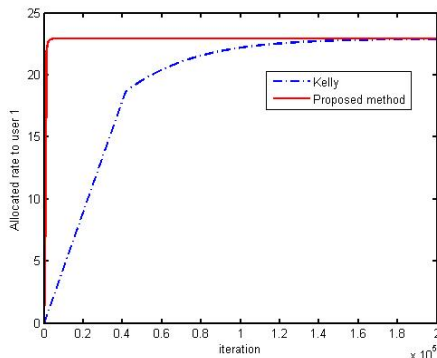


Fig. 6. Comparing rate of user 1 in proposed method with Kelly’s method in the 1<sup>st</sup> topology

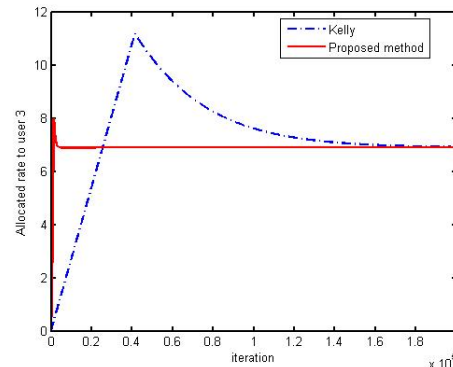


Fig. 7. Comparing rate of user 3 in proposed method with Kelly’s method in the 1<sup>st</sup> topology

The time evolution of ‘k’ parameters in the proposed method for typical users 2 and 3 are depicted in Fig. 8.

The second network topology is depicted in Fig. 5 in which 20 users, denoted: by ‘S’ are traversing a network which consists of 28 links and each user’s destination denoted by ‘D’ letter. The links that are traversed by each user traffic are shown below:

Table 1. The route traversed by each user traffic in topology 2

User 1: L1-L17-L27-L25-L7	User 11: L2-L18-L27-L24-L6
User 2: L1-L17-L27-L23-L22-L5	User 12: L2-L10-L20-L22-L5
User 3: L1-L17-L27-L24-L6	User 13: L8-L26-L24-L13-L5
User 4: L1-L16-L15-L7	User 14: L8-L26-L24-L13-L5
User 5: L1-L17-L27-L23-L22-L5	User 15: L8-L26-L24-L6
User 6: L2-L10-L11-L4	User 16: L8-L15-L14-L6
User 7: L2-L10-L11-L4	User 17: L3-L20-L21-L4
User 8: L2-L10-L11-L4	User 18: L3-L20-L21-L4
User 9: L2-L18-L27-L25-L7	User 19: L3-L11-L4
User 10: L2-L18-L27-L25-L7	User 20: L3-L19-L27-L25-L7

For topology of Fig. 5 we have selected ‘k’ of the Kelly method and the initial ‘k’ of the proposed method in the iteration (28) to be 0.001.

The normalized link capacity for all links is selected to be 30. We also have used  $\rho_r = 0.01, \forall r \in R$  and  $\varepsilon=0.01$  in relation (28).

We have selected  $\sigma=1$  and  $\gamma=600$  in Eq. (10). The users’ utility functions are in logarithmic form with the following parameters:

Table 2. User utility function parameters for topology 2

$\omega(1) = 0.5$	$\omega(5) = 0.4$	$\omega(9) = 0.25$	$\omega(13) = 0.3$	$\omega(17) = 0.2$
$\omega(2) = 0.5$	$\omega(6) = 0.7$	$\omega(10) = 0.3$	$\omega(14) = 0.2$	$\omega(18) = 0.25$
$\omega(3) = 0.3$	$\omega(7) = 0.3$	$\omega(11) = 0.2$	$\omega(15) = 0.25$	$\omega(19) = 0.3$
$\omega(4) = 0.3$	$\omega(8) = 0.25$	$\omega(12) = 0.25$	$\omega(16) = 0.3$	$\omega(20) = 0.2$

In Fig. 9, the rate allocated to the typical user 11 has been compared between the proposed method and Kelly method. It can be verified that the proposed algorithm has a higher convergence rate.

In Figs.10-13, the aggregate rates through bottleneck links "2", "4", "8" and "27" have been compared and it can be concluded that the proposed method is faster in convergence rate.

The time evolution of ‘k’ parameters in the proposed method for typical users 4 and 11 are depicted in Fig. 14.

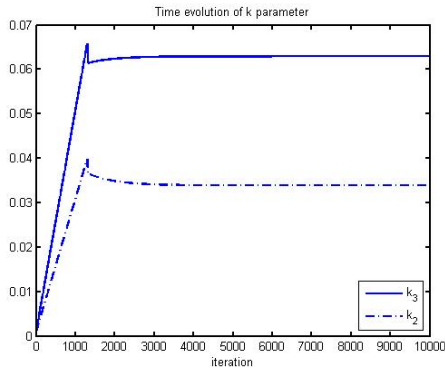


Fig. 8. Time evolution of parameter 'k' in the proposed method in the 1<sup>st</sup> topology

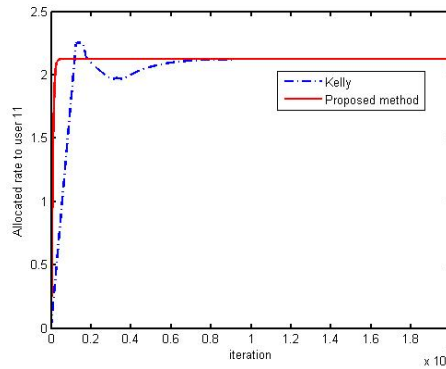


Fig. 9. Comparing rate of user 11 in proposed and Kelly methods in the 2<sup>nd</sup> topology

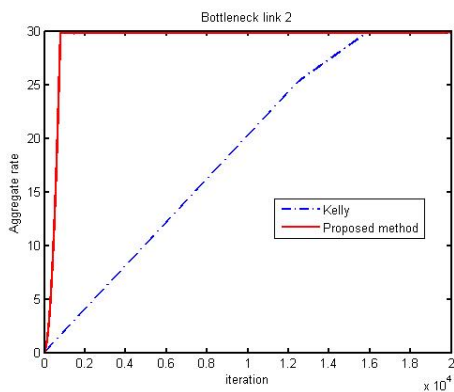


Fig. 10. Comparing aggregate rate through bottleneck 2 in different methods in the 2<sup>nd</sup> topology

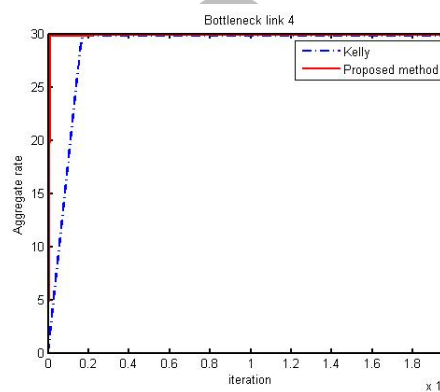


Fig. 11. Comparing aggregate rate through bottleneck 4 in different methods in the 2<sup>nd</sup> topology

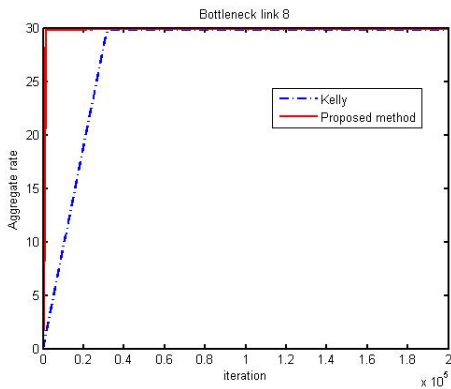


Fig. 12. Comparing aggregate rate through bottleneck 8 in different methods in the 2<sup>nd</sup> topology

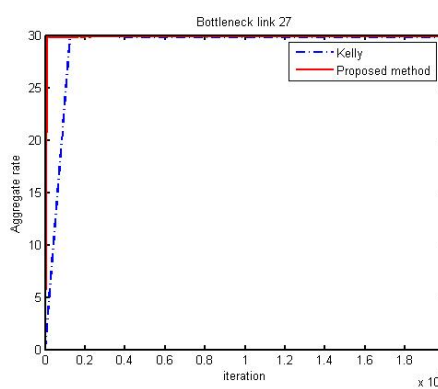


Fig. 13. Comparing aggregate rate through bottleneck 27 in different methods in the 2<sup>nd</sup> topology

**b) Loss in rate allocation of the Kelly method**

In this part, the parameter 'k' in the Kelly method is higher than part (a) and loss occurs in the network.

For the topology of Fig. 4 the 'k' parameter of the Kelly method and the initial 'k' of iteration (28) are selected to be 0.4. Other parameters are the same as part (a).

In Fig. 15 the aggregate rate through typical bottleneck link "2" are compared between the Kelly and the proposed methods.

The number of loss events (when the aggregate rate through a link exceeds its capacity) in the Kelly and the proposed method are 50 and 0 respectively.

In Fig. 16, the time evolutions of parameter k are sketched for typical users 4 and 11.

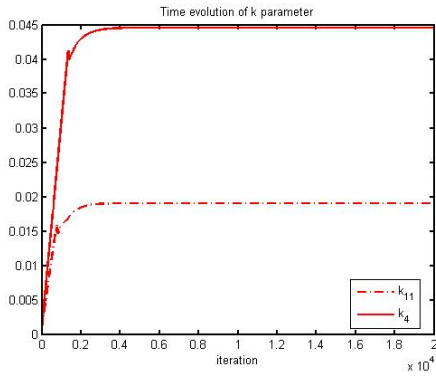


Fig. 14. Time evolution of two typical 'k' parameters in the proposed method in the 2<sup>nd</sup> topology

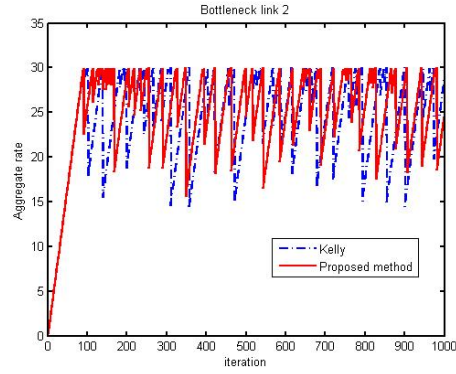


Fig. 15. Comparing aggregate rate through bottleneck 2 in different methods in the 1<sup>st</sup> topology

For the topology of Fig. 5 the 'k' parameter of the Kelly method and the initial 'k' of iteration (28) are selected to be 0.095. Other parameters are the same as part (a).

In Fig. 17, the rate allocated to the typical user 11 has been compared between the proposed and Kelly methods.

In Figs. 18-21 the aggregate rates through bottleneck links "2", "4", "8" and "27" have been compared between the Kelly and the proposed methods.

The time evolution of 'k' parameters in the proposed method for typical users 4 and 11 are depicted in Fig. 22.

The number of loss events in the Kelly and the proposed method are 1092 and 0 respectively.

It must be mentioned that the correct selection of the simulation parameters is of crucial importance in having a realistic simulation scenario.

For example, the incorrect selection of  $\gamma$  and  $\sigma$  parameters of the penalty function as it can be checked in Fig.1, can lead us to an incorrect estimation of ideal link penalty function.

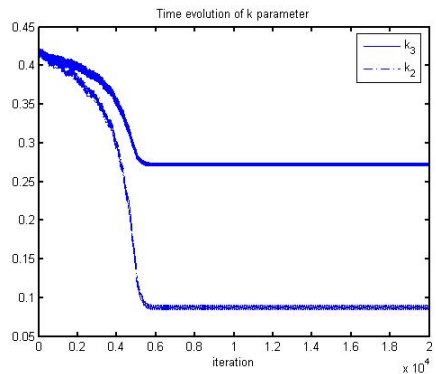


Fig. 16. Time evolution of parameter 'k' in the proposed method in the 1<sup>st</sup> topology

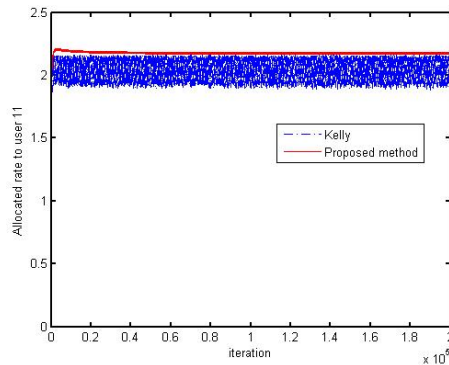


Fig. 17. Comparing rate of user 11 in proposed and Kelly methods in the 2<sup>nd</sup> topology

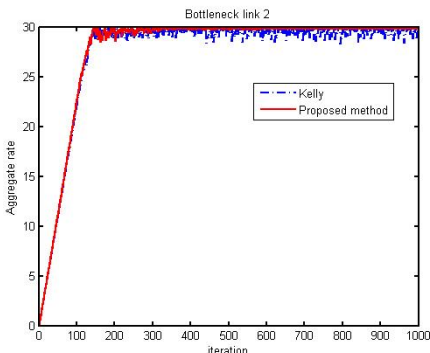


Fig. 18. Comparing aggregate rate through bottleneck 2 in different methods in the 2<sup>nd</sup> topology

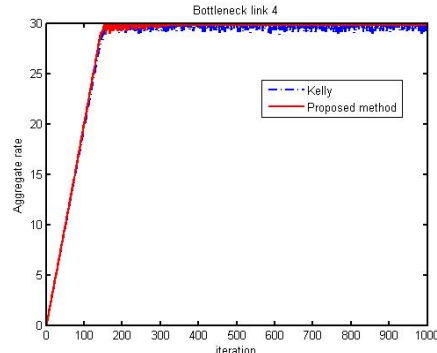


Fig. 19. Comparing aggregate rate through bottleneck 4 in different methods in the 2<sup>nd</sup> topology

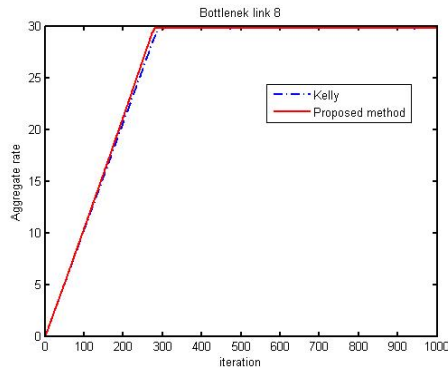


Fig. 20. Comparing aggregate rate through bottleneck 8 in different methods in the 2<sup>nd</sup> topology

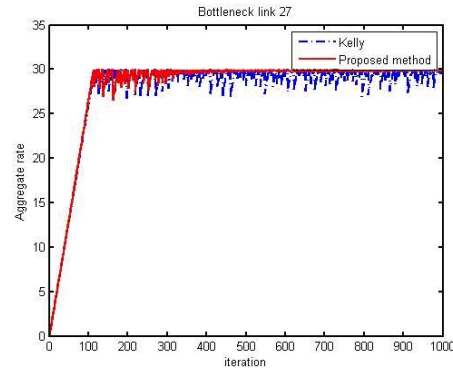


Fig. 21. Comparing aggregate rate through bottleneck 27 in different methods in the 2<sup>nd</sup> topology

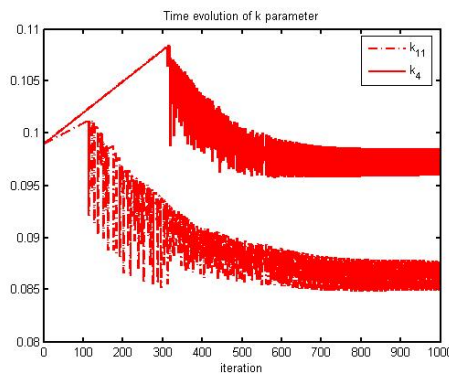


Fig. 22. Time evolution of two typical 'k' parameters in the proposed method in the 2<sup>nd</sup> topology

## 5. CONCLUSION

One of the deficiencies of the Kelly's method in [2] is the lack of presentation of a way by which each user can compute its 'k' parameter in a distributed way so that he (she) can ensure the loss-free property in his (her) allocated rate.

In this work, we have presented an algorithm by which users can change their 'k' parameter and the loss-free property of the allocated rate is guaranteed in a number of network topologies. We have investigated three different scenarios of communication networks including a simple *one link-one user* scenario, *one link-multi user* case and general *multi user-multi link* case. For a general network topology, finding an analytic and distributed solution for equation (19) is very difficult, thus we have tried to find an approximate and distributed solution for 'k' parameter. For this purpose we have used penalty function (10) and iterative method (28) and have shown by simulation that, the proposed method has a higher convergence rate and no loss with respect to the Kelly method.

One shortcoming of this distributed solution is that the proposed rate allocation algorithm may not be loss-free in general, but the important point about the approximate method in (28) is its distributed structure, i.e., each user can choose its 'k' parameter intelligently with only the help of the congestion parameters  $\lambda_{i,n}$  and  $\lambda_{i,n-1}$  that are fed back to him from the network.

## NOMENCLATURE

A	routing matrix
$c_j$	link 'j' capacity
C	links capacity vector
J	set of network links

$k_r$	gain parameter for user 'r'
$p_j(.)$	link 'j' penalty function
R	set of network users
T	time (iteration)
$u_r(.)$	user 'r' utility function
U	vector of user utility functions
V(.)	lyapunov function
$x_r$	rate (throughput) allocated to user 'r'
X	users' allocated rate vector

### Greek letters

$\varepsilon$	small positive gain parameter
$\gamma$	penalty function parameter
$\lambda_i$	aggregate penalty of user i's route
$\mu_j(.)$	link 'j' penalty function
$\omega_r$	user 'r' utility function parameter
$\Omega$	user's utility function parameter vector
$\Upsilon$	penalty function parameter
$\rho$	gain parameter in our algorithm
$\sigma$	penalty function parameter

### REFERENCES

- Jacobson, V. (1988). Congestion avoidance and control. *Comput. Commun. Rev.*, 18(4), 314-329.
- Kelly, F., P., Maulloo, A. K. & Tan, D. K. H. (1998). Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.*, 49(3), 237-252.
- Sheikholeslam, F., Tahani, V. & Zangeneh, H. R. Z. (2000). Periodic solutions of nonlinear equations based on extended stability theorems of nonlinear systems. *Iranian Journal of Science & Technology*, 24(1), 111-117
- Tahani, V. & Sheikholeslam, F. (1999). Total stability of fuzzy feedback control systems. *Iranian Journal of Science & Technology*, 23(Jan), 7-34
- Teshnelab, M. & Afyooni, D. (2004). An artificial intelligent system for traffic forecasting in virtual stations of highways Tehshnehlab. *Iranian Journal of Science and Technology, Transaction B: Engineering*, 28(3) B, 2004, pp 395-400
- Shenker, S. (1995). Fundamental design issues for the future Internet. *IEEE J Selected Areas Commun.*, 13(7), 1176-1188.
- Kelly, F. P. (1997). Charging and rate control for elastic traffic. *Eur. Trans. Telecommun.*, 8, 33-37.
- Mo, J. & Walrand, J. (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5), 556-567.
- Massoulié, L. & Roberts, J. (1999). Bandwidth sharing: objectives and algorithms. *Proc. IEEE INFOCOM*, 3, New York, 1395-1403.
- Bertsekas, D. & Gallager, R. (1987). *Data networks*. Prentice Hall, Englewood Cliffs, NJ.
- Kunniyur, S. & Srikant, R. (2000). End-to-end congestion control schemes: utility functions, random losses and ECN marks. *IEEE INFOCOM 2000-Tel Aviv*, Israel.
- La, R. J. & Anantharam, V. (2002). Utility-based rate control in the internet for elastic traffic. *IEEE Trans. On Networking*, 10(2), 272-286.
- Kalantari, A. & Kouhsari, S. M. (2005). A distributed computing approach for power system analysis. *Iranian Journal of Science and Technology*, 29(B4).

## APPENDIX

S. M. Proof of theorem 2:

We will follow the lines of proof, which is presented in [1] with some modifications.

The assumptions on  $\omega_r > 0$ ,  $r \in R$  and  $p_j$ ,  $j \in J$  ensure that  $V(X)$  is strictly concave on  $X \geq 0$  with an interior maximum; the maximizing value of  $X$  is thus unique. Observe that:

$$\frac{\partial}{\partial x_r} V(X) = \frac{\omega_r}{x_r} - \sum_{j \in R_r} p_j \left( \sum_{s: j \in R_s} x_s \right) \quad (a)$$

Setting these derivatives to zero identifies the maximum. Furthermore, we have:

$$\begin{aligned} \frac{d}{dt} V(X(t)) &= \sum_{r \in R} \frac{\partial \mathcal{G}}{\partial x_r} \cdot \frac{d}{dt} x_r(t) \\ &= \sum_{r \in R} k_r(X) \cdot \frac{1}{x_r} \cdot \left( \omega_r - x_r \cdot \sum_{j \in R_r} p_j \left( \sum_{s: j \in R_s} x_s \right) \right)^2 \end{aligned} \quad (b)$$

As  $k_r(X)$  is a positive parameter with a positive limit as  $t$  goes to infinity, it can be easily verified that  $V(X(t))$  is strictly increasing with 't', unless  $X(t) = X^*$ , the unique  $X$  maximizing  $V(X(t))$  which is the same solution as Eq. (4) displays. Thus the function  $V(X(t))$  is a Lyapunov function for the system.

Although our proof is presented for a continuous-time system, we can easily extend this proof to a discrete-time system as Kelly described in [1].

Archive of SID

Archive of SID