A Fuzzy-Hierarchical Algorithm for Proportionally-Fair Rate Allocation to Elastic Users

Pejman GUDARZI†, Hossein SAIDI†, and Farid SHEIKHOESLAM†, Nonmembers

SUMMARY Fairness is one of the most important features of a rate allocation strategy. Proportional fairness criterion has been recently proposed by F.P. Kelly and his colleagues. In this paper, we have proposed a two-level hierarchical technique which allocates proportionally-fair rates to the network elastic users. Part of the network links which are used commonly by the end-users and are congestion prone, constitute the higher (first) level of the hierarchy. In this level, the users with common path in the network are grouped as virtual users. End-users and remaining network links constitute the lower (second) level of hierarchy. To improve the convergence rate of the algorithm, a combination of Jacobi method and fuzzy techniques is deployed in the higher level of hierarchy. Implementing such fast algorithms in the higher level (which is topologically simpler than the whole network), reduces the computational complexity with respect to the use of such algorithms in the whole network. Additionally, the lower level penalty function computation is done once in each N iterations, which reduces the computational complexity furthermore. The simulation results show that the proposed algorithm outperforms that of Kelly in the convergence speed.

key words: proportional fairness, elastic traffic, penalty function, FIS, utility function

1. Introduction

Two different methods exist for accomplishing congestion control in the data networks. One is window-based method in which the number of outstanding packets in the network is regulated by adjusting the size of a congestion window to a reference value [1]. In the rate-based method, we look at the network traffics as fluid flows and algorithms such as the Kelly’s method are used in order to achieve some fairness criteria in rate allocation [2].

There are different fairness criteria such as max-min, proportional and minimum potential delay fairness [3]. Selecting a fairness criterion depends on the network’s designer strategy.

For example in the max-min criterion, the focus is on the users with lowest rates whereas in proportional criterion the objective is to maximize the sum of the logarithms of the user rates and penalize more the users who use long routes in the network. In minimum potential delay criterion, L. Massoulié et al. define a delay measure in terms of the user rates and try to minimize that.

In this paper we assume that the network traffic can adapt itself to the network conditions. In another word, we use the term ‘elastic’ for the traffic as it was introduced by S. Shenker in [4] and used in the Kelly’s paper [2]. Examples of such traffic type are TCP traffic in the current Internet and ABR traffic in the ATM networks.

In the current paper, a fuzzy-hierarchical method is used to achieve proportional fairness in the rate allocation. The hierarchical model decomposes the network into two levels of hierarchy.

Part of the network links which are used commonly by the end-users and are congestion prone, constitute the higher (first) level of the hierarchy. Higher (first) level of hierarchy is associated with the shared links or backbone part of the network. In this level, the users with common path in the network are grouped as virtual users. Users within each virtual user, traverse a common path in the backbone. End-users and remaining network links constitute the lower (second) level of hierarchy.

The proposed algorithm reduces the communication overhead and increases the convergence speed in the higher level. Also this algorithm reduces the computational overhead. In the lower level of the hierarchy, penalty function computation is done once in each N iterations, which reduces the computational complexity.

To improve the convergence rate of the algorithm, a combination of Jacobi method and fuzzy techniques is used in the higher level of hierarchy. Although the mixed-fuzzy hierarchical algorithm is more complex than the plain hierarchical, implementing such fast algorithms in the higher level, reduces the computational complexity with respect to the use of such algorithms in the whole network. The proposed fuzzy-hierarchical algorithm and plain hierarchical (hierarchical) algorithm (which uses only the Jacobi method in the higher level of hierarchy) have been compared with the Kelly’s algorithm using simulation.

The paper is organized as follows. In Sect. 2 we review some related works and especially in detail the work of F.P. Kelly et al. [2]. In Sect. 3, the proposed algorithm is described. Section 4 is devoted to stability and convergence analysis of the proposed algorithm. Section 5 presents the simulation results, which is followed by the summary and conclusion in Sect. 6.

2. Background

As it is known, large networks such as the Internet have been designed to be decentralized and depend on the well-defined behavior from end-hosts. The increasing complexity and size of these networks make centralized rate allocation impractical. Without a centralized control, the network users
have a great deal of freedom in sharing the available bandwidth in the network.

To achieve the flow control, congestion avoidance and bandwidth allocation, the researchers have proposed different rate allocation algorithms to be implemented at the end-hosts in a decentralized manner [5]–[8].

The most widely used flow control/congestion avoidance mechanism in the current Internet is TCP, which is a window-based mechanism. TCP, however, does not necessarily lead to a fair or efficient rate allocation of the available bandwidth [1].

Recently, Kelly et al. have proposed an algorithm that results in proportional fairness criterion and they used a Lyapunov function approach for stability analysis of their rate allocation method [2].

Mo and Walrand [1] have proposed and studied another fair window-based end-to-end congestion control mechanism, which is similar to TCP Vegas [9] but has a more sophisticated window-updating rule. They have shown that the proportional fairness can be achieved by their (p, 1)-proportionally fair algorithm and max-min fairness can be achieved as a limit of (p, α)-proportionally fair algorithm as α goes to infinity.

It is clear that fairness is a desirable feature of a rate allocation algorithm. Users’ preferences can be captured by appropriate utility functions. Due to the various requirements of different applications, it is likely that the users will have different utility functions [4]. For example, suppose that a user is transferring a file. The per-transfer delay is inversely proportional to the rate it receives. Hence, the delay might be modelled as the utility function of the user which is a function of its rate. This reveals this fact that although fairness is a desirable property, fairness by itself may not be sufficient. A good rate allocation mechanism should not only be fair, but also should allocate the available bandwidth in such a way that the overall utility of the end-users is maximized [10].

In the sequel, we will review the model and the rate allocation algorithm which are used by Kelly and then, we will describe the proposed algorithm.

### 2.1 Problem Formulation

Consider a network with a set J of resources or links and a set R of users and let Cj denotes the finite capacity of link j ∈ J. Each user r has a fixed route Rr, which is a nonempty subset of J. Also, define a zero-one matrix A, where Aij = 1 if link j is in user r’s route Rr and Aij = 0 otherwise. When the allocated rate to the user r is x, user r receives utility U, (x,). The utility U, (x, ) is an increasing, strictly concave and continuously differentiable function of x over the range x ≥ 0. Furthermore, assume that the utilities are additive so that the aggregate utility of rate allocation X = (x, r ∈ R) is: \( \sum_{r \in R} U_r(x_r) \).

The Kelly’s formulation of the problem is as follows:

\[
\text{SYSTEM}(U; A, C):
\begin{align*}
\text{Max} & \quad \sum_{r \in R} U_r(x_r) \\
\text{Subject to} & \quad A^T X \leq C \\
\text{Over} & \quad X \geq 0
\end{align*}
\]

The first constraint says that the total rate through a link cannot be larger than the capacity of the link. Given that the system knows the utility functions U of the users, this optimization problem might be mathematically tractable. However, in practice, the network is not likely to know U of each user; additionally it is impractical for a centralized system to compute and allocate the users’ rates due to the scale of the network. Hence, Kelly in [2] has proposed to consider the following two simpler problems. Suppose that each user r is given the price per unit rate \( \lambda_r \) [11]. Given \( \lambda_r \), user r selects an amount that he or she is willing to pay per unit time, \( \omega_r \), and receives a rate \( x_r = \omega_r / \lambda_r \). Then, the user r optimization problem becomes selecting \( \omega_r \):

\[
\text{USER}(U_r; \lambda_r):
\begin{align*}
\text{Max} & \quad U_r(\omega_r / \lambda_r) - \omega_r \\
\text{Over} & \quad \omega_r \geq 0
\end{align*}
\]

The network, on the other hand, given the amounts that users are willing to pay, \( \Omega = (\omega_r, r \in R) \), attempts to maximize the sum of weighted log functions \( \sum_{r \in R} \omega_r \log(x_r) \). Then the network’s optimization problem can be written as follows:

\[
\text{NETWORK}(A, C, \Omega):
\begin{align*}
\text{Max} & \quad \sum_{r \in R} \omega_r \log(x_r) \\
\text{Subject to} & \quad A^T X \leq C \\
\text{Over} & \quad X \geq 0
\end{align*}
\]

Note that the network does not require the true utility functions \( U_r(\cdot), r \in R \), and to carry out its computations pretends that user r utility function is \( \omega_r \log(x_r) \). It is shown in [2] that one can always find vectors \( x^* = (\lambda^*_r, r \in R) \), \( \Omega^* = (\omega^*_r, r \in R) \) and \( X^* = (x^*_r, r \in R) \) such that \( X^* \) solves NETWORK\((A, C; \Omega^*)\), \( \omega_r^* \) solves USER\(_r\) \((U_r; \lambda_r)\) and \( \lambda_r^* = x^*_r / \lambda_r^* \) for all \( r \in R \). Furthermore, the rate allocation \( X^* \) is also the unique solution to SYSTEM\((U; A, C)\). As we have mentioned earlier, the format of users’ utility functions are in close relationship to the fairness criteria that exists in rate allocation [1].

From now on, to implement proportional fairness [2] it is assumed that the users’ utility functions are in logarithmic form.

It must be mentioned that in the special case of logarithmic utility functions, USER problem has a unique solution equal to the user’s \( \omega \) and does not need to be solved by the users periodically. Thus, we only focus on the NETWORK problem.

**Definition** [2], [12]:

A vector of rates \( X = (x_r, r \in R) \) is per unit charge proportionally fair if it is feasible, that is \( X \geq 0 \) and \( A^T X \leq C \), and if for any other feasible vector \( X^* \) the aggregate of
proportional changes is zero or negative[2]:

\[ \sum_{r \in R} \omega_r \frac{x_r^* - x_r}{x_r} \leq 0 \]  

(4)

If we assume that \( \omega_r = 1, \forall r \in R \) then we reach to the *proportional fairness* criterion. From now on, for notational simplicity, we drop the prefix *per unit charge* for the general case and refer to the general definitions as *proportionally fair*.

The Kelly’s discrete time algorithm for solving NET-WORK \((A, C; \Omega)\) is as follows:

\[ x_r[n + 1] = \left( x_r[n] + k_r \cdot (\omega_r - x_r[n] \cdot \sum_{j \in R} \mu_j[n]) \right)^+ \]  

(5)

Where \( \{x\}^+ \triangleq \max(0, x) \) and:

\[ \mu_j[n] = p_j \left( \sum_{i \in J} x_i[n] \right) \]  

(6)

Parameter \( k_r \) controls the speed of convergence in Eq. (5), \( p_j(y) \) is the amount that link \( j \) penalizes its aggregate traffic \( y \) and is a non-negative, continuous increasing function (see Fig. 1).

One of the interpretations is that using (5), the system tries to equalize \( \omega_r \) with \( x_r[n] \cdot \sum_{j \in R} \mu_j[n] \) by adjusting \( x_r[n] \).

In [2] it is shown that the unique, optimal and proportionally-fair equilibrium point of Eqs. (5), (6) is:

\[ \omega_r = x_r^* \cdot \sum_{j \in R} p_j \left( \sum_{i \in J} x_i^* \right), \ r \in R \]  

(7)

It is necessary to mention that as Kelly et al. have discussed in [2], for satisfying the constraints of relations (1) to (3), we may use the following form of the link penalty function:

\[ p_j(y) = (y - c_j + \epsilon)^+ / \epsilon^2, \epsilon \to 0, \ j \in J \]  

(8)

Where, \( c_j \) is the capacity of \( j \)-th link and \( \epsilon > 0 \) is a small positive number.

2.2 A Closer Look

Although the Kelly’s algorithm has a set of outstanding features in allocating fair rates to competing users in a data network, we try to improve its performance in the following two aspects.

a) Using high-speed algorithms such as Jacobi method in large networks requires more computation than simple gradient descent algorithm. Hence, sometimes we may prefer using the simple gradient descent algorithm and ignore the faster convergence speed that we could gain in using high-speed algorithms in the entire network.

b) As we have described before, for implementing the Kelly’s algorithm, each user \( r \) must compute and report its \( \omega_r \) based on congestion information \( \lambda_r \) which is received from the network. In large networks implementation of this algorithm may require enormous overhead related to the exchange of the update information between nodes and end-users and this makes the algorithm practically hard to implement.

As you can see in Fig. 2, ‘*hierarchical structure*’ means that the network can be partitioned into a number of virtual users (they are specified by letters \( A, B, C, D \) and \( E \) in Fig. 2) which share common paths in the entire network.

In the hierarchical model, part of the network that carries virtual users’ traffic is subject to the congestion is called ‘*backbone*’ (dashed line in Fig. 2 is the backbone boundary). It is assumed that the links in the lower level of hierarchy are not subject to the congestion.

As an example of a hierarchical topology, we can imagine a combination of campus LAN networks in a larger MAN network.

In this paper, the item (a) is addressed by using a plain hierarchical method and then improving it further using a combination of fuzzy and Jacobi methods. The high-speed algorithm is used at the first level of the hierarchy to allocate rate to each virtual user. This algorithm is applied to the backbone of the network which is less complex; therefore overhead and computational tasks are reduced.

Also, we reduce the computational load that was mentioned in item (b), in the second (or lower) level of hierarchy. In the second level, each network node needs to compute and transmit the penalty function value once every \( N \) iterations, based on the aggregate input rate. If the parameter \( N \) is selected to be large, the computation and transmission overheads would be reduced, but this would decrease the convergence speed as well. So selecting a specific value of \( N \) is a trade off between the computation and transmission...
overheads and the convergence speed.

3. Proposed Algorithm

The proposed algorithm has been described in two steps; in part (3.1), the hierarchical aspect of the fuzzy-hierarchical algorithm is described (as a plain hierarchical algorithm). In part (3.2), the fuzzy controller is applied to the hierarchical algorithm.

3.1 Hierarchical Rate Allocation

First look at an example. Consider Fig. 3. Let’s assume that the network is consisted of 11 elastic sources that are included in four source virtual users. Dotted lines show the boundaries of the virtual users and thick lines show the aggregate flow of each virtual user that is traversing through the links that belong to backbone (these links are denoted by letters L6, L7 and L8). Each source (destination) of information is denoted by s (d) and as mentioned before, the rate associated with each (source, destination) pair is denoted by x. Links are unidirectional and in Fig. 3, links 6, 7 and 8 constitute the backbone.

As Kelly has shown in [2], the stabilized rates of users are:

\[ x_i^* = \frac{\lambda_i}{\Lambda_i^*}, \quad r \in R \]

Where:

\[ \Lambda_i^* = \sum_{j \in R} p_j \left( \sum_{u \in R_u} x_u^* \right) \]

Since it is assumed that the congestion may only occur in the links which belong to the backbone, we may consider that \( \Lambda_i^* \) is only affected by the backbone links and is approximated by:

\[ \Lambda_i^* \approx \sum_{j \in R, \ j \in \text{Backbone}} p_j \left( \sum_{u \in R_u} x_u^* \right) \]

(9)

For example, for users \( s_1 \) and \( s_2 \) in Fig. 3, we would have:

\[ x_1^* = \frac{\omega_1}{\Lambda_1^*}, \quad x_2^* = \frac{\omega_2}{\Lambda_2^*} \]

(10)

Define:

\[ \Lambda_1^* \triangleq p_6 \left( \sum_{x \in L_6 \cup L_8} x_u^* \right) \]

Where \( \Lambda_1^* \) is the aggregate penalty of users \( s_1 \) and \( s_2 \) in the backbone of the network (link ‘6’ in this case), which is approximately equal to \( x_1^* \) or \( x_2^* \). Then, at the equilibrium point, the aggregate rate of virtual user ‘1’ is:

\[ x_1^* + x_2^* = \frac{\omega_1}{\Lambda_1^*} + \frac{\omega_2}{\Lambda_2^*} \leq \frac{\omega_1 + \omega_2}{\Lambda_1^*} \]

(11)

In another word, the virtual user ‘1’ might be regarded as a user with logarithmic utility function \( (\Omega_1 \log(x_1)) \) in which \( \Omega_1 = \omega_1 + \omega_2 \).

If we denote the aggregate rate of virtual user ‘1’ with \( \chi_1 \), at the equilibrium point we have:

\[ x_1^* = \frac{\omega_1 + \omega_2}{\Lambda_1^*} \chi_1 \]

(12)

By considering Eqs. (10) and (12) and the assumption that \( \Lambda_i^* \approx \Lambda_i^* \), then:

\[ x_i^* = \frac{\omega_i}{\Omega_i} \chi_i^* \]

(13)

Now, in the mathematical terms [13], let \( S \triangleq \{ S_i \mid i = 1, 2, \ldots, Q \} \) and \( D \triangleq \{ D_i \mid i = 1, 2, \ldots, Q \} \) be the sets that represent the virtual sources and virtual destinations. Where, \( Q \) represents the number of virtual sources (destinations). For example, in Fig. 3 we have \( Q = 4 \) and \( S_1 = \{ s_1, s_2 \}, D_3 = \{ d_6, d_7 \} \).

If the rate associated with virtual user \( i \) at iteration \( n \) is denoted by \( \chi_i[n] \), and the rate of end users (as mentioned before) are denoted by small \( x \) letter, the algorithm behaves in the following manner:

At the beginning, the algorithm works in the first level of hierarchy and allocates rates to the virtual sources using some high-speed algorithm (such as Jacobi method). Then, each virtual user assigns some proportions of its rate to each end-user within the virtual user. Afterwards, by defining a temporary variable \( w \), each user updates its corresponding \( w \) parameter and when these new parameters are sent back to the virtual users, the first-level algorithm repeats its computations.

If the assumption in Eq. (9) is true, when the system is in the vicinity of equilibrium point, users’ rates are close to the optimal values. It will be shown in Sect. 4, that by repeating this procedure, the rates will converge to the optimal rates. We must emphasize here that the \( w \) parameters which are updated in the algorithm by end-users have not the interpretation of users’ willingness to pay (in contrast with what is discussed in [2] about \( \omega \) and are merely temporary variables. The rate assignment by virtual user \( i \) to a user \( u \)
located within virtual user $i$ is:

$$x_u[n + 1] = x_i[n] \cdot \frac{W_i[n]}{\Lambda_i[n]}, \quad n = 0, 1, 2, \ldots$$

$$i = 1, 2, \ldots, Q, \ u \in i$$  \hfill (14)

Where notation $u \in i$, means that user $u$ is located within virtual user $i$ and:

$$W_i[n] = \sum_{u \in i} w_u[n]$$  \hfill (15)

Updating $\chi_i[n]$ in Eq. (14) is as follows:

$$\chi_i[n + 1] = \left\{ \begin{array}{lr}
\chi_i[n] + \frac{K_i \cdot (W_i[n] - \chi_i[n] \cdot \Lambda_i[n])}{\chi_i[n] + \chi_i[n] \cdot \frac{\partial \Lambda_i(t)}{\partial \chi_i(t)}}, & i = 1, 2, \ldots, Q \\
\alpha_u \cdot \frac{w_u[n]}{\chi_i[n]} & \text{otherwise}
\end{array} \right.$$  \hfill (16)

This is basically the same as Eq. (5) or Kelly’s equation, where we substitute the gradient descent method with the Jacobi method, and additionally we have used the approximation of the $\Lambda$ by using the virtual user concept.

In this equation:

$$\Lambda_i[n] = \sum_{j \in R_i, k \in \text{Backbone}} p_{j} \left( \sum_{i \in R_j} x_u[n] \right)$$

Using the virtual user concept introduces approximation error in $\Lambda$ and therefore in the final allocated rates. To compensate for this approximation error, each $w$ parameter is updated in a time scale which can be much larger than that of $\chi$ using the following relation:

$$w_u[n + 1] = \left\{ \begin{array}{lr}
\chi_i[n] \cdot \frac{W_i[n]}{\Lambda_i[n]} & \text{otherwise}
\end{array} \right.$$  \hfill (17)

Where $w_u[0] = \omega_u$ (the user-logarithmic utility function parameter), $u \in i, i = 1, 2, \ldots, Q$ and $N$ is some large and finite positive integer. The larger we select the parameter $N$, the further we can reduce the overhead associated with the exchange of the $w_u$ parameters from Eq. (17) to Eq. (16) and the processing required to compute and transmit the penalty function to the end-users once in each $N$ iterations. Hence, large values of $N$ can decrease this transmission overhead, but this will decrease the convergence speed.

$\alpha_u$ is a positive constant ($0 < \alpha_u < \delta_u, \forall i, u \in i$) that controls the convergence speed in Eq. (17) and $\delta_u > 0$ is an upper bound for $\alpha_u$.

The idea behind Eq. (17) is that users try to adjust their final rates which are assigned to them by first-level algorithm ($w_u[n]/\Lambda_i[n]$) to the Kelly’s rates ($\omega_u/\lambda_i[n]$) by changing their $w$ parameters.

In Sect. 4, we will conclude the stability of the plain hierarchical method and its convergence to the optimal rates, using the corollary associated with Theorem 2.

Also it can be seen that the stability does not depend on any specific value of parameter $N$ (i.e. $N$ can be any positive integer).

3.2 Fuzzy-Hierarchical Rate Allocation

For further increasing the convergence rate of the first level algorithm (Eq. (16)), we have used some fuzzy techniques in which we intelligently control the gain parameters $K_i$ (for all $i$) in such a way that they increase the overall convergence rate of the algorithm. In the sequel, the FIS (Fuzzy Inference System [14], [15]) controller is described in detail.

Consider the network topology of Fig. 4. There are 38 elastic users which compete for their fair rates in a network of 54 links. The convergence speed of Eq. (16) might be increased using a simple FIS which controls each user gain parameter $K_i$ in this equation in a distributed manner. This fuzzy model consists of four basic parts.

3.2.1 Congestion Metric

This parameter behaves as input to the model and is considered to be:

$$\zeta_i = \frac{\chi_i \cdot \Lambda_i - W_i}{W_i}, \quad i = 1, 2, \ldots, Q$$  \hfill (18)

The congestion metric in Eq. (18) is a large positive value for high congestion and tends to $-1$ for low congestion levels in the network.

3.2.2 Input Membership Functions

The ‘poor,’ ‘good’ and ‘excellent’ membership functions for input variable $\zeta_i$ are depicted in Fig. 5.

3.2.3 Output Membership Functions

The output of FIS controller is $K$ parameter and its membership functions are found based on a heuristic manner. For the topology of Fig. 4, its ‘low,’ ‘medium’ and ‘high’ membership functions are shown in Fig. 6.
3.2.4 Inference Engine

The inference engine is consisted of the following three simple rules:

**Rule1:**
If $\zeta$ is poor then $K_i$ is low.

**Rule2:**
If $\zeta$ is good then $K_i$ is medium.

**Rule3:**
If $\zeta$ is excellent then $K_i$ is high.

In general, the purpose of using this type of fuzzy-based hierarchical algorithms is to find a rate allocation strategy in which the virtual users can adjust their gain parameter $K_i$ in Eq. (16) using a fuzzy controller.

This is done based on the congestion information received from the network. Distributed structure [16] is one of the outstanding features of this algorithm i.e. each virtual user is capable of computing its appropriate gain parameter $K_i$ independently and only the congestion information of the network is needed.

In Sect. 4, the stability of the fuzzy-hierarchical method and its convergence to the optimal rates in Eq. (7) have been proved using Lyapunov function approach. Also it is shown that the stability does not depend on any specific value of $N$ (i.e. $N$ can be any positive integer).

4. Stability Analysis

In fact, to demonstrate the convergence and stability of the fuzzy-hierarchical method, first we must show the stability and convergence of Eq. (16) which will be a form of time-varying projected Jacobi algorithm [16] (parameter $K_i$ changes in time by fuzzy controller). Then, we must show that the $w$ parameters in Eq. (17) can also converge to some positive values.

The convergence and stability of the time-varying projected Jacobi and the fuzzy-hierarchical algorithms are proved using Theorems 1 and 2 respectively. Then, using a corollary, it is shown that the plain hierarchical algorithm is a special case of the fuzzy-hierarchical algorithm; therefore Theorem 2 would apply to it as well.

**Theorem 1:** Let $J = J_L \cup J_H$ where subscripts $L$ and $H$ denote the lower level and the higher level (backbone) parts of the model respectively. Consider the following continuous-time system (for $i = 1, 2, \ldots, Q$):

$$
\frac{d}{dt} \chi_i(t) = K_i(t) \cdot \frac{W_i - \chi_i(t) \cdot \Lambda_i}{\chi_i(t)} + \chi_i(t) \cdot \frac{\partial}{\partial \chi_i(t)} \Lambda_i
$$

(19)

Also assume that:

$$K_i(t) > 0 \text{ and } W_i > 0, \quad i = 1, 2, \ldots, Q$$

Then the function:

$$V(\chi) = \sum_{i=1}^{Q} W_i \cdot \log \chi_i - \sum_{j \in J_H} \int_0^{\chi_j} \chi_j \cdot p_j(y)dy$$

(20)

is a Lyapunov function for the above system. The vector $\chi^*$ maximizing $V(\chi)$ is the stable point of the system (19) and also the discrete-time system (16), to which all trajectories converge [2]. Here, $R_i$ denotes the path that is traversed by virtual user $s$ in the backbone. $K_i(t)$ is the numerical value of fuzzy rules in part (3.2.4) and $p_j(y)$ is a non-negative, continuous, increasing penalty function of $y$.

Proof is shown in Appendix A.

**Theorem 2:** Consider Eqs. (16), (17). Assume that the network topology can be expressed in a hierarchical manner. Furthermore, assume that $1 \leq \lambda_s[n] / \Lambda_s[n] \leq M < \infty$ for $n \leq n_0$, where, $M \geq 1$ and $n_0$ (an integer) are some positive constants.

Also in Eqs. (16) and (17) assume $K_i$ is being changed by a fuzzy controller and $K_i, \alpha_u$ are small enough such that $\chi_i[n]$ and $w_u[n]$ ($i = 1, 2, \ldots, Q, u \in j$) remain finite for $n \leq n_0$ and that the following approximation is true in continuous-time for $t > n_0$:

$$\frac{\partial}{\partial w_u(t)} \left( \frac{\lambda_s(t)}{\Lambda_s(t)} \right) \approx 0$$
∀i, j ∈ {1, 2, . . . , Q}, u ∈ i, v ∈ j

Then system (16), (17) converge to the unique equilibrium point that leads to the user rates which are proportionally-fair and optimal as in Eq. (7).

In the special case that Eq. (9) is valid, we have in continuous-time, λi(t) ≈ λi, n0 = 0 and M = 1, i = 1, 2, . . . , Q, u ∈ i.

Proof of Theorem 2 is shown in Appendix B.

Corollary: In the special case of the plain hierarchical algorithm, parameter Ki(t) in Eq. (19), is considered to be a positive constant. Therefore the same procedure as in Theorem 2 can be applied to prove the stability and convergence of the plain hierarchical algorithm to the optimal rates.

Theorem 3: Assume that the approximation (21) is true and N in the Eq. (17) is large enough such that Eq. (16) has been approximately converged. Now at equilibrium if we have:

\[ 1 \leq \frac{\lambda^*_u}{\lambda^*_i} \leq 2, i = 1, 2, \ldots, Q, u \in i \]  \hspace{1cm} (22)

then, there exists an upper bound δu for αu in Eq. (17) such that this equation is convergent. This upper bound is equal to:

\[ \delta_u = \frac{2 \omega_u \cdot \Lambda^*_i}{\lambda^*_u} \]  \hspace{1cm} (23)

Where, \( \Lambda^*_i \) and \( \lambda^*_u \) are the stabilized values of \( \Lambda_i[n] \) and \( \lambda_u[n] \) parameters. Furthermore, the stabilized value of \( w_u[n] \) in Eq. (17) would be:

\[ w_u^\Delta = \lim_{n \to \infty} w_u[n] = \frac{\delta_u}{2}, u \in i, i = 1, 2, \ldots, Q \]  \hspace{1cm} (24)

Proof is shown in Appendix C.

5. Simulation Results

Here, a macroscopic fluid flow model [5] is used. The network is simulated and the algorithms are being compared with the Kelly algorithm.

Two different hierarchical algorithms have been used for the simulation. In one of them, a simple Jacobi method has been used in the first level of the hierarchy and in the other one, a mixed fuzzy-Jacobi method has been deployed in the first level to improve the convergence speed.

For simulation, we have used the topology that is given in Fig. 4. Link’s penalty functions are considered to have the forms as follow.

In Kelly’s method, the link penalty functions are:

\[ p_j(y) = (y - c_j + \epsilon_1)^+ / \epsilon_1^2, \hspace{0.5cm} j \in J \]  \hspace{1cm} (25)

In the hierarchical methods, the penalty functions are:

\[ p_j(y) = \epsilon_2 \cdot \tan(\pi y/(2c_j)), \hspace{0.5cm} j \in J \]  \hspace{1cm} (26)

where y is the aggregate rate of users traversing the link j and \( \epsilon_1, \epsilon_2 \) are some small positive constants. The smaller the values of these constants are, the better approximation of the solution of the SYSTEM (U; A, C) problem [2] we would have.

The users’ utility functions are in logarithmic form and use the parameters that are listed in Table 1.

In Fig. 4, symbols starting with S and D represent the sources and destinations respectively and gray nodes that are specified by letter e are edge nodes that are located between the lower level and higher level (backbone) links. The common path that is traversed by each virtual user in the backbone of the network of Fig. 4 is specified in Table 2. The link capacities of Fig. 4 are shown in Table 3.

Table 1 User utility function parameters.

<table>
<thead>
<tr>
<th>User</th>
<th>( \omega )</th>
<th>User</th>
<th>( \omega )</th>
<th>User</th>
<th>( \omega )</th>
<th>User</th>
<th>( \omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
<td>11</td>
<td>0.2</td>
<td>21</td>
<td>0.3</td>
<td>31</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>12</td>
<td>0.5</td>
<td>22</td>
<td>0.3</td>
<td>32</td>
<td>0.7</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>13</td>
<td>0.3</td>
<td>23</td>
<td>0.4</td>
<td>33</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td>0.3</td>
<td>14</td>
<td>0.3</td>
<td>24</td>
<td>0.7</td>
<td>34</td>
<td>0.3</td>
</tr>
<tr>
<td>5</td>
<td>0.4</td>
<td>15</td>
<td>0.4</td>
<td>25</td>
<td>0.25</td>
<td>35</td>
<td>0.21</td>
</tr>
<tr>
<td>6</td>
<td>0.7</td>
<td>16</td>
<td>0.7</td>
<td>26</td>
<td>0.3</td>
<td>36</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>0.3</td>
<td>17</td>
<td>0.25</td>
<td>27</td>
<td>0.22</td>
<td>37</td>
<td>0.3</td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>18</td>
<td>0.3</td>
<td>28</td>
<td>0.5</td>
<td>38</td>
<td>0.3</td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>19</td>
<td>0.21</td>
<td>29</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.3</td>
<td>20</td>
<td>0.5</td>
<td>30</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Common paths that are traversed by each source virtual user of Fig. 4 in the backbone.

<table>
<thead>
<tr>
<th>Source virtual user</th>
<th>Common path in the backbone</th>
</tr>
</thead>
<tbody>
<tr>
<td>[S30,S31]</td>
<td>L6-L14-L26-L24</td>
</tr>
<tr>
<td>[S32,S33]</td>
<td>L5-L8-L13-L22-L26</td>
</tr>
<tr>
<td>[S34,S35]</td>
<td>L1-L6-L17-L27-L24</td>
</tr>
<tr>
<td>[S9,S10,S11]</td>
<td>L1-L4-L17-L28-L21</td>
</tr>
<tr>
<td>[S18,S19,S20,S21]</td>
<td>L2-L4-L18-L28-L21</td>
</tr>
<tr>
<td>[S22,S23,S24,S25,S26]</td>
<td>L3-L4-L20-L21</td>
</tr>
<tr>
<td>[S34,S35,S36,S37,S38]</td>
<td>L4-L8-L12-L23-L21</td>
</tr>
<tr>
<td>[S1,S2,S3,S5,S6]</td>
<td>L1-L7-L17-L27-L25</td>
</tr>
</tbody>
</table>

In Eq. (25) we have selected \( \epsilon_1 = 10^{-2} \) or only 0.2% of the minimum link capacities and we have used \( \epsilon_2 = 10^{-8} \) in Eq. (26). Thus the final rate assignments are close approximations of the actual solution of SYSTEM problem [2]. Also, \( k_i = 0.00003 \) is used in the Kelly’s method. In the hierarchical methods, \( N = 1000 \), \( Q = 11 \) and \( \alpha_u = 0.3 \) are used for each \( u \in i \) in Eq. (17).

Parameters \( K_i \) in Eq. (16) are controlled using a fuzzy controller whose input and output membership functions are depicted in Figs. 5 and 6 respectively.

In the fuzzy method, we have used the Mamdani method for reasoning with rules denoted in Sect. 3.2.4 and the defuzzification is based on the centroid method [14].

Some users’ rates are shown in Fig. 7 to Fig. 10. In
Table 3  Link capacities in Fig. 4.

<table>
<thead>
<tr>
<th>Link#</th>
<th>Capacity</th>
<th>Link#</th>
<th>Capacity</th>
<th>Link#</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>19</td>
<td>50</td>
<td>37</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>20</td>
<td>50</td>
<td>38</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>21</td>
<td>50</td>
<td>39</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>22</td>
<td>50</td>
<td>40</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>23</td>
<td>50</td>
<td>41</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>24</td>
<td>50</td>
<td>42</td>
<td>200</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>25</td>
<td>50</td>
<td>43</td>
<td>200</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>26</td>
<td>50</td>
<td>44</td>
<td>200</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>27</td>
<td>50</td>
<td>45</td>
<td>200</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>28</td>
<td>50</td>
<td>46</td>
<td>200</td>
</tr>
<tr>
<td>11</td>
<td>50</td>
<td>29</td>
<td>200</td>
<td>47</td>
<td>200</td>
</tr>
<tr>
<td>12</td>
<td>50</td>
<td>30</td>
<td>200</td>
<td>48</td>
<td>200</td>
</tr>
<tr>
<td>13</td>
<td>50</td>
<td>31</td>
<td>200</td>
<td>49</td>
<td>200</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>32</td>
<td>200</td>
<td>50</td>
<td>200</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>33</td>
<td>200</td>
<td>51</td>
<td>200</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>34</td>
<td>200</td>
<td>52</td>
<td>200</td>
</tr>
<tr>
<td>17</td>
<td>30</td>
<td>35</td>
<td>200</td>
<td>53</td>
<td>200</td>
</tr>
<tr>
<td>18</td>
<td>50</td>
<td>36</td>
<td>200</td>
<td>54</td>
<td>200</td>
</tr>
</tbody>
</table>

Fig. 7  Rate allocated to user ‘6’ in different methods.

Fig. 8  Rate allocated to user ‘18’ in different methods.

In these figures, the proposed methods are compared with the Kelly’s method. For example, in Fig. 7, the fuzzy-hierarchical, hierarchical and Kelly methods converge in 500, 2500 and 25000 iterations respectively i.e. the fuzzy-hierarchical method is converged 50 times faster than the Kelly’s method and 5 times faster than the plain hierarchical method.

In Figs. 11–13, the aggregate rate of users that traverse the bottleneck links are compared for these three different methods. For example, in Fig. 11, the fuzzy-hierarchical, hierarchical and Kelly methods converge in 2000, 12000 and 210000 iterations respectively i.e. the fuzzy-hierarchical
method is converged 60 times faster than the Kelly’s method and 6 times faster than the plain hierarchical method.

It can be easily verified that the rate of convergence of the proposed methods are higher than that of Kelly and the fuzzy-hierarchical method is superior to the plain hierarchical method.

6. Conclusions

In this paper, we have proposed a two-level hierarchical technique which allocates proportionally-fair rates to the network users. The convergence speed of the proposed algorithm is improved using a combination of high-speed algorithms (such as the Jacobi or approximate Newton method) and fuzzy techniques in the higher level of the hierarchy. The higher level of the hierarchy is composed of those network links that are shared by some subsets of end-users and are subject to the congestion. Also users are grouped into virtual users such that each virtual user is consisted of some end-users traversing the same route in the higher-level. In the proposed algorithm, each lower-level link computes its penalty function once every $N$ iterations based on the aggregate flow that passes through it. This fact leads to the reduction of the computation and transmission overheads in this level of hierarchy. On the other hand, since the high-speed algorithms are implemented only in higher level of hierarchy which is less complex than the whole network, increased computational complexity due to such fast algorithms is negligible.

Acknowledgments

The authors want to express their gratitude to Mr. M. Mojiri and Mr. M.A. Khosravifard for their collaboration in preparing the final manuscript.

References


Appendix A: Proof of Theorem 1

We follow the lines of proof, which is presented in [2]
with some modifications. The assumptions on \( W_i > 0, \)
\( i = 1, 2, \ldots, Q \) and \( p_j(.) \), \( j \in J_i \) ensure that \( V(\chi) \) is strictly concave on \( \chi \geq 0 \) with an interior maximum; the maximizing value of \( \chi \) is thus unique [2]. Observe that:
\[
\frac{\partial}{\partial \chi_i} V(\chi) = \frac{W_i}{\chi_i} - \sum_{j \in R_i} p_j \sum_{s \in R_i} \chi_s
\]

(A-1)

Setting these derivatives to zero identifies the maximum. Furthermore:
\[
\frac{d}{dt} V(\chi(t)) = \frac{d}{dt} \left( \frac{\partial}{\partial \chi_i} V(\chi) \right)
\]
\[= \sum_{i=1}^{Q} K_i(t) \cdot \left[ W_i - \chi_i \cdot \sum_{j \in R_i} p_j \left( \sum_{s \in R_i} \chi_s \right) \right]^2 \chi_i \cdot \frac{w_i}{\chi_i} + \chi_i \cdot \frac{\partial \Lambda}{\partial \chi_i}
\]
\[= \left( \frac{w_i}{\chi_i} + \chi_i \cdot \frac{\partial \Lambda}{\partial \chi_i} \right) \sum_{j \in R_i} p_j \sum_{s \in R_i} \chi_s
\]

(A-2)

As \( K_i(t), \ W_i, \ \chi_i \) and \( \partial \Lambda_i/\partial \chi_i \) are all positive parameters, it can be easily verified that:
\[
\frac{d}{dt} V(\chi(t)) \geq 0, \quad \forall \chi > 0
\]

(A-3)

Also \( V(\chi(t)) = 0 \) if and only if for each \( i \) we have:
\[
\chi_i = \frac{W_i}{\sum_{j \in R_i} p_j \left( \sum_{s \in R_i} \chi_s \right)}
\]

(A-4)

Thus, the function \( V(\chi(t)) \) is a Lyapunov function for the system and \( \chi(t) = \chi^* \) (the solution vector of the above equation) is the stable equilibrium point of Eq. (19) to which all the trajectories converge.

Now, we consider the discrete-time system (16). The vector \( \chi \) is an equilibrium point for the system (16) if and only if it solves:
\[
W_i = \chi_i \cdot \sum_{j \in R_i} p_j \left( \sum_{s \in R_i} \chi_s \right)
\]

(A-5)

But this is precisely the stationary condition implied by the partial derivatives (A-1) of the function \( V(\chi) \), a strictly concave function with a unique maximum.

**Appendix B: Proof of Theorem 2**

It can simply be verified that Eq. (17) is equivalent to:
\[
w_{u[n + 1]} = \begin{cases} 
\left\{ w_{u[n]} + \alpha \cdot \left( \frac{w_{u[n]} - \log w_{u[n]}}{w_{u[n]}} \right) \right\}^+ & \text{for } n = 0, N, \ldots \\
\text{otherwise} & \text{otherwise} \\
w_{u[n]} & i = 1, 2, \ldots, Q, u \in i
\end{cases}
\]

(A-6)

We re-write Eq. (A-6) in the form:
\[
w_{u[n + 1]} = \begin{cases} 
\left\{ w_{u[n]} + \alpha \cdot \Psi_i(w_{u[n]}) \right\}^+ & \text{for } n = 0, N, \ldots \\
\text{otherwise} & \text{otherwise}
\end{cases}
\]

(A-7)

Where:
\[
\Psi_i(w_{u[n]}) = \frac{\alpha \cdot \log w_{u[n]}}{w_{u[n]}}
\]

(A-8)

In continuous-time, by differentiating \( \Psi_i(w_u(t)) \) with respect to \( w_u(t), v \in j \) and by considering the fact that \( \frac{\partial \Lambda_i}{\partial \chi_i} > 0, \ u \in i \) we have for \( t > t_0^i \):
\[
\frac{d}{dt} \psi_i(w_u(t)) = \begin{cases} 
\frac{1}{\Lambda_i(t)} \frac{\partial \psi_i(w_u(t))}{\partial w_u(t)} & \text{if } i = j, u = v \\
0 & \text{if } i \neq j \text{ or } u \neq v
\end{cases}
\]

(A-9)

Now, consider the following continuous-time systems for \( t > t_0^i \):
\[
\begin{align*}
\frac{d}{dt} w_u(t) &= \frac{1}{N} \cdot \alpha \cdot \Psi_i(w_u(t)) \\
\frac{d}{dt} p_j(t) &= \frac{K_i(t) \cdot \left[ W_i - \chi_i \cdot \sum_{j \in R_i} p_j(t) \right]}{w_i(t) + \chi_i \cdot \frac{\partial \Lambda_i}{\partial \chi_i}}
\end{align*}
\]

(A-10)

(A-11)

Where, \( N \) is a constant which is already defined in Eq. (17) and \( R_i \) denotes the path that is traversed by virtual user \( i \) in backbone and \( p_j(y) \) is a non-negative, continuous, increasing function of \( y \). We define:
\[
k_i(\chi) = \frac{K_i(t)}{w_i(t) + \sum_{j \in R_i} \Psi_i(w_u(t))}
\]

(A-12)

Furthermore, assume:
\[
k_i(\chi) > 0 \text{ and } W_i(t) > 0, \quad i = 1, 2, \ldots, Q
\]

(A-13)

And consider the functions:
\[
V_1(W(t)) = \frac{\sum_{i=1}^{Q} \left( \Psi_i(w_{u[i]}(t)) \right)^2}{2}
\]

(A-14)

Now, we show that the following function is a Lyapunov function for the system (A-10), (A-11):
\[
V(\chi, W(t)) = V_1(W(t)) - \Psi_1(\chi)
\]

(A-15)

We can write for \( i = 1, 2, \ldots, Q \):
\[
\frac{\partial^2 V_2(\chi)}{\partial \chi_i^2} = -\frac{W_i}{\chi_i} - \sum_{j \in R_i} p_j \left( \sum_{s \in R_i} \chi_s \right) < 0
\]

(A-16)

It is also clear from Eq. (A-13) that:
\[
\frac{\partial^2 V_1(W(t))}{\partial \chi_i^2} \approx 0, \quad i = 1, 2, \ldots, Q
\]

(A-17)
On the other hand, from assumption (21) and relation (A·13), we have:

\[
\frac{1}{2} \frac{\partial^2 V_i(W(t))}{\partial w_i^2(t)} = \frac{\partial}{\partial w_i(t)} \left\{ \Psi_i(w_i(t)) \cdot \frac{\partial \Psi_i(w_i(t))}{\partial w_i(t)} \right\} = \left( \frac{\partial \Psi_i(w_i(t))}{\partial w_i(t)} \right)^2 + \Psi_i(w_i(t)) \frac{\partial^2 \Psi_i(w_i(t))}{\partial w_i^2(t)} \tag{A·18}
\]

But, it can be easily verified from (21), (A·9) that:

\[
\frac{\partial^2}{\partial w_i^2(t)} \Psi_i(w_i(t)) \approx 0, \quad i = 1, 2, \ldots, Q, \ u \in i \tag{A·19}
\]

From (A·18), (A·19) it can be seen that:

\[
\frac{\partial^2 V_i(W(t))}{\partial w_i^2(t)} = 2 \left( \frac{\partial}{\partial w_i(t)} \Psi_i(w_i(t)) \right)^2 > 0 \quad \forall u \in i, \ i = 1, 2, \ldots, Q \tag{A·20}
\]

From relation (A·14), we can verify that:

\[
\frac{\partial^2}{\partial u^2(t)} V_2(\chi) = 0, \forall \ u \in i, \ i = 1, 2, \ldots, Q \tag{A·21}
\]

Equations (A·16)–(A·21) indicate that the function (A·15) is strictly convex in terms of its argument vector \((\chi, W(t))\) \(\geq 0\) with an interior minimum. We also can write for \(t > n_0\):

\[
\frac{d}{dt} V_1(W(t)) = \sum_{i=1}^{Q} \sum_{u \in i} \left( \frac{\partial V_i}{\partial \Psi_i(w_i(t))} \cdot \frac{d}{dt} \Psi_i(w_i(t)) \right) = \sum_{i=1}^{Q} \sum_{u \in i} \left( \frac{\partial V_i}{\partial \Psi_i(w_i(t))} \cdot \frac{\partial \Psi_i(w_i(t))}{\partial w_i(t)} \right) - \sum_{i=1}^{Q} \sum_{u \in i} \left( \frac{2\alpha_u}{N} \cdot \left( \Psi_i(w_i(t)) \right)^2 \cdot \frac{\partial \Psi_i(w_i(t))}{\partial w_i(t)} \right) \leq 0 \tag{A·22}
\]

Equality (a) rises from the chain rule and relation (A·9), equality (b) rises from Eq. (A·10) and (c) results from (A·9) and (A·13).

By considering assumption (21) it can be shown by relation (A·22) that Eq. (A·13) is a Lyapunov function for the system (A·10), hence \(w_i\) parameters converge independent of \(\chi\); parameters for each \(i\) and \(u\), i.e. we can write:

\[
\lim_{t \to \infty} \frac{d}{dt} w_i(t) = 0, \quad \forall u \tag{A·23}
\]

From continuous-time version of (15) we have:

\[
\lim_{t \to \infty} \frac{d}{dt} W_i(t) = \lim_{t \to \infty} \sum_{u \in i} \frac{d}{dt} w_i(t) = 0, \quad \forall i \tag{A·24}
\]

And from chain-rule:

\[
\frac{d}{dt} V_2(\chi(t)) = \sum_{i=1}^{Q} \frac{\partial V_2}{\partial \chi_i} \cdot \frac{d}{dt} \chi_i(t) + \sum_{i=1}^{Q} \frac{\partial V_2}{\partial W_i} \cdot \frac{d}{dt} W_i(t) \tag{A·25}
\]

Thus, as \(t\) goes to infinity, we can conclude from (A·24) and (A·25) that:

\[
\frac{d}{dt} V_2(\chi(t)) \to \sum_{i=1}^{Q} \frac{\partial V_2}{\partial \chi_i} \cdot \frac{d}{dt} \chi_i(t) = \sum_{i=1}^{Q} k_i(\chi) \cdot \frac{1}{\chi_i} \tag{A·26}
\]

Furthermore, we can write:

\[
\frac{d}{dt} V = \frac{d}{dt} V_1 - \frac{d}{dt} V_2 \tag{A·27}
\]

Thus, as \(t \to \infty\) we have:

\[
\frac{d}{dt} V(\chi, W(t)) \to \frac{1}{N} \sum_{i=1}^{Q} \sum_{u \in i} \left( \frac{2\alpha_u}{N} \cdot \left( \Psi_i(w_i(t)) \right)^2 \cdot \frac{\partial \Psi_i(w_i(t))}{\partial w_i(t)} \right) - \sum_{i=1}^{Q} k_i(\chi) \cdot \frac{1}{\chi_i} \quad \leq 0 \tag{A·28}
\]

Above relations prove the global asymptotical stability of system (A·10), (A·11) and show that eventually \(V(\chi, W(t))\) is a Lyapunov function for system (A·10), (A·11) as \(t \to \infty\) for any initial conditions. \(V(\chi, W(t))\) is strictly convex in terms of its arguments and is a strictly decreasing function of \(t\) as \(t \to \infty\) unless \((\chi, W(t)) = (\chi^*, W^*)\) the unique \((\chi, W)\) minimizing \(V(\chi, W(t))\) which results when we have the followings:

\[
\Psi_i(w_i(t)) = 0 \tag{A·29}
\]

\[
W_i(t) = \chi_i(t) \cdot \sum_{j \in R_i} p_j \left( \sum_{x \in R_i} \chi_x(t) \right) \leq 0 \tag{A·30}
\]

Equations (A·29) and (A·30) lead the rate allocation algorithm to unique rates. Although the above proof is for continuous-time systems, as we saw in Theorem 1, the system (16), (17) converge to the same unique stable point as the system (A·10), (A·11). From Eqs. (A·8), (A·29) and (A·30) and the fact that the solution (7) that Kelly et al. propose, is unique, optimal and proportionally-fair [2], it can be concluded that the user rates have converged to the optimal rates.

\[ \square \]

Appendix C: Proof of Theorem 3

We can re-write Eqs. (A·6) or (17) at the update instants in
the following form (we associate variable \( n = 0, N, 2N, \ldots \) in Eq. (17) with the variable \( n = 0, 1, 2, \ldots \)):

\[
w_u[n + 1] = w_u[n] + \alpha_u \cdot \left( 1 - \frac{\lambda^1_u[n] \cdot w_u[n]}{\omega_u \cdot \Lambda^1_u[n]} \right)
\]

\[n = 0, 1, 2, \ldots \] (A·31)

Where:

\[
w_u[0] = \omega_u, \ u \in i, \ i = 1, 2, \ldots , Q
\]

As it can be seen in Eq. (A·31), we have ignored the \( \{ \cdot \}^+ \) operator in this equation, but we will show that under assumption (22), \( w_u[n] \) remains always non-negative if \( \alpha_u \) remains lower than its associated upper bound in (23).

Using mathematical induction and assumption (21) we can write:

\[
w_u[n + 1] = (1 - \alpha_u \cdot \theta_u)^{n+1} \omega_u + \alpha_u \cdot \sum_{i=0}^{n} (1 - \alpha_u \cdot \theta_u)^i
\]

\[n = 0, 1, 2, \ldots \] (A·32)

Where:

\[
w_u[0] = \omega_u, \ \theta_u \overset{\Delta}{=} \frac{\lambda^1_u}{\omega_u \cdot \Lambda^1_u}, \ u \in i, \ i = 1, 2, \ldots , Q
\]

In which we have assumed from relation (21) that:

\[
\frac{\lambda^1_u[n]}{\Lambda^1_u[n]} \equiv \frac{\lambda^1_u}{\Lambda^1_i}, \ \forall n
\]

For ensuring the convergence of Eq. (A·32) we must have:

\[
| 1 - \alpha_u \cdot \theta_u | < 1
\]

Equivalently we have:

\[
0 < \alpha_u < \frac{2}{\theta_u}
\] (A·34)

As we previously assumed that, \( 0 < \alpha_u < \delta_u \), we can write:

\[
\delta_u = \frac{2}{\theta_u}
\] (A·36)

Now, we investigate the reason why we have assumed that \( w_u[n] \) remains non-negative. We consider two separate cases:

Case 1:

\[
0 < \alpha_u \leq \frac{1}{\theta_u}
\] (A·37)

In this case, it is clear from (A·32) that we always have:

\( w_u[n] > 0 \) for each \( u \) and \( n \).

Case 2:

\[
\frac{1}{\theta_u} < \alpha_u < \frac{2}{\theta_u}
\] (A·38)

From assumption (22) we can re-write inequality (A·38) in the form:

\[
\omega_u \cdot \frac{\Lambda^1_u}{\Lambda^1_i} < \omega_u < 2 \omega_u \cdot \frac{\Lambda^1_u}{\Lambda^1_i} \Rightarrow \omega_u < 2 \omega_u
\] (A·39)

Equation (A·32) is equivalent to:

\[
w_u[n + 1] = (1 - \alpha_u \theta_u)^{n+1} \omega_u + \alpha_u \left( 1 - (1 - \alpha_u \theta_u)^{n+1} \right)
\]

\[n = 0, 1, 2, \ldots \] (A·40)

Consider Eq. (A·40), for odd \( n = 2k + 1 \) and from relation (A·38) we can conclude that: \( w_u[n + 1] > 0 \) for each \( u \) and \( n = 2k + 1, \ k = 0, 1, 2, \ldots \).

For even \( n = 2k \), we re-write Eq. (A·32) in the following form:

\[
w_u[n + 1] = (1 - \alpha_u \theta_u)^{n+1} \omega_u + \alpha_u \left( 1 - \alpha_u \theta_u \right)^n
\]

\[n = 2k, \ k = 1, 2, \ldots \] (A·41)

Remark: If we assume that \( -1 < d < 0 \), then the following inequality is true:

\[d^{2k} + d^{2k+1} > 0 \text{ for } k = 0, 1, 2, \ldots \]

The reason is that from the assumption \( -1 < d < 0 \), we conclude that \( 1 + d > 0 \) and as \( d^{2k} > 0 \) for \( k = 0, 1, 2, \ldots \), we can write:

\[d^{2k} + d^{2k+1} = d^{2k} (1 + d) > 0 \text{ for } k = 0, 1, 2, \ldots \]

From relation (A·38) and the above remark and the fact that \( \alpha_u \) is positive, we can write:

\[\alpha_u \sum_{i=0}^{n-1} (1 - \alpha_u \cdot \theta_u)^i > 0, \ n = 2k, \ k = 1, 2, \ldots \] (A·42)

Thus, it is sufficient to check that the following expression in the RHS of Eq. (A·41) is positive:

\[\ell[k] = (1 - \alpha_u \cdot \theta_u)^{2k} \omega_u + \alpha_u \left( 1 - \alpha_u \cdot \theta_u \right)^{2k} \]

\[k = 0, 1, 2, \ldots \] (A·43)

For \( \omega_u \leq \alpha_u < 2 \omega_u \), it is clear from (A·38) and (A·43) that: \( \ell[k] > 0 \).

For \( \frac{\omega_u}{2} < \alpha_u < \omega_u \), we re-write Eq. (A·43) in the form:

\[\ell[k] = (1 - \alpha_u \cdot \theta_u)^{2k} \left[ \omega_u \cdot (1 - \alpha_u \cdot \theta_u) + \alpha_u \right] \]

\[k = 0, 1, 2, \ldots \] (A·44)

Equivalently, we have:

\[\ell[k] = (1 - \alpha_u \cdot \theta_u)^{2k} \left[ \omega_u + \alpha_u \cdot \left( 1 - \frac{\lambda^1_u}{\Lambda^1_i} \right) \right] \]

\[k = 0, 1, 2, \ldots \] (A·45)

From (22) and the fact \( \frac{\omega_u}{2} < \alpha_u < \omega_u \), we can conclude from (A·45) that for each \( k \), \( \ell[k] > 0 \).

Thus, for \( \frac{\omega_u}{2} < \alpha_u < \frac{\omega_u}{2} \) or \( \frac{\omega_u}{4} < \alpha_u < 2 \omega_u \) (from (22)) and for all \( n \), we have: \( w_u[n] > 0 \).
In equilibrium, from Eq. (A.40), the stabilized value of $w_u$ will be:

$$w_u^\Delta \triangleq \lim_{n \to \infty} w_u[n] = \frac{1}{\theta_u} = \frac{\delta_u}{2} = \omega_u \cdot \Lambda_u^\Delta \cdot \lambda_u^\Delta,$$

(A.46)

\[ u \in i, \ i = 1, 2, \ldots, Q \]

---

**Pejman Gudarzi** was born in Shiraz-IRAN in 1972. He received the B.S. degree in Electronics from Sharif University of Technology Tehran-Iran in 1995. He also received his M.S. degree in Communications from Isfahan University of Technology (IUT), Isfahan-Iran in 1998. Currently he is working toward a Ph.D. in Electrical Engineering in the IUT. Mr. Gudarzi is also with the Iranian Telecommunication Research Center (ITRC). His research interests are multiple access techniques in mobile networks, distributed rate allocation algorithms and congestion control in data networks.

**Hossein Saidi** received the B.S. and M.S. degrees in Electrical Engineering in 1986 and 1989, respectively, both from Isfahan University of Technology (IUT), and D.Sc. from Washington University in St. Louis, USA. Since 1995 he has been with the Dept. of Electrical and Computer Engineering at IUT, Isfahan-Iran. His research interests are high-speed switching and networking, queueing and information theory.

**Farid Sheikholeslam** received the B.S. degree in Electronics from Sharif University of Technology Tehran-Iran in 1990. He also received his M.S. degree in Communications and a Ph.D. in Electrical Engineering from Isfahan University of Technology (IUT), Isfahan-Iran in 1994 and 1998 respectively. Since 1999 he has been with the Dept. of Electrical and Computer Engineering at IUT. His research interests are control algorithms, fuzzy systems, neural networks, stability analysis and nonlinear systems.