Distributed Algorithms for Nash Equilibria of Flow Control Games [1,2]

Tansu Alpcan[3] and Tamer Başar[4]

3rd January 2002

Abstract

We develop a mathematical model within a game theoretical framework to capture the flow control problem for variable rate traffic at a bottleneck node. In this context, we also address various issues such as pricing and allocation of a single resource among a given number of users. We obtain a distributed, end-to-end flow control using cost functions defined as the difference between particular pricing and utility functions. We prove the existence and uniqueness of a Nash equilibrium for two different utility functions. The paper also discusses three distributed update algorithms, parallel, random and gradient update, which are shown to be globally stable under explicitly derived conditions. The convergence properties and robustness of each algorithm are studied through extensive simulations.

1 Introduction

Today's Internet strives to comply with the demands of a broad range of applications, much different from the original design goals. Implementation of RTT (Real-Time Traffic) for increasingly popular applications like VoIP (Voice over IP) or video conferencing is one such example. Pricing of the network resources and charging the users in a way proportional with their usage is yet another one. There is also an increasing need for mechanisms that ensure fair allocation of network resources among the users. Achieving all these goals is only possible with the introduction of new efficient and real-time implementable congestion and flow control schemes, and we are already seeing an effort toward improving and modifying the flow control mechanisms of TCP. Among many different approaches here, the game theoretic one has been enjoying increasing popularity as it provides a fitting framework to study the underlying network optimization problems [1, 2, 3, 4].

[1]Research supported in part by grants NFS ANI 98-13710, AFOSR MURI AF DC 5-36128, and ARMY OSP 35352-6086.

[2]An early version was presented at the Ninth International Symposium on Dynamic Games and Applications, Adelaide, South Australia, December 18-21, 2000.

[3]Coordinated Science Laboratory, University of Illinois, 1308 West Main Street, Urbana, IL 61801 USA. E-mail: alpcan@control.csl.uiuc.edu

[4]All correspondence should be forwarded to: Prof. Tamer Başar, Coordinated Science Laboratory, University of Illinois, 11308 West Main Street, Urbana, IL 61801-2307 USA. Tel: (217) 333-3607; Fax: (217) 244-1653; E-mail: tbasar@control.csl.uiuc.edu

Game theory provides a natural framework for developing pricing mechanisms to solve rate control, fairness and even routing problems. Users on the Internet are of completely noncooperative nature in terms of their demands for bandwidth, and this leads specifically to the use of noncooperative game theory for flow and congestion control. An appropriate solution concept here is the noncooperative Nash equilibrium [5]. In this approach, a distributed, noncooperative network game is defined, where each user tries to minimize a specific cost function by adjusting his flow rate, with the remaining users' flows fixed. An advantage of this approach comes from the fact that it leads to distributed schemes, and not a centralized control for the network, which fits well with today's as well as tomorrow's expected trends of decentralized computing.

Most network games in the literature are focused on elastic, best-effort type traffic [2]. As an example of a study that addresses the flow control problem in a game theoretic framework, we can cite Altman and Başar [1], who show that if an appropriate cost function and pricing mechanism are used, one can find an efficient Nash equilibrium for a multiuser network, which is further stable under different update algorithms. Another game-theoretic study is the one by Korilis and Lazar [6], who investigate the existence of Nash equilibria of the flow control model introduced earlier in [7]. They develop a general approach to study the existence of Nash equilibria by using the concept of best reply correspondence.

We consider in this paper a more general model, with two components. The first pertains to a classical admission control mechanism [8, p.494], where the users are admitted to the network after given a certain QoS (Quality of Service) guarantee in accordance with the available resources of the network. The guaranteed minimum flow rate meets the requirements of the intended RTT application. The second component of the model concerns elastic flow, and it accommodates a wide range of traffic types, from medium to high elasticity. The distributed end-to-end control system is modeled as a network game where users, or players, adjust their excess flows rates, or strategies, according to their individual needs but also by taking into account the state of the network. The cost function we adopt for this purpose features, in addition to a relevant pricing function, an inherent feedback mechanism, enabling the users to acquire the basic (essential) information about the state of the network. The noncooperative game framework provides equilibrium conditions for the system and most importantly, the market structure, where supply and demand for bandwidth determine the allocation of network resources and prices. Fairness is also an important issue, and this is built into the network so that those users who are willing to pay for resources more than others receive a proportionately larger portion of the resources. We model the individual user's demand for bandwidth in terms of two different utility functions: an affine and a logarithmic function.

In the next section, we provide a description of the proposed model, and in section 3 we derive a unique Nash equilibrium. In section 4, we investigate various update algorithms and establish stability conditions. Simulation results are presented in section 5, which are followed by the concluding remarks of section 6.

2 The Model and The Cost Function

We consider a bottleneck node in a general network topology, with a certain level, C, of available bandwidth, which is shared by N users or connections. The i^{th} user's flow rate, λ_i, consists of two parts: The guaranteed minimum flow rate, $\lambda_{i,min}$, and the variable excess flow rate, x_i, defined as the difference between the total flow and the minimum flow: $x_i = \lambda_i - \lambda_{i,min}$. The guaranteed flow rate is negotiated between the user and the network at the time of the connection setup and remains constant thereafter. It plays a crucial role in meeting the QoS requirements necessary for real time traffic types. The problem of giving users guarantees for their requested minimum flows while at the same time preserving the network resources by respecting the bound C on the maximum available bandwidth at the bottleneck node, can be solved through an admission control mechanism.

We assume that during the connection M out of N users request an excess bandwidth on top of their prenegotiated guaranteed minimum flow rates. Hence, we can consider a noncooperative network game where M users compete for the remaining available bandwidth, m, after all guaranteed minimum flows are subtracted from the total capacity: $m = C - \sum_{i=1}^{N} \lambda_{i,min}$. We note that the excess flow rate of a user is elastic in nature, i.e. it has no QoS guarantees and is bounded above by the total available excess bandwidth, m.

The cost function for each user entering the game is defined as the difference between the pricing and the utility functions. This cost function not only sets the dynamic prices, but also captures the demand of a user for bandwidth. The first term of the cost function, the pricing function, is defined as:

$$P_i(x_i, x_{-i}) = \frac{k_i}{m - (x_i + x_{-i})^+}(x_i)^2 \ , \tag{2.1}$$

where x_i is the excess bandwidth taken by the i^{th} user, x_{-i} is defined as $x_{-i} := \sum_{j \neq i} x_j$, and $k_i \geq 0$ is the pricing parameter determined by the network. The pricing term not only sets the actual price, but also has the regulatory function of providing the user with feedback about the network status via the denominator term, $m - \sum_j x_j$. This term may also be seen as capturing the delay experienced by the i^{th} user. As the sum of flows of users approach the available capacity, m, the denominator of (2.1) approaches zero, and hence the price increases without bound. This preserves the network resources by forcing the users to decrease their elastic flows. At the same time, a proportional relationship between demand and price is obtained, which ensures that the prices are set according to market forces. The price per unit flow is chosen to be proportional to the flow rate itself, resulting in the quadratic term, x_i^2. This pricing structure discourages a user to pick an arbitrary value for the minimum flow rate, $\lambda_{i,min}$, with the intention of minimizing the fixed costs. In addition, it eliminates wide fluctuations in the excess rate, x_i.

The second part of the cost function, the utility function U_i, quantifies the user's utility for having excess bandwidth and captures to some extent the 'human factor'. Although it cannot be exactly known to the network, some statistical estimates can be collected, taking into account habits of specific type of a user over a certain period of time. A reasonable assumption is to take it to be strictly concave for elastic flows. Specifically, a logarithmic function can best represent the utility function of the user in this case [9]. Hence, a possible realistic utility function for a user demanding excess flow can be defined in terms of x_i as:

$$U_i(x_i) = \ln(1 + x_i) + d_i \quad , x_i \geq 0 \ \ \forall i, \tag{2.2}$$

where d_i is a positive constant and will have no effect in the optimization process. Based on the given pricing and utility functions, the cost function for user i is simply, $J_i = P_i - U_i$. In other words, the flow rate of the i^{th} user results from the interaction between price and demand:

$$J_i(x_i, x_{-i}) = \frac{k_i x_i^2}{m - (x_i + x_{-i})} - \ln(1 + x_i) - d_i, \quad x_i \geq 0 \tag{2.3}$$

Notice that the excess utility function, and hence the cost function, are defined only in the region where $x_i \geq 0$. In the next section we will show that even if $x_i \leq 0$ is allowed, the Nash equilibrium solution will still be the nonnegative.

A drawback of the realistic utility function above is that it leads to nonlinear reaction functions for the users. Therefore an analytical analysis of the cost function with this utility is very difficult and limited, if not impossible, even though an existence and uniqueness result (on Nash equilibria) could be obtained, as we will do in the next section. In order to make the analysis tractable, and to obtain explicit results, we will use linear utility functions for the users, which lead to a set of linear equations as reaction functions. Accordingly, for tractability we will also adopt as the utility function of user i:

$$\tilde{U}_i(x_i) = a_i x_i + \tilde{d}_i \quad , x_i \geq 0 \ \forall i, \tag{2.4}$$

where a_i is a positive constant not exceeding 1, and $\tilde{d}_i > 0$. One possible interpretation for the linear utility $\tilde{U}_i$ is that it constitutes a linear approximation to the actual utility function at a given point. In this case, the system is analyzed locally in the vicinity of the chosen point. The parameter a_i is the slope of the utility function at that point, say x_i^0:

$$a_i := \frac{\partial U_i(x_i^0)}{\partial x_i} \ \Rightarrow 0 < a_i \leq 1 \quad , \forall i \ .$$

The counterpart of the cost function (2.3) with the linear utility function (2.4) is

$$\tilde{J}_i(x_i, x_{-i}) = \frac{k_i x_i^2}{m - (x_i + x_{-i})} - a_i x_i - \tilde{d}_i, \ \ x_i \geq 0 \ . \tag{2.5}$$

Another interpretation for the linear utility would come from a worst-case perspective. The constant a_i can be chosen so as to provide an upper bound for marginal utility, or the slope of the logarithmic function at any given point x_i. It will be shown later that given the total flow rates of all other users, x_{-i}, the optimal flow rate of the i^{th} user under linear utility, with $a_i = 1$, is always higher than the one under logarithmic utility. Hence, linear utility (2.4) with $a_i = 1$ leads to a worst-case flow for the i^{th} user, namely

$$a_i = \max_{x_i \geq 0} \frac{\partial U_i(x_i)}{\partial x_i} \Longleftrightarrow a_i = \max_{x_i \geq 0} \frac{1}{1 + x_i} \Rightarrow a_i = 1, \quad \forall i$$

The parameter $\tilde{d}_i$ in (2.4) is the same as in (2.2), and since it is a constant it can be ignored in the subsequent analysis. Notice that the same cost function structure (2.5) is arrived at in both local and worst-case analyses, with a_i chosen as described above. Combining the worst-case and local analyses in a single step simplifies the problem at hand significantly.

3 Existence and Uniqueness of a Nash Equilibrium

We first prove the existence of a unique Nash equilibrium for the logarithmic utility model, described by the cost function (2.3). Next, we give a similar result under the linear-utility cost function (2.5). Additionally, we derive the reaction functions in the linear-utility case and compute the equilibrium point explicitly. We conclude the section with a result (a Proposition) justifying the worst-case linear-utility function analysis.

3.1 Existence and Uniqueness under Logarithmic Utility Functions

The underlying M-player noncooperative game here is defined in terms of the cost functions $J_i(x_i, x_{-i})$, $i = 1, \ldots, M$, where J_i is defined by (2.3), and the constraints

$$x_i \geq 0 \tag{3.1}$$

$$\sum_{j=1}^{M} x_j \leq m \ . \tag{3.2}$$

The first constraint is dictated by the fact that the i^{th} user has requested a flow rate of at least $\lambda_{i,min}$. The second constraint is a physical capacity constraint, which says that the aggregate sum of all flows in a node cannot exceed its total capacity.

Theorem 3.1. *There exists a unique Nash equilibrium in the network game defined by (2.3) and (3.1)-(3.2), which is also an inner solution.*

Proof. Let $x := (x_1, \ldots, x_M)'$ be the vector of flow rates of the M players, and X be the subset of $\mathbb{R}^M$ where x belongs in view of the constraints (3.1)-(3.2). Then, X is closed and bounded (therefore compact), and is convex, but is not rectangular. On X, each $J_i(x_i, x_{-i})$ is continuous and bounded, except on the hyperplane defined by (3.2) where it is infinite, and is moreover analytic in that region. Its first derivative with respect to x_i is

$$\frac{\partial J_i(x_i, x_{-i})}{\partial x_i} = \frac{k_i x_i^2 + 2k_i x_i [m - (x_i + x_{-i})]}{[m - (x_i + x_{-i})]^2} - \frac{1}{1 + x_i} \ , \tag{3.3}$$

and its second derivative is

$$\frac{\partial^2 J(x_i, x_{-i})}{\partial x_i^2} = \frac{4k_i x_i}{[m - (x_i + x_{-i})]^2} + \frac{2k_i x_i^2}{[m - (x_i + x_{-i})]^3} + \frac{2k_i}{[m - (x_i + x_{-i})]} + \frac{1}{(1 + x_i)^2} \ , \tag{3.4}$$

which is clearly seen to be well-defined and positive on X, except on the hyperplane (3.2). Hence, for an arbitrarily small $\varepsilon > 0$, if we replace (3.2) with

$$\sum_{j=1}^{M} x_j \leq m - \varepsilon \ , \tag{3.5}$$

and denote the corresponding constraint set defined by (3.1) and (3.5) by X_ε, on X_ε we have a game with a convex, compact nonrectangular action set, and a convex cost function for each player. By a standard theorem of game theory (Theorem 4.4, p. 176 in [5]), it admits a Nash equilibrium. Let us denote this equilibrium solution by x_ε^*, to depict its possible dependence on ε. We now claim that provided that $\varepsilon > 0$ is sufficiently small, x_ε^* does not depend on ε, and hence it provides also a Nash equilibrium solution to the original game on X.

To prove this claim, let us study the optimization problem faced by a generic player, i, in the computation of the Nash equilibrium. This optimization problem is

$$\min_{0 \leq x_i \leq m_i(\varepsilon)} J_i(x_i, x_{-i}^*(\varepsilon)) \ , \tag{3.6}$$

where $m_i(\varepsilon) := m - \varepsilon - x^*_{-i}(\varepsilon)$, and $x^*_{-i}(\varepsilon)$ is the total flow from x^*_ε of all players, except the i^{th}. We first note from (3.3) that $\frac{\partial J_i}{\partial x_i}(0, x^*_{-i}(\varepsilon)) < 0$, and hence $x^*_i(\varepsilon)$ cannot be zero, and therefore $x^*_i(\varepsilon) > 0$. We next show that $x^*_i(\varepsilon)$ cannot equal $m_i(\varepsilon)$ either, and hence the solution to (3.6) has to be an inner solution. To see this, let us evaluate (3.3) at $x_i = m_i(\varepsilon)$:

$$\frac{\partial J_i(m_i, x^*_{-i}(\varepsilon))}{\partial x_i} = \frac{k_i m_i(m_i + 2\varepsilon)}{\varepsilon^2} - \frac{1}{1 + m_i} \tag{3.7}$$

Now, unless $m_i(\varepsilon) = O(\varepsilon)$, the first term above becomes unbounded as $\varepsilon \to 0$, dominating the second term and therefore making (3.7) positive, which implies that $x_i = m_i(\varepsilon)$ cannot be the solution to (3.6). Hence, the only way (3.6) can have a boundary solution is if $m_i(\varepsilon) = O(\varepsilon)$, but since there is only a single constraint, namely (3.5), we would have $m_j(\varepsilon) = O(\varepsilon)$ for all $j = 1, \ldots, M$. Hence the total flow would be $O(\varepsilon)$, which contradicts the initial hypothesis that $\sum_{j=1}^M x^*_j(\varepsilon) = m - \varepsilon$. Therefore, for ε sufficiently small, the solution to (3.6) has to be inner, and since this applies to all players, the Nash equilibrium has to be independent of ε for $\varepsilon > 0$ sufficiently small. As a byproduct, we have also proven the other claim of the theorem, which is that the Nash equilibrium is inner.

We next show the uniqueness of the Nash equilibrium. To preserve notation, let $\frac{\partial^2 J(x_i, x_{-i})}{\partial x_i^2}$ given by (3.4) be denoted by B_i. Further introduce, for $i, j = 1, \ldots, M$, $j \neq i$,

$$\frac{\partial^2 J_i(x_i, x_{-i})}{\partial x_i \partial x_j} = \frac{2k_i x_i}{[m - (x_i + x_{-i})]^2} + \frac{2k_i x_i^2}{[m - (x_i + x_{-i})]^3} := A_{i,j} \ ,$$

with both B_i and $A_{i,j}$ defined on X_ε, to avoid singularity on the hyperplane (3.2). Suppose that there are two Nash equilibria, represented by two flow vectors x^1 and x^0, with elements x_i^0 and x_i^1, respectively. Define the pseudo-gradient vector:

$$g(x) = \begin{pmatrix} \nabla_{x_1} J_1(x_1, x_{-1}) \\ \vdots \\ \nabla_{x_M} J_M(x_M, x_{-M}) \end{pmatrix}$$

As the Nash equilibrium is necessarily an inner solution, it follows from first order optimality condition that $g(x^0) = 0$ and $g(x^1) = 0$. Define the flow vector $x(\theta)$ as a convex combination of the two equilibrium points x^0 , x^1 :

$$x(\theta) = \theta x^0 + (1 - \theta) x^1$$

where $0 < \theta < 1$. By differentiating $x(\theta)$ with respect to θ, we obtain

$$\frac{dg(x(\theta))}{d\theta} = G(x(\theta)) \frac{dx(\theta)}{d\theta} = G(x(\theta))(x^1 - x^0) \ , \tag{3.8}$$

where $G(x)$ is defined as the Jacobian of $g(x)$ with respect to x :

$$G(x) := \begin{pmatrix} B_1 & A_1 & \cdots & A_1 \\ A_2 & B_2 & & A_2 \\ \vdots & & \ddots & \vdots \\ A_M & A_M & \cdots & B_M \end{pmatrix}_{M \times M} \ ,$$

where we have used the simpler notation A_i for $A_{i,j}$, since $A_{i,j}$ does not depend on the second index j. Integrating (3.8) over θ from $\theta = 0$ to $\theta = 1$ we obtain

$$0 = g(x^1) - g(x^0) = \left[\int_0^1 G(x(\theta))d\theta\right](x^1 - x^0) \ , \tag{3.9}$$

where $(x^1 - x^0)$ is a constant flow vector. Let $\overline{B_i(x)} = \int_0^1 B_i(x(\theta))d\theta$ and $\overline{A_i(x)} = \int_0^1 A_i(x(\theta))d\theta$. In view of constraints (3.1) and (3.5), we have:

$$B_i(x_i, x_{-i}) > A_{i,j}(x_i, x_{-i}) > 0 \ , \ \forall i, j \ .$$

Thus, it follows that $\overline{B_i(x)} > \overline{A_i(x)} > 0$, for any flow vector $x(\theta)$.

In order to simplify the notation, define the matrix $\mathcal{G}(x^1, x^0)$

$$\mathcal{G}(x^1, x^0) := \int_0^1 G(x(\theta))d\theta = \begin{pmatrix} \overline{B_1} & \overline{A_1} & \cdots & \overline{A_1} \\ \overline{A_2} & \overline{B_2} & & \overline{A_2} \\ \vdots & & \ddots & \vdots \\ \overline{A_M} & \overline{A_M} & \cdots & \overline{B_M} \end{pmatrix}_{M\times M}$$

Lemma 3.2. *The matrix $\mathcal{G}(x^1, x^0)$ is full rank for any fixed x.*

Proof. The matrix $\mathcal{G}$ is full rank if the only vector y satisfying $\mathcal{G}y = 0$ is the null-vector, $y = 0$. Expanding this equation for all i, one obtains

$$\overline{B_i}y_i + \overline{A_i}\sum_{j\neq i} y_i = 0 \ . \tag{3.10}$$

Rearranging the terms in (3.10) and solving for y_i:

$$y_i = \frac{-\overline{A_i}}{\overline{B_i} - \overline{A_i}} \sum_{j=1}^{M} y_j \ , \tag{3.11}$$

where the term $\overline{A_i}/(\overline{B_i} - \overline{A_i})$ is strictly positive as a result of $\overline{B_i} > \overline{A_i} > 0$. Now summing over i, we get

$$\sum_{i=1}^{M} y_i = -\sum_{i=1}^{M} \frac{\overline{A_i}}{\overline{B_i} - \overline{A_i}} \sum_{j=1}^{M} y_j \ ,$$

which can only hold if $\sum_{j=1}^{M} y_j = 0$, and from (3.11), $y_i = 0 \ \forall i$. Thus, we conclude that the matrix $\mathcal{G}$ is full rank. □

Finally, we rewrite (3.9) in the following form:

$$0 = \mathcal{G} \cdot [x^1 - x^0] \tag{3.12}$$

By Lemma 3.2, the square matrix $\mathcal{G}$ is full rank. Therefore, the only possibility for (3.12) to hold is $x^1 - x^0 = 0$. Therefore, the Nash equilibrium is unique. □

The lines of the proof above can actually be used to prove uniqueness of Nash equilibrium whenever it exists for more general M-player games. We state and prove this result below, which should be of independent interest.

Corollary 3.3. *Consider a noncooperative game with M players, with twice differentiable cost functions, $J_i(x_i, x_{-i})$ for player i, and a compact strategy space, X. Let the pseudo-gradient vector be denoted by $g(x)$, i.e.*

$$g(x) := \begin{pmatrix} \nabla_{x_1} J_1(x_1, x_{-1}) \\ \vdots \\ \nabla_{x_M} J_M(x_M, x_{-M}) \end{pmatrix} ,$$

and the Jacobian of $g(x)$ with respect to x be denoted by $G(x)$. Further define

$$\mathcal{G} := \int_0^1 G(x(\theta)) d\theta \ .$$

If for all strategies that are in the interior of the strategy space, X^o, the matrix $\mathcal{G}$ is full rank, then there is at most one equilibrium in X^o.

Proof. Suppose that there exist two Nash equilibria, represented by two flow vectors x^1 and x^0, with elements x_i^0 and x_i^1, respectively. As we consider only inner solutions, it follows from first order optimality condition that $g(x^0) = 0$ and $g(x^1) = 0$. Define the flow vector $x(\theta)$ as a convex combination of the two equilibrium points x^0 , x^1 :

$$x(\theta) = \theta x^0 + (1 - \theta) x^1$$

where $0 < \theta < 1$. By differentiating $x(\theta)$ with respect to θ, we obtain

$$\frac{dg(x(\theta))}{d\theta} = G(x(\theta)) \frac{dx(\theta)}{d\theta} = G(x(\theta))(x^1 - x^0) \ .$$

Integrating this over θ from $\theta = 0$ to $\theta = 1$ we obtain

$$0 = g(x^1) - g(x^0) = \left[\int_0^1 G(x(\theta)) d\theta \right] (x^1 - x^0) \equiv \mathcal{G} \cdot [x^1 - x^0] \ . \tag{3.13}$$

Since $\mathcal{G}$ is full rank, the only possibility for (3.13) to hold is $x^1 - x^0 = 0$. Thus, the Nash equilibrium is unique whenever it exists. □

3.2 Existence of Unique Nash Equilibrium under Linear Utility Functions

Here, we show the existence of a unique Nash equilibrium for cost functions with linear utility. Furthermore, by exploiting the linearity of reaction functions, we compute the equilibrium point explicitly. Since we carry out the analysis for a general a_i, it applies not only to the worst-case analysis, but also to the local analysis, where the logarithmic utility function is approximated by a linear function.

Again, each user minimizes his cost function (2.3), under the constraints given in (3.1) and (3.2). First assuming an inner solution, we have for the i^{th} user:

$$\frac{\partial \tilde{J}_i(x_i, x_{-i})}{\partial x_i} = \frac{k_i x_i^2 + 2k_i x_i m - 2k_i x_i(x_i + x_{-i})}{(m - (x_i + x_{-i}))^2} - a_i = 0 \;, \tag{3.14}$$

which can be solved for x_i, to lead to: [1]

$$x_i = (m - x_{-i})\left[1 \pm \sqrt{\frac{k_i}{k_i + a_i}}\right]$$

The solution with the plus sign is eliminated in view of the constraint $m - x_{-i} \geq x_i$; hence, the only feasible solution is the one with the minus sign:

$$x_i = (m - x_{-i})[1 - \sqrt{\frac{k_i}{k_i + a_i}}] \equiv mq_i - q_i x_{-i} \;, \tag{3.15}$$

where

$$q_i := 1 - \sqrt{\frac{k_i}{k_i + a_i}} \tag{3.16}$$

To complete the derivation we now check the boundary solutions. For the boundary point $x_i = 0$, we observe from (3.14) that $\frac{\partial \tilde{J}_i(x_i, x_{-i})}{\partial x_i} = -a_i$, which means that the user can decrease his cost by increasing x_i. Hence, this cannot be an equilibrium point. For the other boundary points $x_i = m - x_{-i}$ and $x_{-i} = m$, we observe that at these points the cost goes to infinity. As a result, the inner solution is the unique optimal response for the constrained optimization problem of the i^{th} user, for each fixed $x_{-i} < m$. We observe from (3.15) that the unique optimal flow for the i^{th} user is a linear function of the aggregate flow of all other users. This set of M equations can now be solved for x_i, $i = 1, \ldots, M$. To ease the notation, let $\bar{x} := x_i + x_{-i}$. Then, (3.15) can be rewritten as

$$x_i = mq_i - q_i(\bar{x} - x_i) \Rightarrow x_i = \frac{q_i}{1 - q_i} m - \frac{q_i}{1 - q_i} \bar{x}.$$

Summing both sides from 1 to M, and letting

$$\lambda := \sum_{i=1}^{M} \frac{q_i}{1 - q_i} \;, \tag{3.17}$$

we obtain

$$\bar{x} = \lambda m - \lambda \bar{x} \Rightarrow \bar{x} = \frac{\lambda}{1 + \lambda} m \;.$$

Note that λ is well defined and positive, since $0 < q_i < 1 \; \forall i$. Hence $\bar{x} < m$, thus satisfying the underlying constraint. Finally, substituting $\bar{x}$ above into the expression for x_i (in terms of $\bar{x}$), yields the following unique solution to (3.15):

$$x_i^* = \frac{1}{1 + \lambda} \frac{q_i}{1 - q_i} m \;, \; i = 1, \ldots, M \;. \tag{3.18}$$

[1] Here we assume throughout that $x_i \neq m - x_{-i}$, which will be seen shortly not to be an assumption at all.

Note that (3.18) is feasible since it is strictly positive, and $\sum_{i=1}^{M} x_i^* < m$. We summarize this result in the following theorem, whose proof follows from the leading derivation:

Theorem 3.4. *There exists a unique Nash equilibrium in the network game with users having linear utility functions, and it is given by (3.18), where q_i and λ are given by (3.16) and (3.17), respectively.*

We conclude the section with an important proposition, justifying the worst-case analysis based on linear utility functions.

Proposition 3.5. *Given the total flow rates of all users except the i^{th} one, $x_{-i} = \sum_{j \neq i} x_j$, the optimal flow rate of the i^{th} user, $x_{i,nonlin}^{opt}$, having a logarithmic utility and the cost function (2.3) is less than the optimal rate $x_{i,lin}^{opt}$ obtained when the same user has the linear utility (2.4) with $a_i = 1$ and cost function (2.5).*

Proof. The optimal solution of the i^{th} user was already shown to be an inner solution. Differentiating the linear utility cost, $\tilde{J}_i$, given by (2.5), and the logarithmic utility cost, J_i, given by (2.3), both with respect to x_i, we obtain:

$$\tilde{J}_i^{'} = P_i^{'} - \tilde{U}_i^{'} = P_i^{'} - 1 \tag{3.19}$$

$$J_i^{'} = P_i^{'} - U_i^{'} = P_i^{'} - \frac{1}{1 + x_i} \tag{3.20}$$

where a 'prime' denotes partial derivative with respect to x_i. The pricing function P_i in (2.1) is unimodal with a global minimum at $x_i = 0$. Hence, for $x_i \geq 0$, $P_i^{'}$ is a monotonically increasing function passing through the origin. Hence, the point at which (3.19) is zero is strictly larger than the point at which (3.20) is zero, that is $x_{i,nonlin}^{opt} < x_{i,lin}^{opt}$. This completes the proof. □

An intuitive explanation of this result lies in the high marginal demand of worst-case utility, $a = 1$. The marginal demand of a user with linear utility is higher than the one with logarithmic utility. The proposition above is based on this difference in demand.

4 Update Algorithms and Stability

In the previous section, it was shown that a unique equilibrium point exists under different cost functions, where each user attains a minimum cost given the equilibrium flow rates of the other users. In a distributed environment, however, each user acts independently and convergence to this equilibrium point does not occur instantaneously. Hence it is important to study the evolution of various iterative processes toward the unique equilibrium. In the literature, there exist various iterative update schemes with different convergence and stability properties [2]. We consider here three asynchronous update schemes relevant to the proposed model: PUA, parallel update algorithm, which is also known as the Jacobi algorithm; RUA, random update algorithm, and GUA, gradient update algorithm, also known as Jacobi overrelaxation [10]. For the specific model at hand, individual users do not need to know the specific flow rate of other users, except their sum. This feature is of great importance for possible applications, as it simplifies substantially the information flow within the system.

4.1 Parallel Update Algorithm (PUA)

In PUA, the users optimize their flow rates at each iteration, in discrete time intervals $\dots, n-1, n, n+1 \dots$. If the time intervals are chosen to be longer than twice the maximum delay in the transmission of flow information, it is possible to model the system as an ideal, delay-free one. In a system with delays, users update their flows using the available (delayed) information.

One important feature of PUA is that the users are myopic. They optimize their flow rates based on instant costs and parameters, ignoring future implications of their actions. In a delay-free system, this behavior affects convergence rate adversely as it will be seen in the simulations.

For the cost function (2.3), the players use either nonlinear programming techniques to minimize their cost at each iteration or directly the reaction function. The analytical solution to the optimization problem of the i^{th} user turns out to be the root of the third-order equation:

$$k_i x_i^3 + (2k_i(m - x_{-i}) + k - 1)x_i^2 + (2(k+1)(m - x_{-i}))x_i + [m - x_i]^2 = 0 \tag{4.1}$$

Only one root of this equation, denoted $\tilde{x_i}$, is feasible: $0 < \tilde{x_i} < m$. The closed-form solution for this root is at the same time the reaction function, which is highly nonlinear in contrast to the linear reaction function given by (3.15). As the root of (4.1) involves a complicated expression, we write the nonlinear reaction function of the i^{th} user only symbolically :

$$x_i^{(n+1)} = f(x_{-i}^{(n)}, k_i) \tag{4.2}$$

Stability and convergence of the system is as important as the existence of a unique equilibrium. In an unstable system, the flow rates may oscillate indefinitely if there is a deviation from equilibrium. Or, if the system does not have the global convergence property, there exists the possibility of not reaching the equilibrium at all through an iteration starting at an arbitrary feasible point.

We now study the convergence of PUA for the linear reaction case. The update function for the i^{th} user is (from (3.15)):

$$x_i^{(n+1)} = mq_i - q_i x_{-i}^{(n)} \quad \forall i, n \ , \tag{4.3}$$

where q_i was defined in (3.16). Let $\triangle x_i = x_i - x_i^*$, where x_i^* is the flow rate of the i^{th} user at the Nash equilibrium. Then we have

$$\triangle x_i^{(n+1)} = -q_i \triangle x_{-i}^{(n)} \quad , \forall i \tag{4.4}$$

Let

$$\|\Delta x\| = \max_i |\Delta x_i| \ ,$$

and note that from (4.4):

$$\|\Delta x^{(n+1)}\| \leq (M-1) \max_i |q_i| \, \|\Delta x^{(n)}\|$$

Clearly, we have a contraction mapping in (4.4) if $(M-1)\max_i |q_i| < 1$. Thus, the following sufficient condition ensures the stability of the system with linear utility under the PUA algorithm:

$$|q_i| \leq \frac{1}{M} \ , \ i = 1, \dots, M \tag{4.5}$$

One simple way of meeting this condition is to set $q_i = \frac{1}{M} \quad, i = 1, \ldots, M$. From (3.16), (4.5) translates into the following stability condition on the pricing parameter k_i for each i :

$$k_i \geq \frac{(M-1)^2}{2M-1} a_i \ . \tag{4.6}$$

Notice that these apply not only to the analysis in the linear-reaction case, but also to the local analysis of the nonlinear-utility cost function (2.3). Thus, the system is locally stable and convergent under PUA if the condition above is satisfied. Next, we show that the system not only has local stability and convergence property, but it is also globally stable and convergent for the nonlinear-utility cost. Before making a precise statement of this result and proceeding with the proof, we present the following two useful Lemmas.

Lemma 4.1. *For any feasible point* $\mathbf{x_o} = \mathbf{x^{(n)}}$ *at time* (n)*, let* $x_i^{(n+1)}$ *be the outcome of the* i^{th} *user's nonlinear reaction function (4.2). If* $x_i^{(n)} > x_i^*$*, where* $\mathbf{x}^*$ *is the unique equilibrium, then* $x_i^{(n+1)} \leq x_i^{(n)}$*.*

Proof. Assume that $x_i^{(n+1)} > x_i^{(n)} > x_i^*$. Given the flow rate, $x_i^{(n)}$, of the i^{th} user at the time instant n, we linearize the logarithmic utilility, U_i, given in (2.2) around this value, and turn it into (2.4) by defining

$$a_i(x_i^{(n)}) := \frac{\partial U_i(x_i^{(n)})}{\partial x_i} = \frac{1}{1 + x_i^{(n)}} \ .$$

Thus, the nonlinear reaction function (4.2) is linearized to (4.3) at $x_i^{(n)}$. Using the fact that $a_i(x_i^{(n)}) > \frac{\partial U_i(x_i^{(n+1)})}{\partial x_i}$ and following an argument similar to that used in the proof of Proposition 3.5, the resulting flow, $x_{i,lin}^{(n+1)}$, provides an upper bound on $x_i^{(n+1)}$. Combined with the contraction property of linear reaction function, we obtain:

$$x_i^{(n+1)} < x_{i,lin}^{(n+1)} < x_i^{(n)} \tag{4.7}$$

Obviously, (4.7) contradicts the assumption made, and hence $x_i^{(n+1)} \leq x_i^{(n)}$. ☐

Lemma 4.2. *For any feasible point* $\mathbf{x_o} = \mathbf{x^{(n)}}$ *at time* (n)*, if* $x_i^{(n)} < x_i^*$*, then* $x_i^{(n+1)} \geq x_i^{(n)}$*.*

Proof. The proof is very similar to that of Lemma 4.1. Suppose that $x_i^{(n+1)} < x_i^{(n)} < x_i^*$. Then, it can be shown that $a_i < \frac{\partial U_i(x_i^{(n+1)})}{\partial x_i}$, and $x_{i,lin}^{(n+1)}$ provides a lower bound on $x_i^{(n+1)}$. Again using the contraction property,

$$x_i^{(n+1)} > x_{i,lin}^{(n+1)} > x_i^{(n)}$$

As this contradicts the initial hypothesis, $x_i^{(n+1)} \geq x_i^{(n)}$. ☐

Theorem 4.3. *The PUA is globally convergent and stable for both the linear and logarithmic utility cost functions (2.3) and (2.5), under the sufficient condition (4.6).*

Proof. The convergence result for the linear utility case was obtained above. In principle, it is also possible to derive the global convergence result for the logarithmic utility case, using the same method, but this time the reaction function (4.2). This reaction function was obtained as one of the roots of the 3^{rd} order equation (4.1), and is highly nonlinear. Hence the method based on reaction functions becomes practically intractable. We will, therefore, make use of the local and worst-case analyses to obtain a global convergence result. As in Lemma 4.1, the nonlinear reaction function (4.2) can be linearized to (4.3) at each $x_i^{(n)}$. Also note that the existence of a unique feasible Nash equilibrium, $0 < x_i^* < m$, was already established for the network game.

For any feasible initial point, $\mathbf{x_0}$, we have the following cases for the i^{th} user:

In the first case, $x_i^{(n)} > x_i^*$, where $x_i^{(n)} = x_{i,0}$ is the starting point. According to Lemma 4.1, there are two possibilities: $x_i^* < x_i^{(n+1)} < x_i^{(n)}$ and $x_i^{(n+1)} < x_i^* < x_i^{(n)}$. The former case results in a monotonically decreasing sequence bounded below by x_i^*, whereas the latter case leads to an oscillating sequence around the equilibrium.

In the second case, $x_i^{(n+1)} < x_i^*$, where $x_i^{(n+1)}$ can be considered as the starting point at any time instant $(n+1)$. Again by Lemma 4.2, there are two possibilities: $x_i^* > x_i^{(n+1)} > x_i^{(n)}$ and $x_i^{(n+1)} > x_i^* > x_i^{(n)}$. Similar to the previous case, the former leads to a monotonically increasing sequence bounded above by x_i^*, and the latter results again in an oscillating sequence around the equilibrium. In order to analyze these cases, one can define the relative distance to equilibrium as $\triangle x_i^{(n)} := x_i^{(n)} - x_i^*$.

If $x_i^{(n+1)} > x_i^{(n)}$, then the linearized reaction function at $x_i^{(n)}$ provides an upper bound, $x_{i,linear}^{(n+1)}$, on $x_i^{(n+1)}$, following an argument similar to the one in Proposition 3.5. The fact that $\frac{\partial U_i^{(n)}}{\partial x_i} < a_i^{(n)}$ justifies the given bound. Using the contraction property of linearized reaction function (4.4) and the worst-case bound above, we obtain:

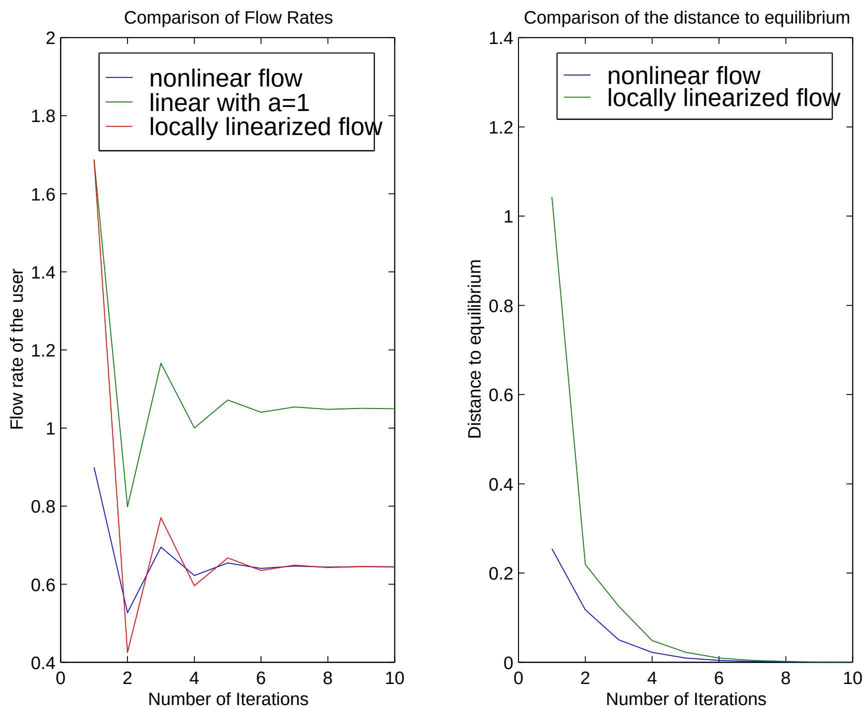
$$\triangle x_{i,linear}^{(n+1)} < \triangle x_i^{(n)} < \triangle x_{i,linear}^{(n)} \quad \forall i \tag{4.8}$$

For $x_i^{(n+1)} < x_i^{(n)}$, following a similar argument, it is easy to show that the relation (4.8) also holds. Therefore, for all possible cases, we have shown that, at each iteration locally linearized flows provide a decreasing upper bound to the iterates of the nonlinear reaction function for the distance to equilibrium. Figure 1 summarizes this discussion graphically.

The flow rate of any i^{th} user converges to the unique Nash equilibrium and the nonlinear system is stable and globally convergent from any feasible initial point $\mathbf{x_0}$. We note that the proof is based on the convergence of the linear system, which is required for the convergence of the nonlinear system. Moreover, condition (4.5), or equivalently (4.6), is sufficient for the convergence of the iteration corresponding to linear and nonlinear reaction functions. □

4.2 Random Update Algorithm (RUA)

Random update scheme is a stochastic modification of PUA. The users optimize their flow rates in discrete time intervals and infinitely often, with a predefined probability p_i, $0 < p_i < 1$, for user i. Thus, at each iteration a random set of users among the M update their flow rates. Again, the users are myopic and make instantaneous optimizations. In the limiting case, $p_i = 1$, RUA is the same as PUA. The non-ideal system with delay is also similar to PUA. The users make decisions based on delayed information at the updates, if the round trip delay is longer than the discrete time interval.



(a) (b)

Figure 1: (a) Comparison of nonlinear, locally linearized: $a_i = \frac{1}{1+x_i}$ and linear worst-case: $a_i = 1$ flow rates, (b) Distance to equilibrium, nonlinear and locally linearized.

For the linear-utility case (2.3) with linear reaction function (3.15), the update scheme may be formulated for the i^{th} user as follows:

$$x_i^{(n+1)} = \begin{cases} mq_i - x_{-i}^{(n)} q_i & , with\ probability\ p_i \\ x_i^{(n)} & , with\ probability\ 1 - p_i \end{cases}$$

Subtracting x_i^* from both sides, we obtain:

$$\triangle x_i^{(n+1)} = \begin{cases} -q_i \triangle x_{-i}^{(n)} & , with\ probability\ p_i \\ \triangle x_i^{(n)} & , with\ probability\ 1 - p_i \end{cases}$$

Taking the absolute value of both sides, and then taking expectations, lead to

$$\begin{aligned} E|\triangle x_i^{(n+1)}| \leq & \ p_i q_i E|\triangle x_{-i}^{(n)}| + (1 - p_i) E|\triangle x_i^{(n)}| \\ \leq & \ p_i q_i \sum_{j=1}^{M} E|\triangle x_i^{(n)}| + (1 - p_i(1 + q_i)) E|\triangle x_i^{(n)}| \end{aligned}$$

Choosing $p_i \geq \frac{1}{1+q_i}$, this expression can further be bounded by

$$E|\triangle x_i^{(n+1)}| \leq p_i q_i \sum_{j=1}^{M} E|\triangle x_j^{(n)}| \ ,$$

and summing over all users we obtain

$$\sum_{i=1}^{M} E|\triangle x_i^{(n+1)}| \leq (\sum_{i=1}^{M} p_i q_i) \cdot \sum_{i=1}^{M} E|\triangle x_i^{(n)}| \tag{4.9}$$

If $\sum_{i=1}^{M} p_i q_i < 1$, $\mu^{(n)} := \sum_{i=1}^{M} E|\triangle x_i^{(n)}|$ is a decreasing positive sequence, and hence converges to the only equilibrium state of (4.9), zero. This implies convergence of each individual term in the summation to zero, which in turn says that $x_i^{(n)} \to x_i^* \ , i = 1, \ldots, M$, with probability 1. Notice that the sufficient condition (4.5) for the stability of PUA also guarantees the stability of RUA for the linear utility case.

The question now comes up as to the choice of p_i that would lead to fastest convergence in (4.9), which we will call the optimal update probability. Maheswaran and Başar [11] show that in a quadratic system without delay, one can find a fairly tight bound for optimal update probability as number of users goes to infinity, and this bound is $\frac{2}{3}$. Repeating the same analysis for this model and linear cost function leads to an exact update probability $p_{opt} = \frac{2}{3}$, which is optimal for a large number of users.

The stability and convergence results obtained also apply to the local analysis of the nonlinear utility function as in PUA. Hence the nonlinear utility case is locally stable under RUA. Moreover, Lemma 4.1 and Lemma 4.2 are valid for RUA, and hence Theorem 4.3 holds, indicating the global stability of the algorithm.

4.3 Gradient Update Algorithm (GUA)

Gradient update algorithm can be described as a relaxation of PUA. For this scheme, we define a relaxation parameter $s_i \ , 0 < s_i < 1$, for i^{th} user, which determines the stepsize the user takes towards the equilibrium solution at each iteration.

For the linear utility case, the algorithm is defined as:

$$x_i^{(n+1)} = x_i^{(n)} + s_i \cdot [(mq_i - x_{-i}^{(n)} q_i) - x_i^{(n)}] \ \forall i, n \tag{4.10}$$

Different from both PUA and RUA, the users are not myopic in this scheme. Although they seem to choose suboptimal flow rates at each iteration instead of exact optimal solutions, they benefit from this strategy by reaching the equilibrium faster. GUA, despite its deterministic nature like PUA, is actually very similar to RUA in analysis. When compared with PUA, as we observe in simulations, GUA converges faster to Nash equilibrium than PUA in highly loaded delay-free systems, where there is a high demand for scarce resources and users act simultanously. An intuitive explanation can be made using the fact that equilibrium point is quite dynamic in loaded systems during iterations. In PUA, users update their flows as if it is static, while in GUA, users behave more cautiously and do not rush to the temporary equilibrium point at each iteration. Thus, the wide fluctuations in flow rates, which can be observed in PUA, are avoided in this case. One can interpret the relaxation parameter s_i also as a measure of this caution. Another advantage of GUA, its relative insensitivity to delays in the system, can also be explained with the same reasoning.

A similar but deterministic version of the convergence analysis of RUA for the linear utility function yields the same convergence result as in RUA, except that p_i is now replaced with s_i:

$$\triangle x_i^{(n+1)} = (1 - s_i)\triangle x_i^{(n)} - s_i q_i \sum_{j \neq i}^{M} \triangle x_j^{(n)} \tag{4.11}$$

$$\Rightarrow |\triangle x_i^{(n+1)}| \leq (1 - s_i)|\triangle x_i^{(n)}| + s_i q_i \sum_{j \neq i}^{M} |\triangle x_j^{(n)}| \ , \forall i \tag{4.12}$$

Choosing $s_i \geq \frac{1}{1+q_i}$, and imposing the condition $\sum_{i=1}^{M} s_i q_i < 1$, the flow rates of the users converge to the unique equilibrium as in other schemes. Using (4.10), we obtain:

$$\lim_{n \to \infty} x_i^{(n)} = x_i^* = mq_i - x_{-i}^* q_i \ \ , \forall i$$

The sufficient condition (4.5) also guarantees the stability of GUA for the linear utility case. Moreover, since the GUA is a modification of PUA, it can be shown that Lemma 4.1 and Lemma 4.2 hold for the GUA as well. Thus, the stability results of the RUA are directly applicable to GUA for both the linear and nonlinear reaction functions.

Next, we investigate the possibility of finding an optimal relaxation parameter, s, for the linear utility case, in the sense that it leads to fastest convergence to the equilibrium. In order to simplify the analysis, we assume symmetric users, resulting in $q_i = q = \frac{1}{M}$, and $s_i = s \ , \forall i$. For the special case of symmetric initial conditions, we obtain from (4.11):

$$\triangle x_i^{(n+1)} = [1 - s(1 + (M-1)q)]\triangle x_i^{(n)}$$

The value of s, leading to fastest convergence in this case is

$$s_{opt} = \frac{1}{1 + (M-1)/M} \ \Rightarrow \lim_{M \to \infty} s_{opt} = 0.5 \ , \tag{4.13}$$

which leads to one-step convergence.

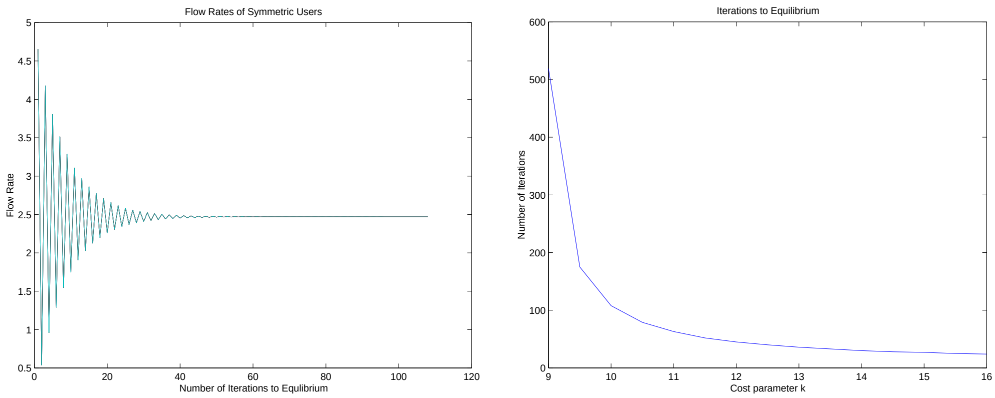
For the general case, however, it is not possible to find a unique optimal value of s, as different starting points for users which result in different $\triangle x_i$ at each iteration, affect the optimal value of s. Using simulations, we conclude that optimal value of s for a delay-free linear system should be in the range $0.5 < s_{opt} < 1$.

The analysis for the linear utility case applies to the nonlinear utility case locally, giving the same local stability and convergence results. One can show that in addition to the local results, global convergence and stability of PUA also apply to GUA. Therefore, GUA converges globally to the unique equilibrium in the nonlinear utility case. As it will be shown in numerical examples, GUA becomes advantageous only under heavy load, and loses its fast convergence property in lightly loaded systems.

5 Numerical Simulations of the Update Schemes

Each update scheme analyzed in the previous section is simulated using MATLAB. The proposed model is tested through extensive simulations for both nonlinear and linear reaction functions. The latter can be considered as either worst-case analysis or local approximation to the nonlinear utility cost. The system is simulated first without delay under all three update schemes: PUA, RUA and GUA. Next, in the second group of simulations, uniformly distributed delays are added to the system for a more realistic analysis. The convergence rate is measured as the number of iterations required to reach the unique Nash equilibrium. As a simplification, we assumed symmetric users in most cases, where cost parameters like, a, k, q, update probability, p, for RUA, and relaxation parameter, s, for GUA are not user specific. Starting condition for simulations is the origin, i.e. zero initial flow, unless otherwise stated. The following criterion is used as the stopping criterion, where M is the total number of users. $\sum_{i=1}^{M} |x_i^{(n+1)} - x_i^{(n)}| \leq M \cdot \epsilon$. The stopping distance is chosen sufficiently small, $\epsilon = 10^{-5}$, for accuracy in all simulations.

5.1 Simulations for Delay-free Case



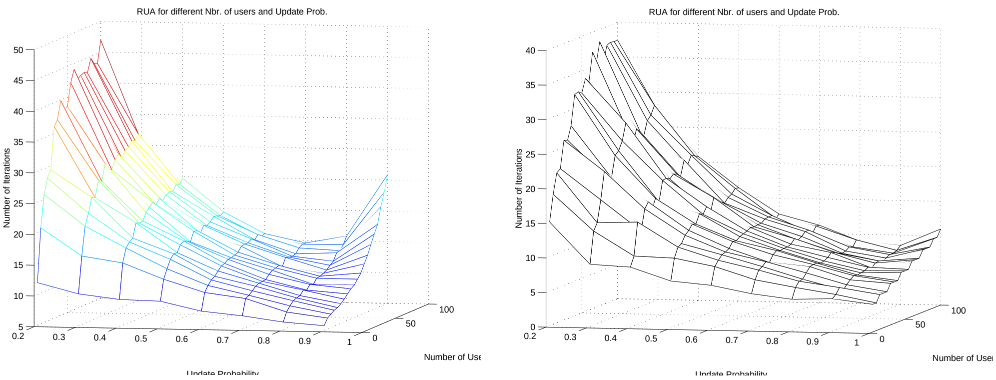
(a) (b)

Figure 2: (a) Flow rates vs. iterations to equilibrium in case of symmetric users and PUA, (b) Convergence rate of PUA for different values of k.

The convergence of the update algorithms for different numbers of users, as a crucial parameter, is investigated throughout the analysis. However, we first implemented the basic PUA algorithm with $M = 20$ users with linear reaction functions and $a = 1$ indicating a high demand for bandwidth. $k = 10$ is chosen to ensure stability. From Figure 2(a), we observe the undesirable, wide oscillations in flow rates of users, which is a disadvantage of PUA under a heavily loaded delay-free system. In this case, although the number of users is small, the low value of pricing parameter k loads the system. Absence of delay in the system also contributes to the instantaneous load, as the users act simultaneously. The instantaneous demand affects the convergence rate significantly in delay-free systems, especially under PUA.

Another important parameter in the system is the price, k. The impact of the price on the system is investigated in the next simulation. Figure 2(b) shows the effect of varying the pricing parameter k under PUA. Again, there are $M = 20$ users. It can be observed that as the price increases, the convergence rate drops. An intuitive explanation for this phenomenon is based on the effect of price on the demand of users. An increase in price results in a decrease in demand and system load, leading to faster convergence. Even though the simulation here is for a delay-free linear-utility system, varying the price leads to similar results under all update schemes for both linear and nonlinear reaction functions. Theoretical calculations based on linear utility, in the previous section, show that the minimum value of k satisfying the stability criterion is 9.2 for this specific case. This bound on k is only a sufficient condition for stability, which is verified in this simulation by observing the convergence of system for $k = 9$. The large number of iterations required, on the other hand, indicates the tightness of the bound.

Next set of simulations investigate the two basic parameters of RUA: M, number of active users, and, p, the update probability. The simulation results in Figure 3(a) verify the theoretical analysis of the previous section for linear utility cost. It is observed that with the increasing number of users the optimal update probability gets closer to the value 2/3. For completeness, the same simulation is repeated for the logarithmic utility cost. Interestingly, we obtained similar results, as shown in Figure 3(b). Due to the structure of logarithmic utility function, the demand of users is less than in the linear utility case, and hence the system is not loaded as much as in the linear case. For the same number of users, we observe that the optimal update probability shifts to higher values. As a conclusion, Figure 3(b) can be considered as a stretched version of Figure 3(a), due to the change in load.



(a) (b)

Figure 3: (a) Convergence rate of RUA as M gets larger, for different update probabilities $0 < p < 1$, and linear utility, (b) Convergence rate of RUA for nonlinear utility.

Similar to RUA, a simulation based on the relaxation parameter s is done for linear cost under GUA. The result confirms the theoretical result (4.13) for symmetric initial conditions. Other initial conditions, however, lead to different optimal values for s, in most cases between 0.6 and 0.8. The result can be interpreted as the variation in the amount of instantaneous demand for bandwidth. In the symmetric initial condition, all users act the same way, leading to higher simultaneous demand, where 'being cautious' or decreasing s is advantageous. For other initial values, the instantaneous demand decreases, where increasing s affects the convergence rate positively. We conclude that GUA is only advantageous in situations with high instantaneous and total demand, which will further be verified in delayed simulations.

Finally, we conclude the simulations without delay by a comparison of the convergence rate of all three algorithms for different numbers of users. The results for the linear reaction function are depicted in Figure 4. We observe clearly that both GUA and RUA are superior to PUA. Another important and promising observation is that the rates of convergence for GUA and PUA are almost independent of the number of users. The simulation is repeated for nonlinear utility cost and for highly and lightly loaded systems. To change the amount of load on the system, the capacity parameter m is varied this time, instead of price k. They affect, as expected, the convergence rate in opposite ways. Obviously, the smaller the capacity, the heavier the load. Similar to the linear utility case, GUA converges faster with increasing number of users. It performs, however, poorer under light load. Same trend is also observed for RUA. One interesting phenomenon is the high performance of PUA under light load. It can be interpreted as a result of the low instantaneous demand due to the variation of utility for different flow rates.

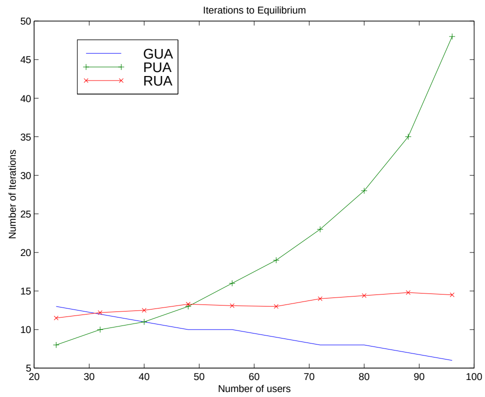
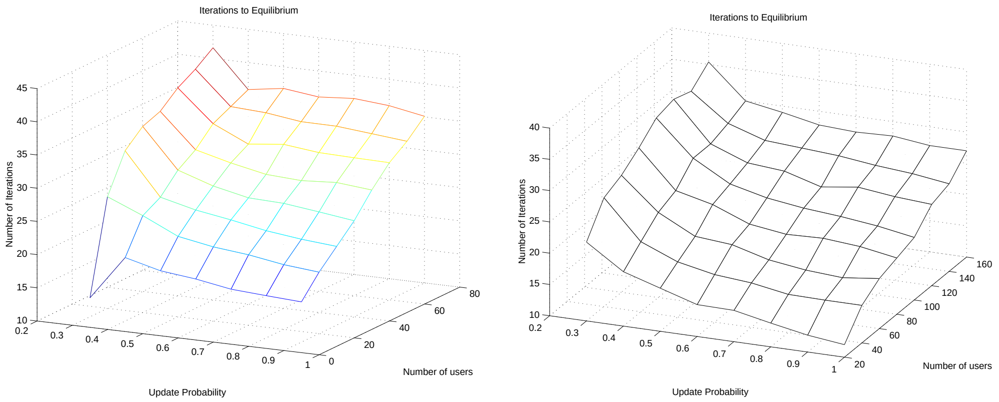


Figure 4: Comparison of Convergence rates of PUA, RUA and GUA for increasing number of users, linear utility, delay-free system.

5.2 Simulations with Delay

In order to make the simulations more realistic, we next introduce the delay factor into the system in following way: users are divided into d equal groups, where each group has an increasing number of units of delay. For example in a four group system, the first group has no delay, the second one has one unit delay, the third group two units of delay, etc. When the simulations are repeated with uniformly distributed delay as described, the results obtained are quite different from the previous ones. PUA, for example, performs better than in the linear utility case without delay. This is possibly caused by the decrease of instantaneous demand, due to the delay factor. This result strengthens the argument made on PUA in the previous section.

In RUA, however, the optimal update probability disappears in contrast to the delay-free case, as can be seen in Figure 5. Again, the underlying cause is the effect of delay factor on instantaneous demand. Another important result is the similarity of the results for linear (a) and nonlinear (b) cases in this simulation. Regarding PUA, we can conclude that it performs better when both instantaneous and total demand are low and system resources are abundant. Such conditions exist for delayed systems with users having logarithmic utility. RUA, on the other hand, performs worse



(a) (b)

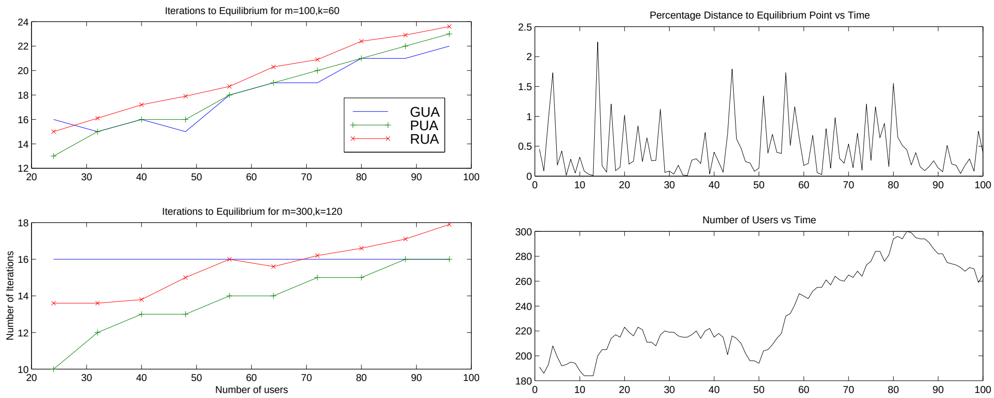
Figure 5: (a) Convergence rate of RUA as M gets larger and for different update probabilities $0 < p < 1$ in a delayed system with $d = 5$, (b) same simulation with the nonlinear system.

with decreasing update probability in such a case.

Next, GUA is investigated under a delay incorporated system for an optimal relaxation parameter. Using the results of several simulations, we conclude that the optimal value of s decreases in the linear utility case, as the delay factor increases. Interestingly, this trend disappears for the nonlinear case. Similar to RUA, GUA also loses its advantage with nonlinear reaction functions in a delayed system under normal load. Comparison of all three algorithms in the delayed nonlinear system for high and low load can be seen in Figure 6(a). In the graph below, prices are halved, while the capacity is tripled with respect to the one above. As expected, PUA performs better than RUA for any load. Under light load, PUA is superior to GUA, with the aid of delay factor and low instantaneous delay due to logarithmic utility of users. As the load in the system increases, GUA performs comparable to PUA, verifying the observation in Figure 6(a).

In another set of simulations, we investigate the robustness of the algorithms under disturbances. Disturbance is added to the system by varying the number of users at each iteration by about %10 of the total number of users on the average. The arrival of users is modeled as a Poisson random process, and connection durations are chosen to be exponentially distributed. Hence, the number of users in the system constitute a Markov chain. Figure 6(b) shows the stability results under different update schemes in terms of the percentage distance to the ideal equilibrium for an example time window. The lower right graph is the result of the simulation with nonlinear reaction function under PUA. We observe that the average distances to the equilibrium vary between %0.5 and %1.5, which indicate that the system is very robust under all schemes and costs.

Finally, a pricing scheme with two classes, a group of priority and a group of regular users is studied. Priority users are charged less -in terms of network credits- than the regular users by setting $k = 200$ vs. $k = 240$ for the excess flow rate, x. A similar disturbance structure to the one above is used by varying the number of users in order to create a more realistic setting. Again the simulation is done with users having a nonlinear reaction function under PUA. The flow rates of two sample users from each class are shown in Figure 7. We observe that the pricing scheme is successful in differentiating the priority user from the regular one. Moreover the robustness of the model is



(a) (b)

Figure 6: (a) Comparison of Convergence rates of PUA, RUA and GUA for increasing number of users, nonlinear cost. (b) Robustness Analysis for PUA in case of the nonlinear utility. Percentage distance and number of users vs. time.

preserved.

6 Conclusion

We have introduced a mathematical model which can be used as a basis for implementation of real time traffic on the Internet. The combination of admission control and end-to-end distributed flow control results in a very flexible framework, which captures all traffic types from low to medium elasticity. Market based approach enables the model to address two major issues, pricing and resource allocation, simultaneously.

A unique Nash equilibrium is obtained, and convergence properties of relevant asynchronous update schemes are investigated both theoretically and numerically. Conditions for the stability of the equilibrium point under three update algorithms are obtained and analyzed in the cases of linear and nonlinear reaction functions. The simulation results suggest the use of GUA or RUA in heavy loaded systems with less delay and high demand for bandwidth. In delayed systems, however, PUA performed better than the other two. The linear analysis not only provided a local approximation to the nonlinear cost, but also established convergence and stability results, helping to solve the general nonlinear cases.

One of the advantages of this model is its flexibility, which at the same time opens up many directions for future research. There are also many open questions requiring further investigation, such as the following: (i) The model is analyzed here for a sample bottleneck node. Possible implementations for a general network topology and routing problem are open issues. (ii) The effect of varying the virtual capacity C and the initial admission scheme are possible points of investigation. In terms of pricing, the relation between fixed and variable prices should be investigated. (iii) Although the model is designed to share network resources with a best effort type distributed, elastic network like the Internet, we have not addressed possible issues regarding the interaction of different

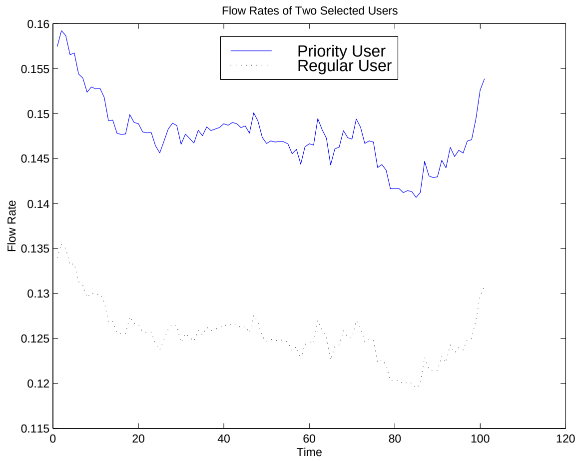


Figure 7: Flow rates of a priority user with $k = 200$ and a regular user with $k = 240$ vs. time.

protocols on the same network. Such an interaction may be a rich source for further research. Priority queueing, for example, might be investigated, as a means of combining the protocols at the router level.

References

[1] E. Altman, T. Başar, T. Jimenez, and N. Shimkin. Competitive routing in networks with polynomial cost. In *Proceedings of the IEEE INFOCOM 2000*, March 2000.

[2] E. Altman and T. Başar. Multi-user rate-based flow control. *IEEE Transactions on Communications*, 46(7):940–949, July 1998.

[3] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking*, 1:510–521, October 1993.

[4] C. Douligeris and R. Mazumdar. A game theoretic perspective to flow control in telecommunication networks. *Journal of the Franklin Institute*, 329:383–402, March 1992.

[5] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* 2nd ed. Philadelphia, PA: SIAM, 1999.

[6] Y. A. Korilis and A. Lazar. On the existence of equilibria in noncooperative optimal flow control. *Journal of the ACM*, 42:584–613, May 1995.

[7] M. T. Hsiao and A. Lazar. Optimal decentralized flow control of markovian queueing networks with multiple controllers. *Performance Evaluation*, 13:181–204, 1991.

[8] D. Bertsekas and R. Gallager. *Data Networks.* 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1992.

[9] S. Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, September 1995.

[10] D. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Compuation: Numerical Methods.* Upper Saddle River, NJ: Prentice Hall, 1989.

[11] R. T. Maheswaran and T. Başar. Multi-user flow control as a Nash game: Performance of various algorithms. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 1090–1095, December 1998.