A Variable Rate Model with QoS Guarantees for Real Time Internet Traffic

Tansu Alpcan and Tamer Başar

Coordinated Science Laboratory, University of Illinois at Urbana-Champaign,
Urbana, Illinois 61801-2307, USA

ABSTRACT

We develop a mathematical model within a game theoretical framework for variable rate real time traffic at a bottleneck node. We address not only the flow control problem, but also pricing and allocation of a single resource among users. A distributed, end-to-end flow control is proposed by introducing a cost function, defined as the difference of pricing and utility functions. For two different utility functions, there exists a unique Nash equilibrium in the underlying game. The paper also introduces three distributed update algorithms, parallel, random and gradient update, which are globally stable under reasonable conditions. The convergence properties and robustness of each algorithm are studied through extensive simulations.

Keywords: Flow control, game theory, real time traffic, Nash equilibrium, pricing, resource allocation

1. INTRODUCTION

The flow control mechanisms of the current Internet, implemented in TCP provide distributed, end-to-end congestion control for the internet traffic.[1] TCP was specifically designed to provide reliable, best-effort type traffic over an unreliable internetwork. Evolution of the Internet over time, however, has resulted in a network which tries to meet very different needs, in contrast with the original design goals. Implementation of RTT (Real Time Traffic) on the Internet for new applications like VoIP (Voice Over IP) or video conferencing is one such example. Pricing of the network resources and charging the users in proportion with their usage is another challenge. Internet is no longer a small, special community as it used to be, and there is an emerging need for additional mechanisms to ensure fair allocation of network resources among the users. Achieving these goals is only possible with new congestion and flow control mechanisms. The implementation of RTT requires certain QoS guarantees in a best-effort type network, to ensure the necessary minimum flow rate for possible applications.

There is ongoing effort for improving and modifying the flow control mechanisms of TCP to satisfy the needs of the current Internet. Most of the literature in this area is concentrated on controlling the best-effort type traffic, where several different approaches have been used.[2,3] These methods can be divided into three groups. The first approach is a centralized one, where the flow of each user is regulated separately by the network. Such a centralized approach is not in line with the distributed philosophy of the Internet, which is more widely acknowledged. A second approach is to provide incentives for end users to support the continued use of end-to-end congestion control, similar to the current mechanism. This can be achieved via policing methods, where those users who do not adapt their flows in the case of a congestion are categorized as *unresponsive* and punished by the network.[2] The third approach is one of the most popular approaches, because of the fact that it addresses not only congestion control problem, but also pricing and fairness issues. The basic principle here is to provide specific pricing mechanisms to end users and to let them adjust their flow rates accordingly. The recent work of Kelly et al.[3] on shadow prices and proportional fairness is an example for this type of approach. Using a feedback mechanism based on shadow prices, one can achieve stability and optimal usage of network resources. Fairness has been defined in this work in the proportional sense, which is a relaxed form of classical min-max fairness.

Game theory provides a natural framework for developing similar pricing mechanisms to solve rate control, fairness and even routing problems. An appropriate solution concept here is the noncooperative Nash equilibrium. 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. Internet, with its users of completely antagonistic nature in terms of their demand for bandwidth, justifies the use of a noncooperative game theoretic

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

framework. A second major advantage of the game theoretical approach comes from the fact that it provides a distributed solution in nature, i.e. it does not propose a central control for the network, and this is in accordance with today's Internet, as well as future trends of decentralized computing. As an example for addressing the flow control problem in a game theoretic framework, we can cite Altman and Başar,[4] 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.

All the approaches mentioned above assume highly elastic, best effort type traffic. Many applications creating such traffic, like FTP, e-mail, etc. are immune to delays and even to interruptions in communication. Implementation of RTT, however, requires QoS guarantees like a minimum flow rate and a maximum delay as well as traffic of low elasticity. In this type of traffic, users cannot decrease their flow rates below a certain threshold without seriously disturbing the communication of the higher layer application. In case of a VoIP application, for example, decreasing the flow rate below certain limits causes undesirable interruptions in the telephone call.

To accommodate these types of requirements, we propose in this paper a distributed flow control mechanism that makes real time internet traffic of low elasticity possible. While giving the users negotiable QoS guarantees necessary for current applications, a network problem cast in a noncooperative game theoretic framework provides excess flow of elastic nature, in addition to the guaranteed minimum flow. Hence, flexibility of the system is improved, by taking into account any possible future applications. The admission control mechanism we consider limits the number of users according to the available bandwidth. For traffic types requiring certain QoS levels, implementation of an admission control scheme is a natural solution.[1] The cost function we adopt consists of pricing and utility functions, and also includes utility parameters, describing the user's demand for excess bandwidth, and network adjustable pricing parameters to ensure stability of the equilibrium.

The proposed control scheme has two distinct parts. The first part consists of users and edge routers, and provides the interface between users and the rest of the network. Edge routers, which we will also refer to as 'network', are responsible for pricing the users by adjusting the parameters in cost functions, policing of malicious users, and security issues. The admission mechanism and QoS guarantees are also within the functions of the network. The second part of the mechanism is a distributed end-to-end control system, where users adjust their excess flows according to their needs but also by taking into account the state of the network. The necessary incentive is created for users by the pricing mechanism based on the adopted cost functions. An inherent feedback mechanism in the cost functions ensures that the users get the basic information about the state of the network. The noncooperative game theoretical framework provides equilibrium conditions for the system and most importantly, the market structure, where supply and demand for bandwidth determines the allocation of network resources and prices. Fairness in the network is established in such a way that the users who are willing to pay more for resources than others at any instant receive a larger proportion of the resources. A basic assumption we make is the rationality of the users, which can be justified by the fact that the term 'user' generally stands for a computer program which determines the flow rate in real time based on given parameters. Although some parameters might be adjusted directly by an individual person, it is very natural to assume that the program itself is inherently rational. This assumption removes many complications encountered in game theory, such as probability of irrational behavior by individual players.

There exist already several well established and extensively studied communication systems with admission control and fixed flow rates with PSTN (Public Switched Telephone Network) being the classical example. For this reason, we will focus more on the second part of the mechanism: distributed end-to-end control of excess flow, which is elastic, in a game theoretical framework, giving the system a significant amount of flexibility. One of the major advantages of our mechanism lies in the fact that it handles all types of traffic from zero to medium elasticity, depending on the needs of the user. It can be considered as a combination of CBR and ABR services in the ATM terminology.

In the next section, we provide a description of the proposed model, and in section 3 we introduce a unique Nash equilibrium. In section 4, we investigate various update algorithms and establish stability conditions. Simulation results are presented in section 5. Due to space limitations, we have left out the proofs of most of the results, which can however be found in the longer version of the paper, available from the authors.

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 of the total flow and the minimum flow:

$x_i = \lambda_i - \lambda_{i,min}$. The guaranteed flow rate, $\lambda_{i,min}$, 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 preserving the network resources at the same time, bounded by the maximum available bandwidth C at the bottleneck node, can be solved with the aid of an admission control mechanism.

The proposed admission control scheme is market based and has a cost and pricing structure. Although it can be compared with classical call blocking schemes, modeled often as $M/M/s/s$ queues in circuit switching,[1] it differs in many respects and has several advantages. In this scheme, a new j^{th} user, $j = N+1$, requesting a minimum flow rate, $\lambda_{j,min}$, determines itself whether to initiate a session or not, under the admission pricing function:

$$P_j^0 = \frac{k_{adm}}{C - (\lambda_{j,min} + \sum_{i=1}^{N} \lambda_i)} \tag{2.1}$$

The constant k_{adm} is determined by the network for pricing purposes. The denominator term $C - (\lambda_{j,min} + \sum_{i=1}^{N} \lambda_i)$ is responsible for setting the price of the resource, in this case the bandwidth, directly proportional to the total demand. The exchange is given the right of denying the user the requested service if the price is higher than a certain maximum threshold value. The mechanism results in a 'soft' call blocking scheme, where the decision of whether or not to block a call is taken not only by the network side, but also by the user, depending on the demand of the particular user for the bandwidth at that instant. At the same time, the network resources are prevented from going down to dangerously low levels in case of a congestion.

The guaranteed flow rate, $\lambda_{i,min}$, despite its necessity for real time applications, is inherently inflexible. Once the user negotiates with the network at the beginning of the connection and is admitted to the system, the terms cannot be changed during the connection. It is conceivable, however, that there might exist applications which would demand additional bandwidth during the course of the connection. In order to add this flexibility to the system, we consider here a network game, in which the i^{th} user can regulate his excess flow, x_i. We note that the excess flow rate is elastic in nature, i.e. it has no QoS guarantees and is bounded above by the total available excess bandwidth, m. This is the remaining available bandwidth after all guaranteed minimum flows are subtracted from the total capacity: $m = C - \sum_{i=1}^{N} \lambda_{i,min}$.

The proposed admission scheme for minimum flow rate, in spite of its differences, can be implemented similarly to classical admission schemes. Therefore, we will focus here on the excess elastic flow part of the model. The network game is defined at the bottleneck node, using a specific cost function and a totally distributed control scheme, where end users adjust their flow rates themselves. Consistent with the assumption of rationality, users minimize their costs, determined by their cost function. Overall control of the network is achieved through setting the pricing parameter and adjusting the maximum available capacity, C, which does not have to be the real physical capacity. Another function of the network is that it detects and limits unresponsive flows in the case of users with malicious intentions.[2] Each user may enter the network game for excess bandwidth and minimizes its cost by regulating its excess flow rate, x_i, after its constant flow rate, $\lambda_{i,min}$, is determined. A natural minimum for the case where the user has no demand for excess bandwidth is $x_i = 0$ or $\lambda_i = \lambda_{i,min}$. The intuitive explanation for this is that the user is completely satisfied with the pre-negotiated constant flow $\lambda_{i,min}$. Then, such a user does not need to enter the network game at all. Excluding such users from the network game simplifies the analysis. As a result, the number of users in the network game, M, can be less than the total number of users, N, or $M \leq N$. The remaining available bandwidth, m, is adjusted accordingly.

The cost function for the users 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, pricing function, is defined as follows:

$$P_i(\lambda_i) = \frac{k_i}{C - \lambda}(\lambda_i - \lambda_{i,min})^2 + l_i \lambda_{i,min} \tag{2.2}$$

Here, λ is the total flow of all users: $\lambda = \lambda_i + \lambda_{-i}$, where λ_{-i} is the sum of the flows of all users except the i^{th} one, and $k_i \geq 0$, $l_i \geq 0$ are pricing parameters determined by the network. Notice that l_i can be considered as the fixed price the user pays for the guaranteed bandwidth $\lambda_{i,min}$. The price value of this parameter is, however, not that important, since it does not affect the optimal flow rate of the user. The pricing term not only sets the actual price, but also has the regulatory function of giving the user a feedback about the network status via the denominator term $C - \lambda$. In queueing systems this term is generally interpreted as the delay. In this context, however, it has a feedback functionality. As the sum of flows of users approach the capacity C, the denominator approaches zero, and hence the price increases without bound. This preserves the network resources by forcing the users to decrease their elastic flows. Concurrently, a proportional relationship between demand and price is obtained, which ensures that the prices are set according to market forces.

The second part of the cost function, the utility function U_i, quantifies the user's utility for having the 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 time period. A reasonable assumption is to define it as strictly concave for elastic flows. As widely used and accepted in economics, a logarithmic function can be chosen as the best approximation to the utility function of the user in this case.

In order to capture the properties of real time traffic to the fullest extent, U_i, the utility of the i^{th} user is examined in two parts: The excess utility function, U_i^e, defined in terms of the excess flow rate, $x_i \geq 0$, quantifies the user's demand for excess bandwidth. On the other hand, the utility for the region $x_i < 0$ is described with either a strictly convex function which models the requirements of the real time traffic from the user's point of view or a zero function. In both cases, if the user's flow rate is less than the guaranteed flow rate, $\lambda_{i,min}$, the general utility drops fast to zero. Having more than the minimum rate on the other hand, might not increase the utility when the user has no demand for excess bandwidth. In this case, the utility function U_i is either a simple step function or strictly convex in the region $\lambda_i < \lambda_{i,min}$ and U_i^e is a constant. In accordance with the previous discussion, we exclude such users from the network game.

For the case where the user demands excess bandwidth, the utility function U_i is formulated in the general form, with the aid of excess utility, U_i^e:

$$U_i(\lambda_i) = \begin{cases} U_i^e(x_i), & \lambda_i \geq \lambda_{i,min} \\ f(\lambda_i), & \lambda_i < \lambda_{i,min} \end{cases} \tag{2.3}$$

where $f(\lambda_i)$ is either strictly convex and increasing or zero in the limiting case. Due to the nature of RTT, $f(\lambda_i)$ is bounded above by a constant d_i, implying the utility obtained from the minimum flow $\lambda_{i,min}$. Additionally, we assume U_i to be continuous, unless $f(\lambda_i)$ is zero. Under the assumption of logarithmic utility, a possible realistic utility function for a user demanding excess flow can be defined in terms of x_i:

$$U_i^e(x_i) = \ln(1 + x_i) + d_i \quad , x_i \geq 0 \ \ \forall i, \tag{2.4}$$

Based on the given pricing and utility functions, the cost function is simply $P - U$. In other words, the flow rate of a user results from the interaction between price and demand in terms of excess flow:

$$J_i(x_i, x_{-i}) = \begin{cases} \dfrac{k_i x_i^2}{m - (x_i + x_{-i})} - \ln(1 + x_i) + e_i, & x_i \geq 0 \\ \dfrac{k_i x_i^2}{m - (x_i + x_{-i})} + l_i \lambda_{i,min} - f(x_i + \lambda_{i,min}), & x_i < 0 \end{cases}, \tag{2.5}$$

where $e_i \equiv l_i \lambda_{i,min} - d_i$, is a constant and has no effect in the optimization process. The same cost function for the i^{th} user can also be expressed in terms of the total flow of the user:

$$J_i(\lambda_i, \lambda_{-i}) = \begin{cases} \dfrac{k_i}{C - \lambda}(\lambda_i - \lambda_{i,min})^2 + l_i \lambda_{i,min} - \ln(1 + \lambda_i - \lambda_{i,min}) - d_i, & \lambda_i \geq \lambda_{i,min} \\ \dfrac{k_i}{C - \lambda}(\lambda_i - \lambda_{i,min})^2 + l_i \lambda_{i,min} - f(\lambda_i), & \lambda_i < \lambda_{i,min} \end{cases} \tag{2.6}$$

Notice that the excess utility function is defined only in the region where $\lambda_i > \lambda_{i,min}$ and utility for the remaining part is defined by the convex function $f(\lambda_i)$. In the next section we will revisit this point and show that a Nash equilibrium cannot occur in the region $x_i \leq 0$ for any i, even if $x_i \leq 0$ is allowed.

A drawback of the realistic utility function above is that it leads to nonlinear reaction functions for the users, which makes analytical analysis 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 next section. In order to make the analysis tractable, however, for explicit results, we will use linear utility functions for the users, which leads to a set of linear equations as reaction functions. Accordingly, we will take as the utility function of user i:

$$U_i^e(x_i) = a_i x_i + d_i \tag{2.7}$$

where a_i is a positive constant not exceeding 1. One possible interpretation for the linear utility is that it constitutes a linear approximation to the actual utility function at any point λ_i. In this case, the system is analyzed locally in the vicinity of the chosen point $\lambda_i = x_i + \lambda_{i,min}$. The parameter a_i, is the slope of the utility function at that point:

$$a_i \equiv \frac{\partial U_i^e(x_i)}{\partial x_i} \Rightarrow a_i \leq 1 \quad ,\forall i \tag{2.8}$$

Another interpretation for the linear utility would be from a worst case perspective. The constant a_i can be chosen so as to provide an upperbound for marginal utility at any given point λ :

$$a_i = \max_{x_i} \frac{\partial U_i^e}{\partial x_i} \Longleftrightarrow a_i = \max_{x_i} \frac{1}{1+x_i} \Rightarrow a_i = 1 \quad ,\forall i \tag{2.9}$$

The upperbound value is based again on the assumption that $x_i \geq 0$ for all i. The parameter d_i is the same as in equation (2.4) and since it is a constant it can be ignored in the subsequent optimization step. It will be shown later that, given the 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. Notice that the same linear utility cost structure 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 NASH EQUILIBRIUM

3.1. Uniqueness under Logarithmic Utility Function

The optimization problem of a single i^{th} user, defined as the minimization of the cost function (2.5), is solved under the following constraints: $x_i > 0$; $x_i < m - x_{-i}$ $,\forall i$. 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 implies that the aggregate sum of all flows in a node cannot exceed its total capacity.

Differentiating the cost function (2.5) of the i^{th} user with respect to x_i everywhere except at $x_i = 0$, we obtain:

$$\frac{\partial J_i(x)}{\partial x_i} = \begin{cases} \dfrac{kx_i^2 + 2k_i x_i[m-(x_i+x_{-i})]}{[m-(x_i+x_{-i})]^2} - \dfrac{1}{1+x_i}, & x_i > 0 \\ \dfrac{kx_i^2 + 2k_i x_i[m-(x_i+x_{-i})]}{[m-(x_i+x_{-i})]^2} - \dfrac{\partial f(x_i+\lambda_{i,min})}{\partial x_i}, & x_i < 0 \end{cases} \quad ,\forall i \tag{3.1}$$

Notice that $\frac{\partial f(\lambda_i)}{\partial x_i}$ is nonnegative by the definition of $f(\lambda_i)$. Hence, (3.1) attains negative values in the interval $x_i < 0$, for each i. Moreover, the cost at $x_i = 0$ is always higher than the cost in the region $x_i > 0$, since the function $f(\lambda_i)$ is bounded above by the constant $d_i > 0$. In other words, a user can decrease the cost by increasing the flow. At a Nash equilibrium, a single user cannot improve his situation by unilaterally changing his own strategy, or flow rate in this case, given the flow rates of other users. Thus, the optimal point x_i^* has to be strictly positive.

For the second constraint, it can be observed that the cost function (2.5) of the i^{th} user becomes positive unbounded as x_i^* approaches $m - x_{-i}^*$. Again, the user can decrease its cost unilaterally by decreasing his flow rate, and hence the boundary point $x_i^* = m - x_{-i}^*$ cannot be an optimal point. The conclusion, therefore, is that every Nash equilibrium has to be an inner solution.

THEOREM 3.1. *There exists a unique Nash equilibrium in the network game with logarithmic utility functions.*

3.2. Uniqueness under Linear Utility Function

Here, we show the existence of a unique Nash equilibrium for the cost function with linear utility. Furthermore, exploiting the linearity of reaction functions, we calculate the equilibrium point explicitly. The analysis in this section 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 (2.5), subject to the positivity and capacity constraints. First assuming an inner solution, we have for the i^{th} user: $\frac{\partial J_i(x)}{\partial x_i} = 0$, which can be solved for x_i, to lead to: $x_i = (m - x_{-i})[1 \pm \sqrt{\frac{k_i}{k_i + a_i}}]$. The solution with the + sign is eliminated in view of the constraint $m - x_{-i} \geq x_i$; hence the only feasible solution is the one with the - sign:

$$x_i = (m - x_{-i})[1 - \sqrt{\frac{k_i}{k_i + a_i}}] \equiv m q_i - q_i x_{-i} \ , \tag{3.2}$$

where the last expression defines q_i. To complete the derivation, we now check the boundary solutions. For the boundary point $x_i = 0$, we observe from $\frac{\partial J_i(x)}{\partial x_i} = 0$ that $\frac{\partial J^i(x)}{\partial x_i} = -a_i$, which means the user can decrease his cost by increasing x_i. Hence, this cannot be an optimal point. For the other boundary point $x_i = m - x_{-i}$, we observe that at that point the cost goes to infinity. As a result, the inner solution is the unique optimal point for the constrained optimization problem of the i^{th} user, for each fixed $x_{-i} < m$. We observe from (3.2) 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, (3.2) can be rewritten as $x_i = m q_i - q_i(\bar{x} - x_i) \Rightarrow x_i = \frac{q_i}{1-q_i} m - \frac{q_i}{1-q_i} \bar{x}$. Sum both sides from 1 to M, and let $\lambda := \sum_{i=1}^{M} \frac{q_i}{1-q_i}$. Then, $\bar{x} = \lambda m - \lambda \bar{x} \Rightarrow \bar{x} = \frac{\lambda}{1+\lambda} m$. 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 (3.2):

$$x_i^* = \frac{1}{1+\lambda} \frac{q_i}{1-q_i} m \ , \ i = 1, \ldots, M \tag{3.3}$$

Note that (3.3) 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 foregoing derivation:

THEOREM 3.2. *There exists a unique Nash equilibrium in the network game with users having linear utility functions, and it is given by (3.3).*

We conclude the section with a result that justifies the worst case analysis based on linear utility functions.

PROPOSITION 3.3. *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 (2.5) is less than the rate $x_{i,lin}^{opt}$ when the same user has the linear utility (2.7) cost function with $a_i = 1$.*

The 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 a distributed environment, each user acts independently and convergence to this point does not occur instantenously. There exist various iterative update schemes with different convergence and stability properties.[5] We consider here three asynchronous update schemes relevant to the proposed model: PUA, parallel update algorithm, which is also known as Jacobi algorithm; RUA, random update algorithm, and GUA, gradient update algorithm, also known as Jacobi overrelaxation.[6] 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 the information flow within the system substantially.

4.1. Parallel Update Algorithm

In PUA, the users optimize their flow rates at each iteration, in discrete time intervals $\ldots n-1, n, n+1 \ldots$. 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, there are subsets of users, updating their flows given the delayed information on the flow.

For the nonlinear cost function (2.5), the players use either nonlinear programming methods 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 3^{rd} order equation: $k_i x_i^3 + (2k_i(m-x_{-i})+k-1)x_i^2 + (2(k+1)(m-x_{-i}))x_i + [m-x_i]^2 = 0$. 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 (3.2). As the root of this 3^{rd} order equation involves a complicated expression, we write the nonlinear reaction function of the i^{th} user only symbolically as $x_i^{(n+1)} = f(x_i^{(n)}, x_{-i}^{(n)}, k_i)$.

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 iteration starting at an arbitrary feasible point. We now study the convergence of PUA. For the linear case, the update function for the i^{th} user is (from (3.2)):

$$x_i^{(n+1)} = mq_i - q_i x_{-i}^{(n)} \quad \forall i, n \quad , \tag{4.1}$$

where q_i was defined in (3.2). Let $\triangle x_i = x_i - x_i^*$, where x_i^* is the flow rate of the i^{th} user at Nash equilibrium and $\triangle x_i^{(n)}$ is the difference between the user's flow rate at the n^{th} instant and its final equilibrium flow. Then we have

$$\triangle x_i^{(n+1)} = -q_i \triangle x_{-i}^{(n)} \quad , \forall i \tag{4.2}$$

Let $\|\Delta x\| = \max_i |\Delta x_i|$, and note that, from (4.2): $\|\Delta x^{(n+1)}\| \leq (M-1)\max_i |q_i| \, \|\Delta x^{(n)}\|$. Clearly, we have a contraction mapping in (4.2) 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: $|q_i| \leq \frac{1}{M}$, $i = 1, \ldots, M$. One trivial way of meeting this condition is to set $q_i = \frac{1}{M}$, $i = 1, \ldots, M$. From (3.2) the bound on q_i translates into the following stability constraint on the pricing parameters, $k_i \geq \frac{(M-1)^2}{2M-1} a_i$. Notice that these apply not only to the analysis in the linear-utility case, but also to the local analysis of the nonlinear-utility cost function (2.5). Thus, the system is locally stable and convergent under PUA if the condition above is satisfied. The next result says that the local stability and convergence property holds not only locally, but also globally.

THEOREM 4.1. *The system is globally convergent and stable under PUA, for both the linear and logarithmic utility cost functions, under the given bound on k_i.*

4.2. Random Update Algorithm

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 $0 < p_i < 1$. Thus, at each iteration a random set of $E\{p \cdot M\}$ 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.

For the linear-utility case with linear reaction function (3.2), the update scheme may be formulated for the i^{th} user as given by (4.1) with probability (w.p.) p_i, and $x_i^{(n+1)} = x_i^{(n)}$ w.p. $(1-p_i)$. Subtracting x_i^* from both sides:

$$\triangle x_i^{(n+1)} = \begin{cases} -q_i \triangle x_{-i}^{(n)} & , \textit{with probability } p_i \\ \triangle x_i^{(n)} & , \textit{with probability } 1-p_i \end{cases} \tag{4.3}$$

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

$$E|\triangle x_i^{(n+1)}| \leq \quad p_i q_i \sum_{j=1}^{M} E|\triangle x_i^{(n)}| + (1 - p_i(1+q_i))E|\triangle x_i^{(n)}| \tag{4.4}$$

Choosing $p_i \geq \frac{1}{1+q_i}$, this can further be bounded by $E|\triangle x_i^{(n+1)}| \leq p_i \cdot q_i \sum_{j=1}^{M} E|\triangle x_j^{(n)}|$, and summing over all users:

$$\sum_{i=1}^{M} E|\triangle x_i^{(n+1)}| \leq (\sum_{i=1}^{M} p_i \cdot q_i) \cdot \sum_{i=1}^{M} E|\triangle x_i^{(n)}| \tag{4.5}$$

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 zero. This implies convergence of each individual term in the summation to converge to zero, which in turn says that $x_i^{(n)} \rightarrow x_i^*$, $i = 1, \ldots, M$, with probability 1. Notice that the sufficient condition 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 (4.5), which we will call the optimal update probability. Maheswaran and Başar[7] show that in a quadratic system without delay, one can find a bound for optimal update probability $p_{opt} \leq \frac{2}{3}$, as number of users goes to infinity. 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 when the number of users is large. 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, Theorem 4.1 holds also for RUA, indicating the global stability of the algorithm.

4.3. Gradient Update Algorithm

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:

$$x_i^{(n+1)} = x_i^{(n)} + s_i \cdot [(mq_i - x_{-i}^{(n)} q_i) - x_i^{(n)}] \ \ \forall i, n \tag{4.6}$$

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 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. So, the wide fluctuations in flow rates are prevented, which can be observed in PUA. One can interpret the relaxation parameter s_i also as a measure of this caution. Another advantage of GUA, its relative immunity 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 replaced with s_i. 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 (4.6), we obtain: As $\mathrm{n} \rightarrow \infty \quad \mathrm{x_i^{(n+1)}} = \mathrm{x_i^{(n)}} \Rightarrow \mathrm{x_i^*} = \mathrm{mq_i} - \mathrm{x_{-i}^* q_i}$, $\forall$i. The given bound on q_i 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 Theorem 4.1 holds 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 (4.6): $\triangle x_i^{(n+1)} = [1 - s(1 + (M-1)q)]\triangle x_i^{(n)}$. The value of s, leading to fastest convergence in this case is

$$s_{opt} = \frac{1}{1 + (M-1)/M} \ \Rightarrow \lim_{M \rightarrow \infty} s_{opt} = 0.5 \ , \tag{4.7}$$

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 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

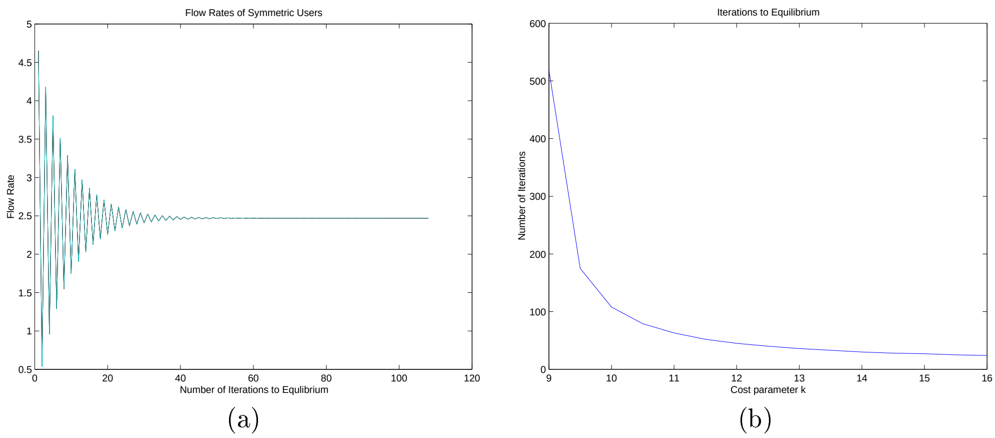
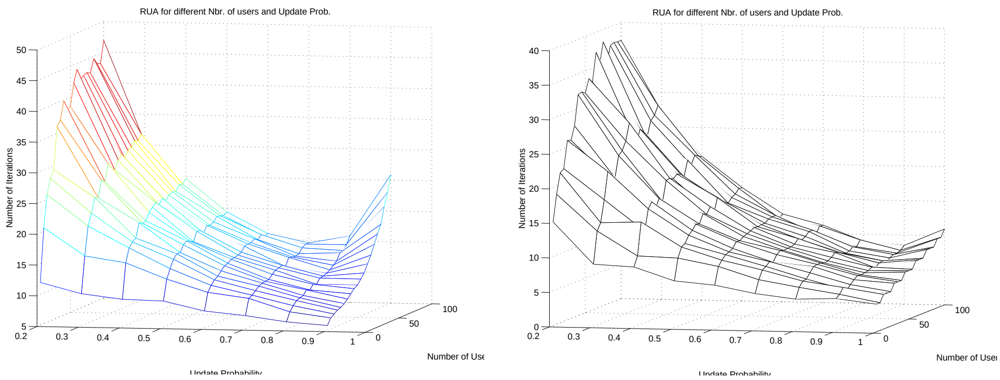


Figure 1. Flow rates vs. iterations to equilibrium in case of symmetric users and PUA (a). Convergence rate of PUA for different values of k (b).

The convergence of the update algorithms for different numbers of users, as a crucial parameter, is investigated throughout the analysis. We first implemented, however, 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 1 (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 1 (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. The bound on k_i 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 2 (a) verify the theoretical analysis in 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 2 (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, Figure2 (b) can be considered as a stretched version of Figure 2 (a), due to the change in load.



(a) (b)

Figure 2. Convergence rate of RUA as M gets larger, for different update probabilities $0 < p < 1$, and linear utility (a). Convergence rate of RUA for nonlinear utility (b).

Similar to RUA, a simulation based on the relaxation parameter s is done for linear cost under GUA. The result confirms the theoretical result (4.7) 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 displayed in Figure 3. 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 less 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.

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, first group has no delay, the second 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.

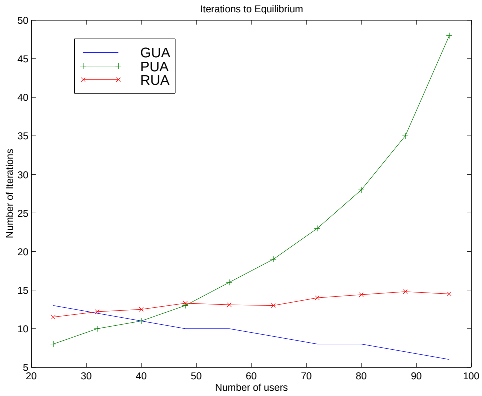


Figure 3. Comparison of Convergence rates of PUA, RUA and GUA for increasing number of users, linear utility, delay-free system.

It is possibly caused by the decrease of instantaneous demand, due to the delay factor. This result strengthens the argument about PUA in the previous section.

In RUA, however, the optimal update probability disappears in contrast to the delay-free case Again, the underlying cause is the effect of delay factor on instantaneous demand. Another important result is the similarity of the results for linear and nonlinear 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 with decreasing update probability in such a case.

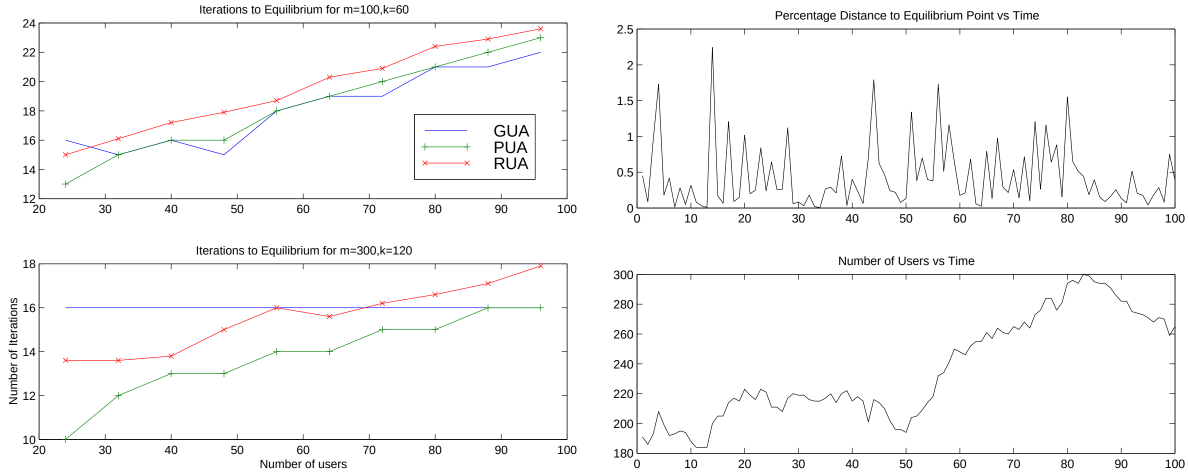
Next, GUA is investigated under a delay incorporated system for an optimal relaxation parameter. Using the results of several simulations, we conclude that the optimal value of s decreases in the linear utility case, as the delay factor increases. Interestingly, this trend disappears for the nonlinear 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 4 (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 4 (a).

In the final set of simulations, we investigate the robustness of the algorithms under disturbances. The disturbance is added to the system by varying the number of users at each iteration by about %10 of the total number of users. The arrival and departure of users are modeled as Poisson random processes, and hence the number of users in the system constitute a Markov chain. The Figure 4 (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.

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



(a) (b)

Figure 4. (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.

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 points. (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 protocols in 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. D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 2 ed., 1992.
2. S. Floyd and K. Fall, "Promoting the use of end-to end congestion control in the internet," *IEEE/ACM Transactions on Networking* , Aug. 1999.
3. F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society* **49**, pp. 237–252, 1998.
4. E. Altman, T. Başar, T. Jimenez, and N. Shimkin, "Competitive routing in networks with polynomial cost," *Proc. IEEE INFOCOM 2000, Tel Aviv, Israel* , Mar. 2000.
5. E. Altman and T. Başar, "Multi-user rate-based flow control," *IEEE Transactions on Communications* **46(7)**, pp. 940–949, July 1998.
6. D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Compuation:Numerical Methods*, Prentice Hall, 1989.
7. R. T. Maheswaran and T. Başar, "Multi-user flow control as a Nash game: Performance of various algorithms," *Proc. 37th IEEE Conference on Decision and Control* , pp. 1090–1095, December 1998.