



HAL
open science

How many parallel TCP sessions to open: a pricing perspective

Bruno Tuffin, Patrick Maillé

► **To cite this version:**

Bruno Tuffin, Patrick Maillé. How many parallel TCP sessions to open: a pricing perspective. 5th international workshop on Internet Charging and QoS Technology, Jun 2006, Saint Malo, France. hal-02367492

HAL Id: hal-02367492

<https://hal.science/hal-02367492>

Submitted on 18 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

How Many Parallel TCP Sessions to Open: A Pricing Perspective

Bruno Tuffin¹ and Patrick Maille²

¹ IRISA-INRIA, Campus universitaire de Beaulieu
35042 Rennes Cedex, France
btuffin@irisa.fr

http://www.irisa.fr/armor/lesmembres/Tuffin/Tuffin_en.htm

² GET/ENST Bretagne, 2 rue de la Châtaigneraie CS 17607
35576 Cesson Sévigné Cedex, France
patrick.maille@enst-bretagne.fr

Abstract. TCP is one of the main transmission protocols used in the Internet. It has also been recently observed that opening parallel TCP sessions might be of interest for a user in order to increase his overall average throughput. We suggest in this paper to charge users per TCP session, and we investigate the resulting game in a homogeneous context: how many sessions should each user open? Given the discrete (and even finite) space of strategies, we propose to implement a probabilistic adaptation algorithm, analyze its theoretical properties and provide numerical illustrations.

1 Introduction

Nowadays, the Internet has become a common tool in daily life. Several protocols may be used to transit data, one of the most prominent being the Transmission Control Protocol (TCP), in its early version [1] or in one of its numerous more efficient versions (slow start, Reno, Vegas...). Basically, a TCP session can be modelled by an Additive-Increase Multiplicative-Decrease (AIMD) process [2], where the rate at which packets are sent increases linearly in time, but suffers a multiplicative decrease as soon as a loss is detected. Then the rate increases linearly again until the next loss, and so on.

Recently, there has been a surge of interest in opening several TCP sessions in order to increase a user's overall throughput for bulk data transfers and by then decreasing transfer time. This concept is used by applications such as GridFTP (dev.globus.org/wiki/GridFTP), or the MultTCP proposed by Oechlin and Crowcroft [3]. The question is thus, how many TCP sockets to open simultaneously? Increasing this number increases the overall throughput, but the gain can be topped by some "technological cost" or, as we will introduce, some financial cost. This induces a game between selfish users, where each user looks at the optimal number of sessions he should open, this number of sessions influencing the overall throughput of other users. The natural framework of analysis is thus the one of non-cooperative game theory (see for instance [4] for an introduction).

In this paper, we assume that each user has to pay a fixed price per open session. Of course, a charge based on the connection duration has to be considered too, but since we consider here the system in steady-state, it can be discarded in the present analysis, and is therefore out of the scope of this paper. Our goal is to analyse the game depending of the pre-specified price (and cost). We want to study the convergence to a so-called Nash equilibrium, that is a point where no user has an interest in unilaterally changing his strategy for the number of open parallel sessions. Though, the number of choices (the number of sessions to open) is discrete, which makes the analysis a little more difficult than in a continuous context. We consider here the use of a discrete learning algorithm to solve the problem in a distributed manner [5, 6]. This algorithm adjusts the number of sessions over time using some feedback on the user's average throughput. It presents the advantages of (i) operating in a probability space to search the best number of sessions, so that it is not stuck into a local optimum (ii) being able to discover mixed strategies (iii) handling deterministic and stochastic situations (iv) being computationally simple and efficient.

In a second step, we discuss the prices that the network manager (the Internet Service Provider (ISP)) should settle in order to maximize social welfare. The corresponding *coordination ratio*, representing the loss of social welfare due to non-cooperation with respect to a centralized optimum, is also considered.

Related Work. Non-cooperative game theory has received a lot of attention in the Internet community within the last decade. It has for instance been used to model the selfish behavior of TCP users (see, among others, [7, 8]), each player of the game representing generally a single TCP session, playing with the AIMD parameters. Notice that, *in parallel to our work*, other authors have also just paid attention the game on the number of TCP connections [9]. Though, their work uses another throughput formula (applied mainly with symmetric users too), with strong assumptions on the goodput at the bottleneck that we do not have to impose here. Furthermore, it does not have a pricing perspective and is mainly devoted to a continuous game and thus does not have to use a learning algorithm as we do.

Pricing has also been recently regarded as a natural way to control congestion in the Internet and to incentivize users to fairly use the resource [10, 11], but few of them have been especially devoted to the relation with TCP (see for example [12]). Again, none of them were dealing with parallel TCP sessions.

Finally, the learning algorithm that we use has successfully been applied in wireless packet networks for prediction and tracking [13, 14], as well as for power control in CDMA networks [15].

Outline. The paper is organized as follows. In Section 2 we introduce the basic model leading in [16] to the key formula for the average throughput in the case of homogeneous sessions. In the same section, we then introduce the game on the number of sessions that each selfish user should open, and present the pricing scheme. Section 3 is devoted to the theoretical analysis of the game. Section 4 illustrates the convergence of the algorithm for different values of parameters.

It also aims at finding out optimal prices when maximizing network revenue or social welfare. Finally we conclude and present some perspectives of research in Section 5.

2 Model

2.1 TCP Model

The basic model comes from [2, 16, 17]. It represents elastic users competing for bandwidth at a link of capacity C , and controlling their send rates via a additive-increase, multiplicative-decrease (AIMD) process. Those processes are often used to model the behavior of TCP sessions. We consider here a homogeneous population of AIMD users, meaning that all users have identical additive increase parameter η and multiplicative decrease parameter β . When dealing with TCP sessions, η is proportional to the inverse of the square of the round-trip time (RTT) [17], so that we assume here that they all have the same RTT .

Assume that there are N such sessions in competition. Let $X_i(t)$ be the sending rate of session i ($1 \leq i \leq N$) at time t . All sessions increase their sending rate according to the additive-increase parameter η until capacity C is reached. This corresponds to a congestion epoch if we assume that no buffering is used in the model. Then exactly one session is selected to immediately decrease its sending rate according to the multiplicative-decrease parameter β . Let T_n be the time of the n -th congestion epoch, and let $X_{i,n}$ be the send rate of user i just after time T_n .

The dynamics of the model is then formalized as follows. Let $Z_{i,n}$ be equal to 1 if the user i undergoes multiplicative decrease at time T_n , and $Z_{i,n} = 0$ otherwise. We have

$$X_i(t) = (1 - (1 - \beta)Z_{i,n})X_{i,n} + \eta(t - T_n), \quad T_n \leq t < T_{n+1},$$

and the send rate at congestion times obeys the recurrence, for $i = 1, 2, \dots, N$,

$$X_{i,n+1} = (1 - (1 - \beta)Z_{i,n})X_{i,n} + \eta S_n$$

with $S_n = T_{n+1} - T_n$ obtained from $\sum_{i=1}^N X_{i,n+1} = C$.

It has been shown in [16] that the average aggregated throughput in steady-state is

$$\bar{x}(N) = C \left(1 - \frac{1}{1 + N \frac{1+\beta}{1-\beta}} \right). \quad (1)$$

Remarkably, this formula is shown to be true in [16] whatever the drop policy used (the choice of the session selected to decrease its rate at each congestion epochs). Several specific policies are also investigated in more details: the proportional one, where the decreased session is chosen with a probability proportional to its sending rate at the congestion epoch; the fixed one, where each one is selected with a fixed (state-independent) probability; or the largest one, where the session with the largest sending rate at a congestion epoch is selected.

One of the conclusions of formula (1) was that opening too many sessions is not worthwhile, because of the management overhead it introduces, given that for N large, the throughput increase is very small when adding a session. The goal of our paper is to investigate this point in more details in the case of users in competition, and assuming that users. Assume that we have I people in competition for the capacity C , and that user i opens N_i sessions. Then, from (1), the total number of sessions is $N = \sum_{i=1}^I N_i$ and, in the homogeneous case, the total throughput of user i is

$$x_i = C \frac{N_i}{N} \left(1 - \frac{1}{1 + N \frac{1+\beta}{1-\beta}} \right)$$

(with our previous notations).

2.2 Pricing Scheme and Game-Theoretic Formulation

The question asked in the paper is: what is the best strategy for user i ? In other words, how many sessions should user i open? The analysis requires the framework of non-cooperative game theory assuming that he reacts selfishly, since user i 's throughput depends on the total number of sessions of other users.

We additionally assume that the network operator wishes to control this number of sessions by incorporating a charge depending on the number of open sessions in order to prevent a too large number. Users' choices are then driven by their utility functions, representing a measure of the happiness or satisfaction gained from the service $U_i = f(x_i) - d(N_i)$ where

- f is the valuation function representing the gain that user i gets from a throughput x_i . We can for instance assume that $f(x_i) = \log(1 + x_i)$ [18].
- d is the charge for opening N_i sessions. It seems here also reasonable to consider it linear in N_i , $d(N_i) = \alpha N_i$. To this charge could be added a (perceived) technological cost of operating several sessions at the same time, in terms of management to “reorder” all data. We neglect it here, but it could easily be incorporated for instance by adding a fixed value α' to α .

User i 's utility function is thus considered to be:

$$U_i(N_1, \dots, N_I) = \log \left[1 + C \frac{N_i}{N} \left(1 - \frac{1}{1 + N \frac{1+\beta}{1-\beta}} \right) \right] - \alpha N_i. \quad (2)$$

The space of strategies \mathcal{S}_i of each user i is then the number of sessions he can open. We assume here that, for some technological reason, this number is upper-bounded by N_{max} . We thus have $\mathcal{S}_i = \{0, \dots, N_{max}\} \forall 1 \leq i \leq I$. this discrete number of choices complicates the pure theoretical analysis of the game. A learning algorithm is therefore used in next subsection.

2.3 A Learning Algorithm to Approach Nash Equilibrium

To solve this problem, we propose to use a decentralized discrete stochastic learning algorithm similarly to what was done in [15] for a power allocation game

in CDMA networks. The goal of each user/player is to maximize his utility. The game is played repeatedly and the optimal strategy is learned.

Player i 's strategy is defined by a probability vector $p_i = (p_{i0}, \dots, p_{iN_{max}})$ with p_{ij} the probability that user i chooses to open j sessions.

Let

$$g_i(p_1, \dots, p_I) = \sum_{N_1, \dots, N_I} U_i(N_1, \dots, N_I) \prod_{j=1}^I p_{jN_j}$$

be the expected utility for player i given strategy probability vectors. We will say that a I -tuple of strategies (p_1^*, \dots, p_I^*) is a Nash equilibrium if $\forall 1 \leq i \leq I$ and for all probability vector p defined over $\{0, \dots, N_{max}\}$,

$$g_i(p_1^*, \dots, p_i^*, \dots, p_I^*) \geq g_i(p_1^*, \dots, p, \dots, p_I^*).$$

Similarly to [15], we assume that the following discrete learning algorithm is used by each player:

1. Set the initial probability vector $p_i(0)$ for each user i . In this paper we will (arbitrarily) choose uniform initial distributions over $\{0, \dots, N_{max}\}$.
2. At each time step k , the number $N_{i,k}$ of sessions open by user i is chosen according to probability vector $p_i(k)$.
3. User i then monitors his throughput x_i and computes his utility function $U_{i,k}$ at time step k .
4. User i updates his probability vector according to the rule

$$p_{ij}^{(k+1)} = \begin{cases} p_{ij}(k) - bu_{i,k}p_{ij}(k) & \text{if } j \neq N_{i,k} \\ p_{ij}(k) + bu_{i,k} \sum_{\ell \neq N_{i,k}} p_{i\ell}(k) & \text{otherwise.} \end{cases}$$

In words, this step consists in adjusting the probability of choosing one's strategy in the next step, considering the utility brought by the current strategy: if that utility is high then the probability of the current strategy is increased, otherwise it is lowered.

5. If the algorithm has not converged goto step 2., otherwise stop.

In the algorithm, parameter b is the step size of the updating rule, and $u_{i,k}$ is a normalized utility

$$u_{i,k} = \frac{U_{i,k} - A_{i,t}}{B_{i,t} - A_{i,t}}$$

with $A_{i,t} = \min_{k \leq t} U_i(k)$ and $B_{i,t} = \max_{i,t} U_i(k)$.

Note that no knowledge of the number of players I is required, nor any specific knowledge of other users' strategies.

2.4 Social Welfare Issues

We furthermore assume that it is computationally costly for the network to support too many simultaneous TCP sessions. This can be taken into account by introducing a (converted to monetary) *network cost* per open session, that

we denote γ . Remark that this cost is not perceived by the users, who are only sensitive to their throughput and to the price they pay. The overall *social welfare* SW when each user i chooses to open N_i sessions is therefore expressed by¹

$$SW = \sum_i f_i \left(1 + C \frac{N_i}{N} \left(1 - \frac{1}{1 + N \frac{1+\beta}{1-\beta}} \right) \right) - \gamma N, \quad (3)$$

with $N = \sum_i N_i$. All the simulations presented in this paper were run with $C = 1$, $\beta = 1/2$, $N_{max} = 5$ and $\gamma = 0.05$.

3 Game Analysis

The game studied in this paper is quite complex, and difficult to study analytically. In this section, we therefore summarize the general results and properties concerning this type of game, and state what could be expected from the use of the algorithm described in the previous section.

3.1 Game Without Pricing ($\alpha = 0$)

When no pricing is introduced, the game becomes easy to solve, since each user has a dominant strategy which consists in opening the maximum number N_{max} of TCP sessions (indeed, from (1), the larger the number of sessions user i opens, the larger his throughput is, whatever the number of other sessions). Such a Nash equilibrium may not be efficient in terms of social welfare, since the computational cost incurred to the network will be maximal. This motivates the use of pricing as a tool to incentivize users to reduce their number of open connexions.

3.2 General Game: Existence of an Equilibrium

When the number of sessions each user can open is upper-bounded by a finite value N_{max} , the game that is played between users is a finite game [19]. It is a classical result in game theory that there always exists at least a Nash equilibrium in mixed strategies for such a game. Nevertheless, for general utility functions, no results of unicity can be given.

3.3 Nash Equilibrium of the Continuous Game

When the strategy set of each user is continuous (which would mean here that a user could open a non-integer number of sessions), the concavity of utility functions can be used to prove the existence and unicity of the Nash equilibrium [20]. This is the case with the utility functions given in (2). Therefore, for any price per session α , there exists a unique Nash equilibrium. Since all users are

¹ Social welfare is defined as the sum of the utilities of all agents (users+network). Here the prices paid by users do not appear, since they are paid to the network and would appear in his utility.

identical, for that equilibrium each user would open the same number N^* of sessions, with N^* satisfying

$$N^* = \arg \max_{x \in [0, N_{max}]} \left\{ f_i \left(C \frac{x}{x + (I - 1)N^*} \left(1 - \frac{1}{1 + (x + (I - 1)N^*)^{\frac{1+\beta}{1-\beta}}} \right) \right) - \alpha x \right\}. \quad (4)$$

3.4 Expected Outcome of the Learning Algorithm

From Theorem 1 in [15], we know that the algorithm can converge only to a point that is a Nash equilibrium of the game. The convergence of the algorithm is proved in [15] when the game has a unique pure Nash equilibrium, which is not the case here, as we will see in the following. In [6] (theorem 3.3), the convergence is established under some assumptions on the utility functions that we were not able to verify.

We ran the algorithm several times for identical initial conditions, and it turned out that, though the probability vectors p_i seem to converge (and actually to converge to pure strategies), the attained distributions may differ, illustrating the fact that there may be several Nash equilibria for the game. Two examples of those distributions with 3 players are displayed in Table 1. Those distributions correspond to pure strategies (Dirac distributions on a single point), leading to different Nash equilibria.

An interesting remark (from the left-hand side of Table 1) is that two identical players may have different optimal strategies at equilibrium, depending on others' choices (due to the discrete nature of the game).

Table 1. Two different outcomes (strategy probability vectors $p_{i,\cdot}$ after 10000 rounds) of the same algorithm with the same initial conditions ($b = 0.03, \alpha = 0.1$)

| # sessions | Player 1 | Player 2 | Player 3 | # sessions | Player 1 | Player 2 | Player 3 |
|------------|----------|----------|----------|------------|----------|----------|----------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 2 | 1 | 1 | 1 |
| 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |

4 Numerical Results

This section aims at illustrating the behavior of the learning algorithm, and at highlighting the interest of pricing in the considered context.

Figure 1 shows a trajectory of the number of sessions chosen by three players in competition up to $k = 300$, with $\alpha = 0.1$ and $b = 0.1$. The curve suggests the convergence to a Dirac distribution for each player, which corresponds to what was observed in Table 1. Again, identical users may have different equilibrium values due to the discrete nature of the game.

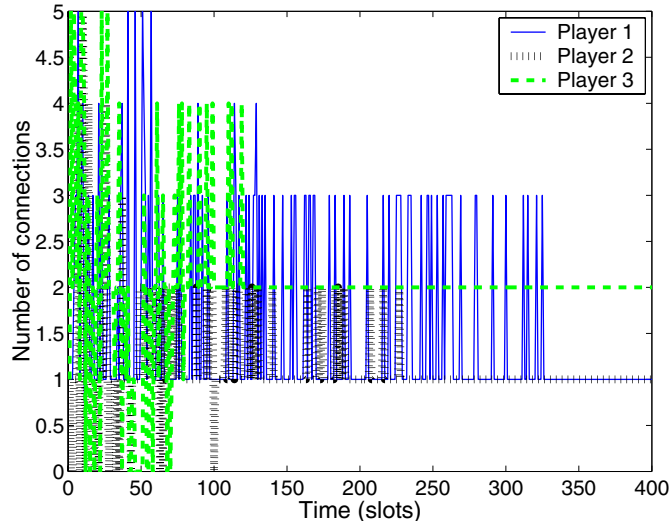


Fig. 1. Number of sessions used by three players in competition: convergence phase ($b = 0.1, \alpha = 0.1$)

Table 2 shows how the number I of players affects the equilibrium probability. We have chosen to show the mean distribution (all Dirac, but at different values) over all players. As expected, the number of open sessions tends to decrease as I increases, since the marginal throughput gain of opening an additional session is smaller, whereas the marginal cost remains fixed to $\alpha = 0.1$.

Table 3 illustrates the modifications on the aggregated strategy distributions for different values of price α .

As expected, we observe a decrease in the number of sessions when the price increases. The impact on the revenue is shown in the last line of Table 3. Due to the discontinuity of the equilibrium strategies in the per session price α (those strategies are Diracs as previously observed), the revenue is not a concave function of the per session price, and therefore it is not obvious to a network manager to discover the price that will yield the largest revenue.

The introduction of prices aims at incentivizing users to better use the network. In other terms, we expect pricing to help reduce the *coordination ratio*,

Table 2. Equilibrium probabilities for various numbers of players, with $\alpha = 0.1$

| Number of TCP sessions | $I = 2$ | $I = 4$ | $I = 6$ | $I = 8$ | $I = 10$ |
|------------------------|---------|---------|---------|---------|----------|
| 0 | 0 | 0 | 0 | 0.125 | 0.2 |
| 1 | 0 | 0.5 | 0.833 | 0.875 | 0.8 |
| 2 | 1 | 0.5 | 0.167 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 |

Table 3. Equilibrium probabilities (aggregated distribution) for various prices α

| Number of TCP sessions | $\alpha = 0$ | $\alpha = 0.05$ | $\alpha = 0.1$ | $\alpha = 0.2$ | $\alpha = 0.3$ |
|------------------------|--------------|-----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 0.333 | 0.333 |
| 1 | 0 | 0 | 0.333 | 0.333 | 0.667 |
| 2 | 0 | 0 | 0.667 | 0.333 | 0 |
| 3 | 0 | 0.333 | 0 | 0 | 0 |
| 4 | 0 | 0.667 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 | 0 |
| Corresponding revenue | 0 | 0.55 | 0.5 | 0.6 | 0.6 |

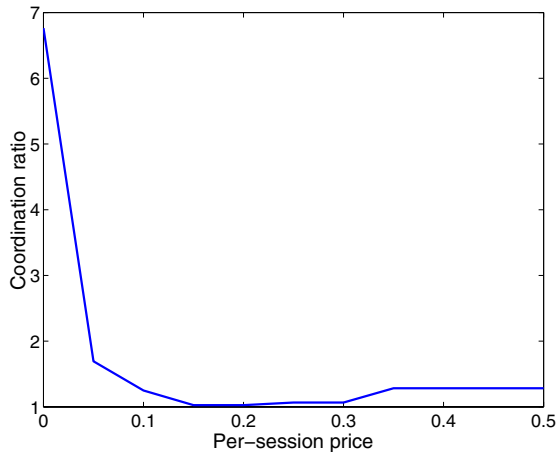


Fig. 2. Coordination ratio for different values of the charge per session (3 players, $b = 0.05$)

defined as the ratio of the maximum attainable social welfare and the actually reached social welfare (at a Nash equilibrium).

Remark: several Nash equilibria are likely to exist in the discrete game we are studying. The coordination ratio gives the loss of efficiency with respect to a centralised decision corresponding to the equilibrium attained by the algorithm. Depending on the algorithm progress, different equilibria can be reached as highlighted in Table 1, giving different values of the coordination ratio. For that reason, Koutsoupias and Papadimitriou [21] suggested to take the Nash equilibrium with the worst social welfare. The coordination ratio associated to the worst Nash equilibrium is called *price of anarchy*. In this paper, we only plot the coordination ratio with the equilibrium given by the learning algorithm, since we cannot derive all Nash equilibria.

Here, computing the maximal social welfare can be hard when considering discrete strategy sets. We therefore computed an upper bound, corresponding to the continuous strategy set case. The effects of the charging factor α on

the coordination ratio can be seen in Figure 2. As expected, the case where the per session charge is null is very bad, since it corresponds to every user opening N_{max} sessions. We also remark that for a judiciously chosen value of α , the social welfare is very close to the optimum value that could be reached if users would collaborate (the coordination ratio is very close to 1). The shape of the coordination ratio curve plotted in Figure 2 is typical of congestion games: while the introduction of prices helps lowering the demand and leads to a better utilisation of resources by reducing the negative externalities (descending phase of the curve), setting too large charges prevents users from entering the game and the resource becomes underused (ascending phase of the curve).

We can therefore conclude that introducing a very simple pricing scheme (fixed per-session price) can lead the initially inefficient Nash equilibrium to an efficient one (where efficiency is in the sense of social welfare), therefore arguing in favor of the use of pricing in such contexts.

5 Conclusions

In this paper, we have considered a game where TCP users compete for bandwidth by opening parallel sessions in order to increase their QoS. The game is controlled by imposing a fixed charge for each open session. We consider that each user implements a (decentralized) discrete learning algorithm to find out his best strategy, and convergence to a Nash equilibrium is discussed. We have also discussed the pricing strategy for the network manager.

As directions for future research, we aim at investigating the more realistic case of heterogeneous sessions, with different round-trip times, meaning different additive-increase parameters. A closed-form expression for the average throughput does not exist in full generality yet, but we could numerically look at the resulting equilibrium, as well as at pricing solutions for incentivizing users to fairly share the resource (this pricing scheme probably using the RTT values).

Acknowledgement

The authors would like to thank R. Chandramouli and Y. Xing for some discussions on the learning algorithm used in the paper.

References

1. Jacobson, V.: Congestion avoidance and control. In: ACM SIGCOMM. (1988) 314–329
2. Baccelli, F., Hong, D.: AIMD, Fairness and Fractal Scaling of TCP Traffic. In: Proceedings of IEEE INFOCOM 02. (2002)
3. Crowcroft, J., Oechslein, P.: Differentiated End-to-End Internet Services using a Weighted Proportional Fair Sharing TCP. ACM Computer Communications Review **47**(4) (2004) 275–303
4. Osborne, M., Rubenstein, A.: A Course on Game Theory. MIT Press (1994)

5. Narendra, K., Thathachar, M.: Learning Automata: An Introduction. Englewood Cliffs: Prentice Hall (1989)
6. Sastry, P., Phansalkar, V., Thathachar, M.: Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information. *IEEE Trans. Systems, Man, and Cybernetics* **24** (1994) 769–777
7. Akella, A., Seshan, S., Karp, R., Shenker, S.: Selfish behavior and stability of the internet: A game-theoretic analysis of tcp. In: Proceedings of ACM SIGCOMM 02. (2002)
8. Altman, E., Jiménez, T., Núñez Queija, R.: Analysis of two competing TCP/IP connections. *Performance Evaluation* **49**(1) (2002) 43–56
9. Zhang, H., Towsley, D., Gong, W.: TCP connection game: A study on the selfish behavior of TCP users. In: Proc. of 13th IEEE Intl. Conf. on Network Protocols (ICNP'05), Boston, Massachusetts, USA (2005)
10. Courcoubetis, C., Weber, R.: Pricing Communication Networks—Economics, Technology and Modelling. Wiley (2003)
11. Tuffin, B.: Charging the Internet without bandwidth reservation: an overview and bibliography of mathematical approaches. *Journal of Information Science and Engineering* **19**(5) (2003) 765–786
12. Altman, E., Barman, D., El Azouzi, R., Ros, D., Tuffin, B.: Pricing Differentiated Services: A Game-Theoretic Approach. *Computer Networks* (2005 (to appear))
13. Chandramouli, R.: A stochastic technique for on-line prediction and tracking of wireless packet networks. In: Proceedings of the Thirty-Fifth Asimolar Conference on Signals, Systems and Computers. (2001) 672–676
14. Kiran, S., Chandramouli, R.: An adaptive energy efficient link layer protocol using stochastic learning control. In: Proceedings of the IEEE International Conference on Communications (ICC). (2003)
15. Xing, Y., Chandramouli, R.: Stochastic learning solution for distributed discrete power control game in wireless data networks. Technical report, Stevens Institute of Technology (2004)
16. Altman, E., Barman, D., Tuffin, B., Vojnović, M.: Parallel TCP Sockets: Simple Model, Throughput and Validation. In: IEEE INFOCOM 2006, Barcelona, Spain (2006)
17. Altman, E., El Azouzi, R., Ros, D., Tuffin, B.: Loss strategies for competing TCP/IP connections. *Computer Networks* (2005 (to appear))
18. Kelly, F.: Charging and rate control for elastic traffic. *European transactions on Telecommunications* **8** (1997) 33–37
19. Fudenberg, D., Tirole, J.: Game Theory. MIT Press, Cambridge, Massachusetts (1991)
20. Rosen, J.: Existence and uniqueness of equilibrium points for concave n -person games. *Econometrica* **33**(3) (1965) 520–534
21. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: Proc of 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS 1999). Volume 1563 of *Lecture Notes in Computer Science*. (1999) 404–413