OPTIMAL RANDOM ACCESS IN NETWORKS WITH TWO-WAY TRAFFIC

Eitan Altman†, Vivek S. Borkar††, Arzad A. Kherani†
† INRIA, 2004 route des Lucioles, 06902 Sophia Antipolis, France. Eitan.Altman, alam@sophia.inria.fr
†† Tata Institute of Fundamental Research, Mumbai, 400 005, INDIA. borkar@tifr.res.in

Abstract - We consider a random access network in which the nodes need to optimize their channel access rates. The nodes are assumed to be rational and interested in their performance seen as a transmitter as well as a receiver. By casting this problem as a non-cooperative game, we derive conditions for the Nash equilibrium. We also show the existence of a Nash equilibrium when the nodes are constrained by their battery power (for this case, the constraints on the access rates of the nodes become coupled). For the special case where all nodes are each other's neighbors, we find that the equilibrium is given by the solution of a system of linear equations. An adaptive distributed scheme is then proposed for learning this equilibrium and its convergence is studied numerically.

Keywords - Game theory, Stochastic approximation algorithm.

I. Introduction

The most prominent medium access scheme in CSMA based networks is that of random access where, after sensing an idle channel, a node waits for a random amount of time before starting its transmission. This provides protection against the possibility of many nodes transmitting simultaneously when the channel becomes idle. Important recent example of a CSMA based network is a wireless ad hoc network that uses the IEEE 802.11 protocol [1] for medium access.

After it has sensed an idle channel, a node has to *optimally* attempt a transmission. By *optimal* attempt we mean that the nodes should try not to be too aggressive in attempting a transmission (thereby risking a collision) and, at the same time, nodes should not be too conservative so as to miss a chance of a successful transmission. For IEEE 802.11, for example, this is same as optimizing the mean of the backoff timer distribution. Clearly this optimal operating point (if it exists) will depend on the topology of the network (that is, the number of nodes and their relative orientation in space). In the case of ad hoc networks however the network topology is not fixed and changes over time owing to mobility or failure of the nodes. Thus there is a need for a node to *adapt* its channel access rate to the changing network topology.

There is significant amount of literature which addresses the above problem (see [2], [3] and references therein). Most of the studies of wireless ad hoc networks take the view which assumes that a constituting device (node) is either a source, a forwarder or a destination of packets and assumes that a node's perceived performance is the rate at which its *transmissions* are successful. These studies implicitly assume that maximizing the success probability for transmissions from the nodes in a network, one can get good performance for nodes acting as receivers (see, for example, [2], [3]). However, contrary to the viewpoint adopted in these models, a node in an ad hoc network is both sender as well as receiver of packets and hence will be interested in a *combined* performance measure that reflects its performance as a sender and as a receiver. These are two conflicting requirements: a node, if it tries to be too aggressive in sending packets, may lose opportunity to receive packets meant for itself and vice versa.

An important example in such a scenario is where a node is transferring data using TCP. Here a node is required to be equally available for transmission of data packets as well as the reception of acknowledgment packets in order to be able to get a better performance from the closed loop behavior of TCP. Yet another example of such a scenario is where multiple applications running on a node are sending as well as downloading files, as in this case also the traffic in both directions are equally important for the node.

Another motivation for the present work comes from looking at proposed algorithms for multicast over IEEE 802.11 networks. The existing literature (see [4] and references therein) looks at a node only as a sender and proposes strategies for the node to exploit the information about the number of its neighboring nodes which are ready to receive. However, it may be possible that using the proposed protocols a sender node may not get a chance to become a receiver frequently, thus losing on the reverse traffic. This is because the proposed protocols usually make many consecutive attempts to transmit a single packet which may reduce the chance of successful reception by other nodes and may also result in increased MAC layer collisions.

As observed in [3], the nodes forming an ad hoc network are expected to be rational, i.e., are interested in maximizing their own performance. This is different from the case of sensor networks where, as in [5], nodes try to optimize some overall network objective and thus the performance seen by an individual node does not matter.

We view the problem as a non-cooperative game with each node trying to optimize its own objective. By using a general performance metric for each node, we show the existence of a Nash equilibrium giving an optimal channel access rate for each node. The existence of a Nash equilibrium is shown also for the case when the nodes are constrained by their battery

This work was supported by research grants from *Indo–French Center for Promotion of Advanced Research (IFCPAR)* under research contract number 2900–IT as well as by the EURO NGI network of excellence.

power (for this case the constraints on the access rates of the nodes become coupled); this requires showing that the constraint defining functions are quasi-concave. For the special case where each node is neighbor of other node (like in Wireless LANs), we propose a stochastic approximation algorithm based iterative scheme to learn the Nash equilibrium. This algorithm is obtained in much a similar way as done in [6] which studied the problem of finding the channel access rates for a network where nodes are interested in success of their transmissions (not in receiving). [6] shows the existence of Nash equilibrium only in the case where nodes are not battery power constrained so that the constraint set for each node is *orthogonal*.

II. The Model and Problem Formulation

Let the network consist of N nodes. Denote the set of nodes by $\mathcal{N}$. For $i,\ j\in\mathcal{N}$ say $i\rightarrow j$ if node i can receive node j's transmission. We assume that $i\rightarrow j$ iff $j\rightarrow i$ and use the notation $i\leftrightarrow j$ to mean either of these two. Associate with each node $i\in\mathcal{N}$, a neighborhood set $\mathcal{N}(i):=\{j\in\mathcal{N}:i\leftrightarrow j\}$. Also let $\mathcal{S}(i):=\{k\in\mathcal{N}\backslash\mathcal{N}(i)\backslash\{i\}:\exists j\in\mathcal{N}(i):k\in\mathcal{N}(j)\}$ be the second hop neighbor set of node i.

We assume that the system operates in a slotted mode and nodes attempt transmissions of packets to their neighboring nodes. We also assume that in any slot, node i has a packet to transmit with a fixed probability p_i. The channel access is random, i.e., in each slot, if node i has a packet to be transmitted, it decides to attempt a transmission (broadcast) with probability ξ_i and decides to receive with probability $(1-\xi_i)$. Thus, in any slot, node i attempts a transmission with probability $p_i\xi_i$. The quantity $\alpha_i = p_i\xi_i$ is called the *attempt probability* of node i. What follows can be easily modified to account for a node i keeping an attempt probability $\xi_{i,j}$ for its neighboring node j, or to some subset of its neighbors (multicast).

Our problem now is to find, for a given network, the values of ξ_i (or $\xi_{i,j}$, as the case may be) which maximize node i's performance.Since p_i is fixed, it is enough to compute α_i in order to find ξ_i. The objective then is to design an algorithm using which a node i computes α_i or $\alpha_{i,j}=p_i\xi_{i,j}$ for itself *adaptively* in a *distributed* manner.

Let $P_s(i)$ be the conditional probability that a transmission attempt from node i is *successful* (conditioned on the event that node i transmits), and $P_r(j,i)$ denote the probability that a transmission from node j is successfully received by node i.

Here one can have various notions of node i's transmission being *successful*. We use the simple (though not restrictive) criteria: node i's transmission is successful if *all* of it's neighboring nodes correctly receive the transmission.

Each node wants to maximise its own utility function which reflects the performance obtained by the node under the sending probabilities selected by the nodes in the network. A common ingredient of the utility function of node i is a combination of the rates at which node i successfully transmits and receives packets. Thus the problem for node i is to maximise

$$U_i(\underline{\alpha}) = A_i\alpha_i P_s(i) + \sum_{j\in\mathcal{N}(i)} A_{i,j}\alpha_j P_r(j,i) - C_i\bar{\alpha}_i P_s(i) \quad (1)$$

such that $\alpha_i\geq 0$ and $\bar{\alpha}_i = 1-\alpha_i\geq 0$. Here A_i, C_i and $A_{i,j}$ are some non-negative constants. The first and second terms here are "rewards" for *success* owing to, respectively, transmission and reception. $A_{i,j}$ will be zero for node j whose transmission can not be directly received by node i. The third term is included to act as a punishment for missed opportunities, thus aiming at maximising node i's use of network. Note that the last term also is the probability of the event where none of the neighboring nodes of node i are sending to node i when node i is ready to receive, thus this term also represents the time wasted by node i in trying to receive when there is nothing to receive.

Our definition of $P_s(i)$ means that none of the first or second hop neighbors of i transmit when node i does. Thus

$$P_s(i) = \Pi_{j\in\mathcal{N}(i)}(1-\alpha_j)\Pi_{k\in\mathcal{S}(i)}(1-\alpha_k). \quad (2)$$

Similarly, it is seen that, for $j\in\mathcal{N}(i)$, $P_r(j,i)$ is,

$$P_r(j,i) = \Pi_{k\in\mathcal{N}(i)\cup\{i\}\backslash\{j\}}(1-\alpha_k). \quad (3)$$

Since the nodes are each trying to optimize their own objectives without any cooperation, this is a noncooperative game. The 'action space' for each node is the interval $[0,1]$ from which it chooses the transmission probability. This is compact convex. Also, each node's objective function (1) is separately concave continuous in each argument (in fact, linear). This is a concave N-person game, thus a Nash equilibrium exists, i.e., a choice $\underline{\alpha}^* = [\alpha_1^*,\cdots,\alpha_N^*]$ such that if all but the $i-$th node transmit with probabilities α_j^*'s, $j\neq i$, then it is optimal for $i-$th node also to use α_i^* [7]. Our aim will be to attain this Nash equilibrium. With this objective, we first seek the sufficient conditions for the Nash equilibrium.

Since for fixed α_j, $j\neq i$, this is a single agent optimization problem faced by the $i-$th node, we consider the corresponding Kuhn-Tucker condition. For any vector $\underline{\alpha}$ of attempt probabilities, let $A^*(\underline{\alpha}) = \{i:0<\alpha_i<1\}$. The (equivalent of) Kuhn-Tucker conditions for a vector $\underline{\alpha}$ to be a Nash equilibrium, i.e., a componentwise local maximum of corresponding utility functions when the other components are unperturbed, are (with $A_{i,i} := A_i + C_i$)

$$A_{i,i}P_s(i) - \sum_{j\in\mathcal{N}(i)} A_{i,j}\alpha_j\frac{P_r(j,i)}{1-\alpha_i} = 0, \qquad \forall\, i\in A^*(\underline{\alpha}) \quad (4)$$

$$A_{i,i}P_s(i) - \sum_{j\in\mathcal{N}(i)} A_{i,j}\alpha_j\frac{P_r(j,i)}{1-\alpha_i} \geq 0, \qquad \forall\, i:\alpha_i = 1 \quad (5)$$

$$A_{i,i}P_s(i) - \sum_{j\in\mathcal{N}(i)} A_{i,j}\alpha_j\frac{P_r(j,i)}{1-\alpha_i} \leq 0, \qquad \forall\, i:\alpha_i = 0. \quad (6)$$

Let α^* be a Nash equilibrium for the game problem and assume, for simplicity, that $A^*(\alpha^*) = \mathcal{N}$. (The case where

$A^*(\alpha^*) \neq \mathcal{N}$ will be studied in Section IV). Let $\underline{\alpha}$ be a column vector whose i^{th} entry is α_i. Also introduce $\mathbf{G}(\underline{\alpha}) := \frac{\partial}{\partial \alpha_i} U_i(\underline{\alpha}) = A_{i,i} P_s(i) - \sum_{j \in \mathcal{N}(i)} A_{i,j} \alpha_j \frac{P_r(j,i)}{1-\alpha_i}$.

A. Effect of Imposing Power Constraints

Till now we have not imposed any restriction on the possible values that α_i's are allowed to take (except that $\alpha_i \in [0,1]$). Since the nodes are battery power constrained, one would like to see the effect of imposing a constraint on α_i so as to use the battery power efficiently. A natural candidate for such a constraint for node i is $T_i \alpha_i + R_i(1-\alpha_i)P_r(i) \leq P_i$, where T_i and R_i are the average power required for transmission and reception of packets, P_i is the average battery power of node i and $P_r(i)$ is the probability that node i is trying to receive while it is not transmitting. $P_r(i) = 1 - \Pi_{j \in \mathcal{N}(i)}(1-\alpha_j)$, i.e., that a node spends R_i amount of power whenever there is a transmission attempt from at least one neighboring node. In practice, the case of interest would be $T_i \geq P_i \geq R_i$. (If $P_i \geq \max(T_i, R_i)$ then, effectively, the α_i's are not battery power constrained.) Note now that the action space of the nodes are dependent on the actions of other nodes. The existence of a Nash equilibrium would follow if the constraint set so obtained is convex [7]. For a general network topology, it can be shown that the constraint defining functions $P_i - T_i\alpha_i - R_i(1-\alpha_i)P_r(i)$ are quasi-concave [8] so that the constraint set is convex. Further, the constraint set is easily seen to be nonempty because the point $\alpha_i = 0, \ \forall i$ is always feasible. Proof of quasi-concavity of the power constraint functions is detailed in [9].

III. A Distributed Algorithm

To compute α_i, the Kuhn-Tucker condition of Equation 4 suggests the following (gradient ascent type) iteration

$$\underline{\alpha}(n+1) = \underline{\alpha}(n) + a(n)\mathbf{G}(\underline{\alpha}), \tag{7}$$

where $\{a(n)\}$ are the usual stochastic approximation step-size schedules, i.e., positive scalars satisfying $\sum_n a(n) = \infty, \ \sum_n a(n)^2 < \infty$. By the standard 'o.d.e. approach' to stochastic approximation [10], this tracks the asymptotic behavior of the ordinary differential equation (o.d.e.) $\dot{x}(t) = \mathbf{G}(x)$. For a general network and coefficients $A_i, A_{i,j}, C_i$, the stability of the equilibrium points of the o.d.e. cannot be a priori assumed (see also [7] for this issue). However, for a special case where all the nodes are neighbors of each other, it can be shown that the (slightly modified) o.d.e. is globally asymptotically stable and hence the suggested iteration above is guaranteed to converge irrespective of the coefficients.

A. The Case of All Nodes Neighbors of Each Other

Consider the special case where for any node $i, \mathcal{N}(i) = \mathcal{N}$, i.e., all the nodes are neighbors of each other. This is a common scenario in wireless LANs spanning a small area (office etc.). The standard Slotted ALOHA system is yet another example of such scenario.

Recently, [2] has also considered a game theoretic approach to delay minimization in Slotted Aloha systems with the *retransmission probabilities* as decision variables. However, it does not consider the problem of nodes computing the optimal retransmission probabilities. The problem there also assumes symmetry, i.e., (unlike our case) all nodes have equal weightage thus resulting in equal optimal retransmission probabilities for each node.

For the present case where $S(i)$ is empty, it is seen that $P_s(i) = \Pi_{j \neq i}(1-\alpha_j)$, and $P_r(j,i) = \Pi_{k \neq j}(1-\alpha_k) = P_s(j)$. Note that $P_r(j,i) = P_r(j,k)$ for any $k, i \neq j$. The condition in Equation 4 is rewritten as $\sum_{j \neq i} A_{i,j} \frac{\alpha_j}{1-\alpha_j} = A_{i,i}, \ \forall i$. Let $\beta_j := \frac{\alpha_j}{1-\alpha_j}$, $\zeta_{i,j} = A_{i,j}, \ i \neq j$ and $\eta_i = A_{i,i}$. This condition in matrix form is,

$$\zeta \underline{\beta} = \underline{\eta}. \tag{8}$$

Remark: Equation 8 gives a complete characterization of the solution of the optimization problem under consideration as a solution to a set of linear equations. This is a considerable simplification given the complex set of equations representing the optimization problem. Now we proceed to give a method to compute this optimum in a distributed manner.

B. The Algorithm

To solve Equation 8, the iteration to be considered is $\underline{\beta}(n+1) = \underline{\beta}(n) + a(n)(\underline{\eta} - \zeta\underline{\beta}(n))$, corresponding to the o.d.e. $\dot{x}(t) = (\underline{\eta} - \zeta x(t))$, whose stability, again, can not be apriori assumed. We thus consider the modified o.d.e. having same critical points

$$\dot{x}(t) = \zeta(\underline{\eta} - \zeta x(t)). \tag{9}$$

This will be stable if the matrix ζ is invertible, whence ζ^2 will be positive definite. The solution will be that of linear system

$$\zeta \underline{\eta} - \zeta^2 \underline{\beta} = 0, \tag{10}$$

which then has the same solution as (8). Thus node i will be solving the i^{th} row of (10). The iteration at node i is thus

$$\underline{\beta}(n+1) = \underline{\beta}(n) + a(n)\zeta(\underline{\eta} - \zeta\underline{\beta}(n)). \tag{11}$$

The algorithm run by the nodes based on the above iteration is detailed as follows.

1) Set the slot number $n = 0$. Initialize $\alpha_i^{(0)}, \ 1 \leq i \leq N$, to some small positive values. Also let $N(i) = 0$, the last slot number when node i updated its attempt probability.
2) For $1 \leq i \leq N$, node i does the following operations:
   - It either decides to transmit with probability $\alpha_i^{(n)}$, or decides to receive with probability $1 - \alpha_i^{(n)}$.
   - If decided to transmit, a node sends the data packet destined for all the neighboring nodes. It also transmits the information relevant for other nodes for updating their attempt probabilities based on (11). In particular, node i transmits $\alpha_i^{(n)}, N(i), \mathcal{N}(i)$.

- If decided to sense the channel, do the following:

• If node i receives a signal that can be decoded correctly, then it checks if the received signal contains the update information. If yes, update node i's local information about its neighbors based on information extracted from the transmission received from its first hop neighbor. Here node i updates its estimate of $\alpha_j,\ j \neq i$, only if the value of $N(j)$ that it has now received is more than node i's copy of $N(j)$.

• Update $\alpha_i^{(n)}$ based on the updated information.

• Set $N(i) = n$.

3) $n = n + 1$, Go to step 2.

IV. Numerical Results

In this section we present some numerical results from a software implementation of the algorithm. The network parameters are chosen to motivate certain possible problems that can arise when using the proposed algorithm. Modifications to the proposed algorithm to solve these problems are simultaneously provided in this section.

In all the results presented here the learning parameter was $a(n) = \frac{0.1}{(n \bmod 1e5)+1.0}$. The periodic nature of $a(n)$ used is to ensure that a transient (owing to node failure or mobility) dies out quickly. Note that this does not satisfy the usual stochastic approximation constraints. However, the periodic resetting of the learning sequence can be used for a time-varying situation as long as this variation is on a sufficiently slow timescale.

Unique Solution: Inactive Constraint Consider the symmetric case where $A_{i,j} = K$ for some K. For this case the ζ^2 matrix is invertible and hence unique solution for the system of linear Equation 10 exists and is given by: $\beta_i = \frac{1}{N-1},\ \ \forall i$ hence $\alpha_i = \frac{1}{N},\ \ \forall i$. This is also evident from simulation results shown in Figure 1.

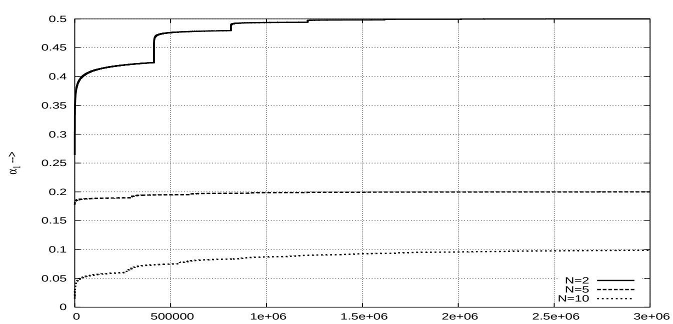

Fig. 1. Plot showing node 1's estimate of attempt probability as a function of simulation time for different values of $N = 2, 5$ and 10. The weights $A_{i,j}$ were same for all pairs (i,j).

Consider the case where $N = 2$ and $A_{i,j} = \frac{1}{\log(i+j)}$. For this problem again the ζ^2 matrix is invertible and hence the unique solution for the system of linear equations exists. Figure 2 gives the simulation result for this case. Now we consider the case where $N = 3$ and $A_{i,j} = \frac{1}{(i+j)}$. The unique solution for the system of linear equations exists. Figure 3 gives the simulation result for this case. These three examples provide cases where the algorithm of Section III-A works without any modification. Next we present some cases where the algorithm requires some modification.

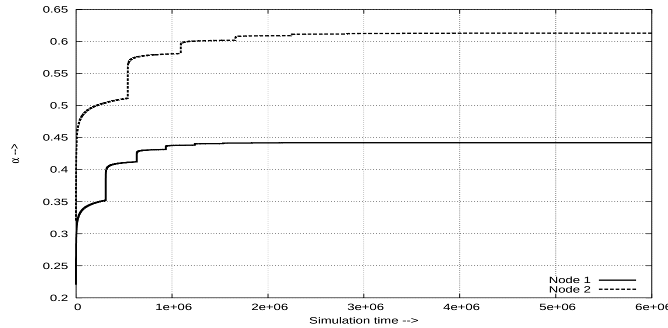

Fig. 2. $N = 2$, the weights were $A_{i,j} = \frac{1}{\log(i+j)}$.

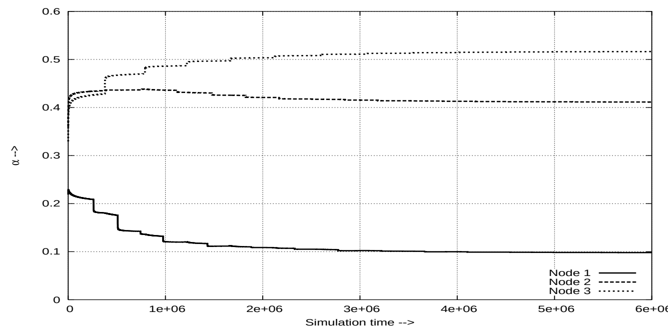

Fig. 3. $N = 3$, the weights were $A_{i,j} = \frac{1}{i+j}$.

Unique Solution: Active Constraint Note that in the network parameters considered so far, the Nash equilibrium were feasible (as they were strictly positive and less than 1). Consider now the case where $N = 4$ and $A_{i,j} = \frac{1}{\log(i+j)}$. For this system, it can be seen that for this case the unique solution to the Kuhn-Tucker condition is $\alpha_1 = -0.269,\ \alpha_2 = 0.307,\ \alpha_3 = 0.415,\ \alpha_4 = 0.458$. However, a negative attempt probability is certainly not a feasible solution. Note also that negative value of α_1 implies that node 1 will have an effective attempt probability of 0. While $\alpha_i > 1$ implies that node i will be transmitting all the time and has no chance to receive. Thus, to avoid these undesirable behavior, we modify the algorithm as follows:

1) Each node i keeps a lower and an upper bound on the attempt probability $0 < \alpha_{i,min} < \alpha_{i,max} < 1$.
2) If in any slot node i's computed value of α_i is below a lower bound on the attempt probability, i.e. $\alpha_i \leq \alpha_{i,min}$, it changes the computed α_i to $\alpha_{i,min}$. $\alpha_{i,max}$ is also used similarly.
3) The probabilities associated with nodes that have hit the constraint boundaries continue to be updated subject to the constraint that they remain in the permitted range. This is to take care of any freak episodes of hitting the constraint boundary due to 'noise', errors, etc.

Figure 4 gives the simulation result for this case with $\alpha_{i,min} = 0.001$. It is seen that node 1 achieves the lower bound and stays there while the other nodes attain their (now modified) equilibrium.

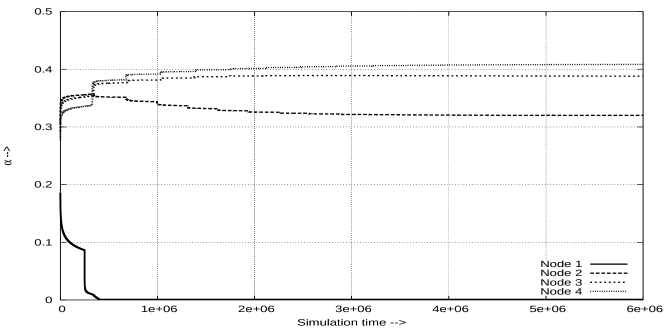

Fig. 4. $N = 4$, the weights were $A_{i,j} = \frac{1}{\log(i+j)}$.

Nonunique Solution Consider the case where $N = 3$ and the coefficient matrix is given by $A = [1\ 1\ 0; 1\ 1\ 1; 0\ 1\ 1]$. For this case there can be multiple solutions to the Kuhn-Tucker condition. Figure 5 shows the values of α_1 and α_3 as obtained from simulation of the proposed algorithm for different initial values. Note from the figure that the initial value of α affects the equilibrium point achieved. Owing to symmetry of the problem, it is desirable to have $\alpha_1 = \alpha_3$, however the proposed algorithm fails to achieve this fairness. Thus, non-uniqueness of the solution leads to two undesirable features, i.e., dependence on initial condition and possible unfairness. To avoid this problem we modify the algorithm so

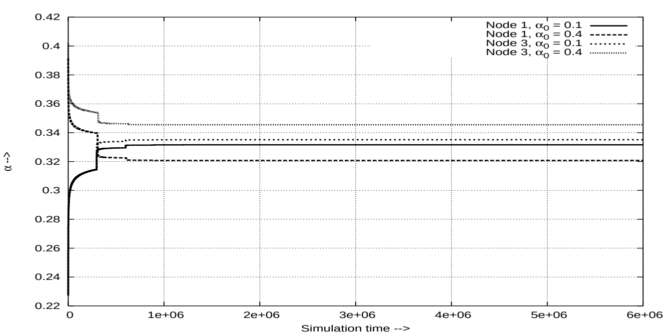

Fig. 5. Estimates of attempt probability for $N = 3$.

that each node adds a small value ϵ to the diagonal elements of its estimate of the ζ matrix. Figure 6 gives simulation results using $\epsilon = 0.01$. The algorithm converges to a value of $\alpha_1 = \alpha_3 = 0.329$ irrespective of the initial value of α_i. Thus it is seen from the simulations that the proposed slight modification solves both the problems mentioned above, i.e., that of dependence on initial value and that of unfairness.

V. Conclusion

We considered the problem of a node computing its own optimal channel access rate in a random access network with two-way traffic. We considered a realistic scenario where a node is interested in both receiving as well as transmitting packets. We proved existence of Nash equilibrium for our problem as well as for the case where nodes are battery power

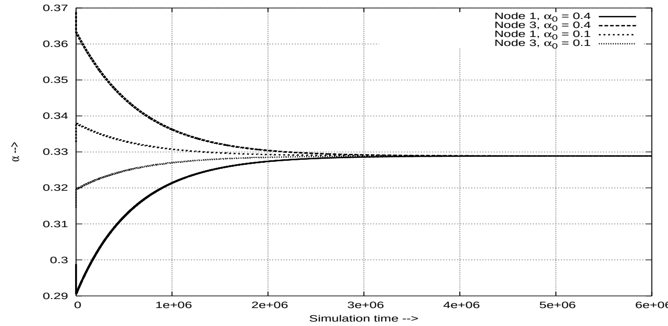

Fig. 6. Estimates of attempt probability for $N = 3$.

constrained so that the constraint set (action space) of different nodes is coupled. We proposed a distributed scheme for adapting random access for the mentioned scenario. The advantage of the scheme is its simplicity thus making it attractive from implementation point of view.

The presented iteration is guaranteed to be stable by exploiting the special structure of the case where a node is neighbor of all the other nodes (as is frequently in the case of Wireless LANs); this also required a modification to the initial iteration of Equation 7. The actual iteration of Equation 7, though it can be used for a general network, is not guaranteed to converge. We are now considering a development similar to that presented here for general topologies.

References

[1] IEEE Computer Society LAN MAN Standards Committee, "Wireless lan medium access control (mac) and physical layer (phy) specifications.," IEEE Std. 802.11-1997. New York, 1997.

[2] E. Altman, R. El Azouzi and T. Jimenez, "Slotted Aloha as a Stochastic Game with Partial Information," in *WiOpt'03, Sophia Antipolis, France, 2003.*

[3] V. Srinivasan, P. Nuggehalli, C. Chiasserini and R. Rao, "Cooperation in wireless ad hoc networks.," in *INFOCOM 2003*.

[4] P. Chaporkar and S. Sarkar, "Stochastic control techniques for throughput optimal wireless multicast," in *Control and Decision Conference (CDC), Maui, Hawaii, December 9-12, 2003*.

[5] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Information Theory*, 2000.

[6] V. S. Borkar and A. A. Kherani, "Random Access in Wireless Ad Hoc Networks as a Distributed Game," in *WiOpt 2004*.

[7] J. B. Rosen, "Existence and uniqueness of equlibrium points for concave n-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, July 1965.

[8] A. Mas-Colell, M. D. Whinston and J. R. Green, *Microeconomic Theory,* Oxford Univ. Press, 1995.

[9] "`www-sop.inria.fr/mistral/personnel/Arzad-Alam.Kherani/pimrc_ex.ps`".

[10] H. J. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications,* Springer-Verlag, 1997.