A Distributed Algorithm to achieve Cesaro-Wardrop Equilibrium in Wireless Sensor Networks

M. Farukh Munir*, Arzad A. Kherani** and F. Filali*
*Institut Eurécom, Department of Mobile Communications, Sophia Antipolis, France
**Indian Institute of Technology Delhi, New Delhi India.
Email: {munir,filali}@eurecom.fr, alam@cse.iitd.ernet.in

***Abstract*—Stability and delay constraints have significant impact on the design and operation of wireless sensor networks. In this paper, we propose a closed architecture for data sampling in wireless sensor networks. Examples show that the proposed scheme outperforms the traditional layered scheme, both in terms of stable operating region as well as the end-to-end delays.**

We then propose a distributed routing scheme for a broad class of wireless sensor networks which converges (in the Cesaro sense) to the set of Cesaro-Wardrop equilibria. The scheme is based on multiple time-scale stochastic approximation algorithms. Convergence is established using standard results from the related literature and validated by simulation results. Our algorithm can adapt to changes in the network traffic and delays.

I. INTRODUCTION

Wireless Sensor networks (WSNs) is an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. We consider a WSN in which the sensor nodes are sources of delay insensitive traffic that needs to be transferred in a multi-hop fashion to a common processing center. We propose an adaptive and distributed routing scheme for a general class of WSNs. The objective of our scheme is to achieve Cesaro Wardrop equilibrium, an extension of the notion of Wardrop equilibria that first appeared in [3] in the context of transportation networks. It states that the journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route. This notion is defined in Equation (1) later in this paper.

The initial motivation for us for this problem was the challenge of designing a routing protocol which can adapt to the traffic requirements in a sensor network as these can change over time fur to energy outtage. Keeping the finite energy source at each node in mind, it is valuable to use several paths simultaneously in carrying traffic from a sensor to a given sink. Further, using multiple routes by flow-splitting can result in a fair consumption of available energy sources. Our algorithm is actually an adaptation of the algorithm proposed in [1] to the case of WSNs. In the algorithm of [1], each source uses a two time-scale stochastic approximation algorithm. Difference in the two algorithms are:

Institut Eurécom's research is partially supported by its industrial members: BMW Group Research & Technology - BMW Group Company, Bouygues Télécom, Cisco Systems, France Télécom, Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales.

1) In WSNs that we consider, each node has an attribute associated with it namely the channel access rate. The delay on a route depends on the attributes of the nodes on the route. However, in order to maintain some long term data transfer rate, each node needs to adapt its attribute to routing.
2) The difference in time scales that we use for various learning/adaptation schemes helps us prove convergence of our algorithm (such a proof is not present in [1]).

The organization is as follows. The network and traffic model is presented in Section II. Section III analyzes two different approaches for data sampling in WSNs. In Section IV, we present our routing algorithm which converges to a Wardrop equilibrium. Results from simulations are presented in Section V. Section VI concludes the paper and outlines future work.

II. NETWORK AND TRAFFIC MODEL

In this paper, we consider a static WSN with n sensor nodes. Given is an $n \times n$ neighborhood relation matrix N that indicates the node pairs for which direct communication is possible. We will assume that N is a symmetric matrix, i.e., if node i can transmit to node j, then j can also transmit to node i. For such node pairs, the $(i, j)^{th}$ entry of the matrix N is unity, i.e., $N_{i,j} = 1$ if node i and j can communicate with each other; we will set $N_{i,j} = 0$ if nodes i and j can not communicate. For any node i, we define $N_i = \{j : N_{i,j} = 1\}$, which is the set of neighboring nodes of node i. Similarly, the two hop neighbors of node i are defined as $S_i = \{k \notin N_i \cup \{i\} : N_{k,j} = 1 \, for\, some\, j \in N_i\}$. Note that S_i does not include any of the first-hop neighbors of node i.

Each sensor node is assumed to be sampling (or, sensing) its environment at a predefined rate; we let λ_i denote this sampling rate for node i. The units of λ_i will be $pkts/s$, assuming same packet size for all the nodes in the network. In this work, we will assume that the readings of each of these sensor nodes are statistically independent of each other so that distributed compression techniques are not employed.

Each sensor node wants to use the sensor network to forward its sampled data to a *common* fusion center. Thus, each sensor node acts as a forwarder of data from other sensor nodes in the network. We will assume that the buffering capacity of each node is infinite, so that there is no data loss in the network. We will allow for the possibility that a sensor

1-4244-1457-1/08/$25.00 © IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.

node discriminates between its own packets and the packets to be forwarded.

We let ϕ denote the $n \times n$ routing matrix. The $(i, j)^{th}$element of this matrix, denoted $\phi_{i,j}$, takes value in the interval $[0, 1]$. This means a probabilistic flow splitting as in the model of [2], i.e., a fraction $\phi_{i,j}$ of the traffic *transmitted* from node i is forwarded by node j. Clearly, we need that ϕ is a stochastic matrix, i.e., its row elements sum to unity. Also note that $\phi_{i,j} > 0$ is possible only if $N_{i,j} = 1$. *Our objective in this paper is to come up with an algorithm using which any node (say i) is able to converge to the corresponding row of the matrix ϕ corresponding to the Wardrop equilibrium.*

III. Data Collection Mechanism

There are various ways of achieving the average sampling rate of λ_i for all the nodes. We will see later in the paper the qualitative behavior of a Wardrop equilibrium in sensor networks depends crucially on the data collection mechanism employed. In this paper, we consider two possibilities of data collection mechanism:

A. Open System (Layered Architecture)

This is the traditional slotted Aloha based system with a layered architecture where the application layer (sampling process for WSNs) does not directly interact with the lower layers (the random access MAC in our case). Such schemes were extensively used in the Packet Radio Network literature. The analysis of the model that we consider above is also available in the PRN literature (see for example [2]). The problem of stability that we will see is that for a given *sampling rate*, one needs to jointly optimize the *channel access rate* and the routing in order to optimize on delays. We will also see that the sampling rate at a node may be restricted by the sampling rate of the other *downlink* nodes. Further, in order to maintain stability of a node's transmit buffer, one needs to be operating far from the maximum allowed sampling rate (this is because, under the assumption of Bernoulli sampling process, the average queue length grows exponentially with an increase in the sampling rate). In addition, in this model, the sampling rate is not directly related to the channel access rates (unless it is an outcome of an optimization problem like the one we consider). Thus, there is an extra dimension that needs to be optimally controlled.

B. Closed System (Cross-layered Architecture)

Under this mechanism, there is a strong coupling between the channel access process and the sampling process. This approach has the advantage that one does not need to find an optimal sampling rate all over again on changing the channel access rates. The coupling (*cross-layer* optimization) automatically regulates the sampling process for any change in the channel access process.

The combined channel access/data sampling mechanism is as follows: Node i decides to attempt a channel access with probability α_i in any slot (else, it is sensing the channel for any possible transmissions). If decided to attempt a transmission, the node first checks if there is any packet available in its transmit queue. We have following possibilities:

1) No packets waiting in the transmit queue: In this case, the MAC layer of node i will ask the appropriate upper layer to sense data and provide it with a new packet. This packet is then attempted a transmission.
2) At least one packet waiting to be forwarded: In this case, node i will serve the head-of-line packet from its transmit queue.

Note that under this mechanism the transmit queue of node i can have at most one packet in the transmit queue that was generated at node i. It can however have multiple packets in the transmit queue to be forwarded, i.e., those packets that were initially generated at some other node, and have arrived at node i to be forwarded. Clearly, under this scheme if the transmit queue of node i contains a packet that was generated at node i itself, then this packet will be the head-of-line packet till the time it leaves the transmit queue of node i.

C. Applications for Closed System

The closed scheme is meant to be used in applications where a WSN is used to observe the time variation of a random field over the space on which the network is deployed. For such applications, one can think of a temporal priority mechanism for transmitting packets so as to reduce the overall transmissions in the network. In particular, our sampling scheme amounts to the assumption that a node assigns highest priority to the most recent packet generated by the node (this priority is defined over the packets generated by the node, and does not include the packets that a node receives to forward).

A complete stability analysis of the above mentioned data collection schemes is presented in [7], and hence, not repeated here. For the Open system, we will assume a given set of channel access rates. We will see that the routing algorithm is able to select a good operating point that guarantees stability (as long as such a point exists for the given value of channel access probabilities).

IV. The Routing Algorithm for Open and Closed System

We assume that the system operates in discrete time, so that the time is divided into (conceptually) fixed length slots. The system operates on CSMA/CA MAC. Assuming that there is no exponential back-off, the channel access rate of node i (if it has a packet to be transmitted) is $0 \leq \alpha_i \leq 1$. Thus, α_i is the probability that node i, if it has a packet to be transmitted, attempts a transmission in any slot. A node can receive a transmission from its neighbor if it is not transmitting and also no other neighboring node is transmitting.

Under the above model there will be a delay, say $y_{j,i}$ of the packet from node j to be served at node i; this packet could have originated at node j or may have been forwarded by node j. The Expected delay of a packet transmitted from node j is thus $\sum_{i \neq j} \phi_{j,i} y_{j,i}$. Since delays are additive over a path, packets from any node will have a delay over any possible route to the fusion center. A route will be denoted by an ordered set of nodes that occur on that route, i.e., the first element will be the source of the route, the last element will be the fusion center and the intermediate elements will be nodes arranged in the order that a packet traverses on this route. Let the total number of possible routes (cycle-free) be R. Let route i, $1 \leq i \leq R$ be denoted by the set $\mathcal{R}_i$ consisting of R_i elements with $\mathcal{R}_{i,j}$ denoting the j^{th}entry of this route. Then, a traffic splitting matrix will correspond to a Wardrop equilibrium iff for any i

$$\sum_{1\leq j\leq R:\mathcal{R}_{j,1}=i}\left(\prod_{k=1}^{R_j-1}\phi_{\mathcal{R}_{j,k},\mathcal{R}_{j,k+1}}\right) \left(\sum_{k=1}^{R_j-1}y_{\mathcal{R}_{j,k},\mathcal{R}_{j,k+1}}\right)=\sum_{k=1}^{R_l-1}y_{\mathcal{R}_{l,k}\mathcal{R}_{l,k+1}}, \quad (1)$$

for any l with $\mathcal{R}_{l,1}=i$ and such that $\prod_{k=1}^{R_l-1}\phi_{\mathcal{R}_{l,k},\mathcal{R}_{l,k+1}} > 0$, i.e., the delays on the routes that are actually used by packets from node i are all equal.

A. Open System

Nodes iteratively keep updating the one-hop routing probabilities based on the delays incurred for every possible path. Let $\phi(n)$ denote the traffic splitting matrix at the beginning of the n^{th} time slot. Node i does some computation to update the i^{th} row of this matrix. Let $Y^k(n)(\mathcal{R}_{k,1}=i)$ be the new value of the delay of a packet sent by sensor i through route $k(i=\mathcal{R}_{k,1})$. Node i keeps an estimate of the average delay on route k.

$$y^k(n+1)=(1-a)y^k(n)+aY^k(n). \quad (2)$$

Further, after calculating the expected delays at the start of a time slot, each node adapts its routing probabilities to the new expected delays as follows,

$$\phi_{i,\mathcal{R}_{k,2}}(n+1)=(1-b)\phi_{i,\mathcal{R}_{k,2}}(n)+ b\left(\sum_{1\leq l\leq R:\mathcal{R}_{l,1}=i}y^l(n)\phi_{i,\mathcal{R}_{l,2}}(n)-y^k(n)\right) \quad (3)$$

B. Proof of Convergence to Wardrop Equilibrium

We will assume that the learning parameters a and b are such that $a \ll b$. This brings us in the two-level stochastic approximation algorithm framework and, following standard results [4], the update of the traffic split will see the average delays y^l as static so that the effect of the second update will be that all the traffic from node i will be directed to the smallest delay route. The algorithm for updating the delay estimates over route will thus see no effect of the dynamics of the second update scheme except that the statistical properties of the random variables will come from the splitting vector in which each node directs all its traffic on one of the possible routes from the node to the fusion center; note that in general different nodes will be choosing different routes. Thus, by the standard o.d.e. approach to stochastic approximation algorithms [4], [5], the delay updating algorithm will behave like an autonomous ordinary differential equation. The convergence of this differential is guaranteed using arguments similar to those used in [6]. Since the point of convergence satisfies the defining condition of the Wardrop equilibrium, the proposed algorithm will converge to the Wardrop equilibrium. *Note* that this convergence is for the average of delays, this is what we mean by Cesaro-Wardrop equilibrium.

C. Closed System

The updates for this system are going to be the same as that for the Open system. Only new complication here is that one needs to tune the channel access rates, $\alpha_i' s$, also in order to guarantee the long term average data sampling rate. This is easily done because the nodes know (or, can estimate) the statistics of the traffic they are getting from the other nodes and also the success rate of its own transmissions to various neighbors. Using this estimate a node can easily tune its channel access rate to guarantee itself a preset data sampling rate.

In each time slot, i^{th} sensor tries to hold channel for transmission with probability α_i. If the node tries to hold channel in a time slot, it either succeeds in transmitting or fails. If the node succeeds, then if the packet transmitted can be the one which is generated at the current node or it may be the one which the node received from any of the neighboring nodes. Let $n(k)$ be the number of slots in which node has successfully transmitted a packet generated by itself in total k slots. $\hat{\lambda}_i^k$ is the rate of transmission node is able to provide in the k^{th} slot, $\hat{\lambda}_i^k=\frac{n(k)}{k}$.

$$\alpha_i^{k+1}=max\left\{min\left[\alpha_i^k+c\left(\lambda_i-\hat{\lambda}_i^k\right),1\right],0\right\} \quad (4)$$

Where c is a positive learning parameter. Delay and routing probability learning will remain as was in the Open System.

D. Practical Considerations

Delay estimation of paths by a node in every slot can be done by having power of sink so large that it can reach all the sensors in one-hop. Therefore, the sink can *ACK* all the incoming packets so that the sensors will get estimation of the delay incurred by their packets. In cases, where the sink can not transmit an ACK in one-hop, it will be transmitted in a multi-hop fashion. The ACK in multi-hop case follows the same path as the data-packet. However, this will only result in a slow convergence to the Wardrop equilibrium.

V. Implementation Results

We consider a 6-node WSN shown in Fig. 1. It is easily seen that $\phi_{1,0}=\phi_{2,0}=\phi_{4,0}=1$, node 0 being the common destination for all the packets generated in the network. Node 3 can transmit to 1 and 2. Node 5 can transmit to 1 and 4. The routing algorithm thus has to find appropriate value of $\phi_{3,2}$ and $\phi_{5,4}$ in order that the traffic flow in the network corresponds to a Wardrop equilibrium. We consider this simple network to clearly demonstrate the effect of delay and routing learning probabilities. However, the distributed routing algorithm is able to converge to a Wardrop equilibrium for *any-scale* random deployment of WSNs.

Apart from a demonstration of the convergence of the proposed algorithm, we will see in this section that the data

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.

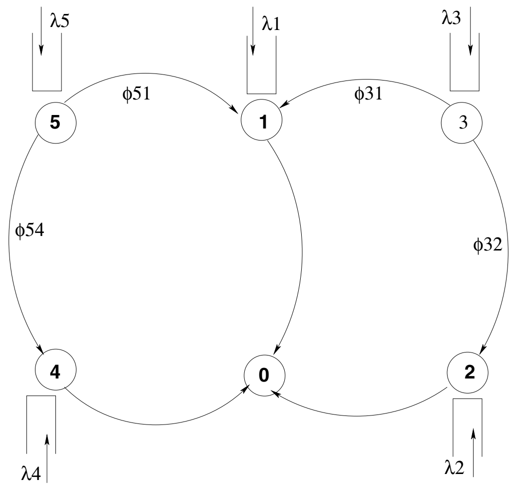

Fig. 1. The Simulated Network consisting of 5 sensors and 1 sink.

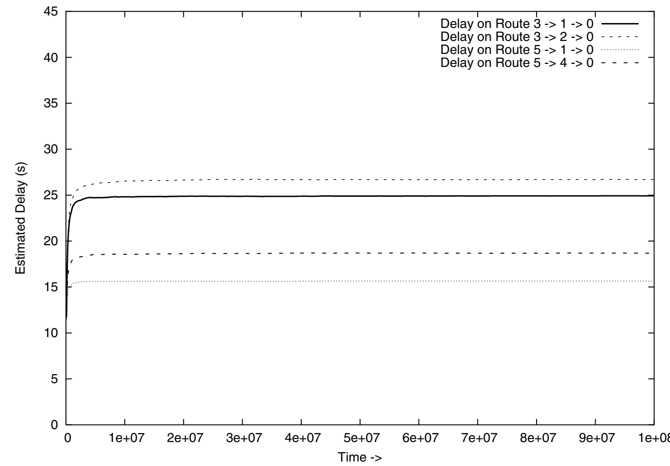

Fig. 2. *Delays* incurred on different routes for open system.

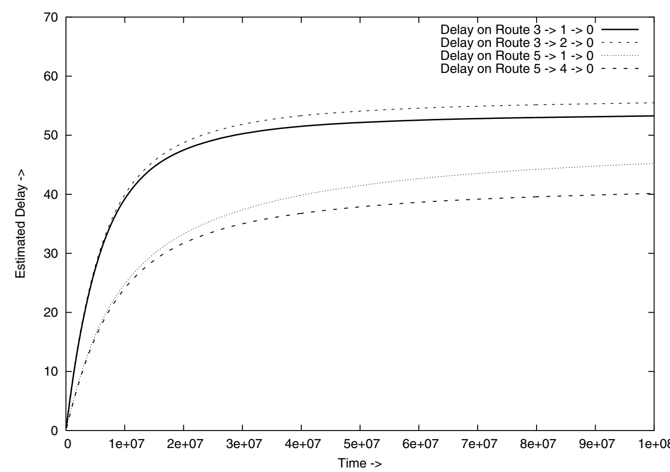

Fig. 3. *Delays* on different routes for open system w.r.t *changing* $\alpha_i^{\prime}s$.

sampling rates that a network can support using the Open architecture is very small. This is essentially because of the stability constraints on the channel access rates. On the other hand, the Closed system can support higher data sampling rates because of the fact that it is essentially self-regulating, guaranteed to be stable while maintaining large data sampling rates; this is because a node generates a new packet only if it has no other packet in the queue. This however does not mean that the Closed system can support arbitrary data sampling rates. Characterizing the *exact* stability region for both systems is an ongoing work. We have implemented the Open as an application layer module, Closed system as a cross-layer (Application-MAC) module, and routing algorithm is incorporated at the routing layer in TinyOS [8].

A. Observations from Open System

In Fig. 2 and 3 we plot, against the slot number, the average delays on the four routes $3 \rightarrow 2 \rightarrow 0$, $3 \rightarrow 1 \rightarrow 0$, $5 \rightarrow 4 \rightarrow 0$, and $5 \rightarrow 1 \rightarrow 0$ for the open system. The data sampling rates were set at $\lambda_1 = 0.01$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$, $\lambda_4 = 0.01$, and $\lambda_5 = 0.04$. Note that the data sampling rates are small. We were forced to select small data rates in order to guarantee stability of the nodes in the network. The channel access rates were set to $\alpha_1 = 0.2$, $\alpha_2 = 0.2$, $\alpha_3 = 0.2$, $\alpha_4 = 0.15$, $\alpha_5 = 0.1$.

1) The delays on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ are very close to each other, with a very fast convergence. Similarly for routes $5 \rightarrow 4 \rightarrow 0$ and $5 \rightarrow 1 \rightarrow 0$. This shows that the algorithm succeeds in achieving a Wardrop equilibrium.
2) Note the high value of delay on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ even for moderate (or, very small) load on the system.
3) Fig. 3 shows the delay obtained by varying the channel access rates to $\alpha_i = 0.1$ for $i = 1, ..., 5$, and $\lambda' s$ remaining the same as earlier. The estimated delays show the *sensitivity* to channel access probabilities. Thus, there is a need to carefully tune the $\alpha_i^{\prime}s$. In Fig. 3, we also see that convergence to a *load-balanced* regime (equal delays on all the possible routes) is violated by changing the $\alpha_i^{\prime}s$. As we will see later, this is not a problem in the closed system because the system adapts its channel access probabilities to meet the target traffic and there is no need of further tuning this parameter.

B. Observations from Closed System

Simulation results for the closed system are presented in Fig. 4, 5, and 6. The data sampling rates were set at $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\lambda_3 = 0.1$, $\lambda_4 = 0.005$, $\lambda_5 = 0.1$. Nodes were expected to adapt their channel access probabilities based on the optimal traffic split used by node 3 and 5.

1) The delays on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ are very close to each other, with a fast convergence. This shows that the algorithm succeeds in achieving a Wardrop equilibrium.
2) For routes $5 \rightarrow 1 \rightarrow 0$ and $5 \rightarrow 4 \rightarrow 0$, the delays are

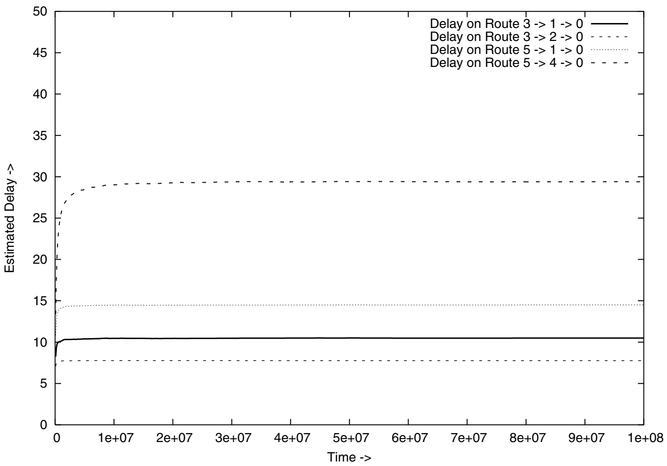

Fig. 4. *Delays* incurred on different routes for closed system.

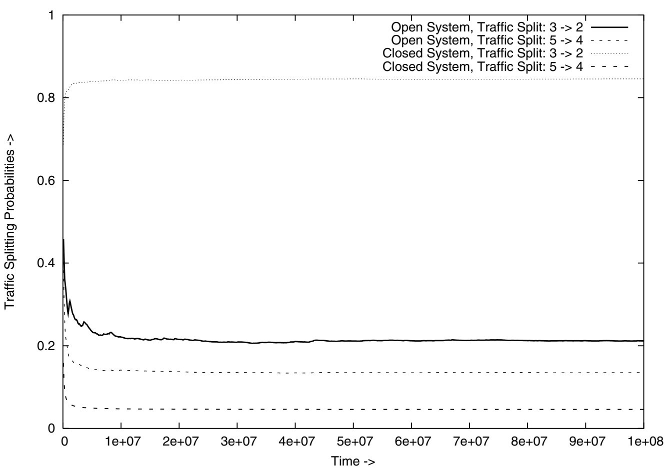

Fig. 5. *Traffic-split* over different routes for open & closed system.

significantly different. This is also reflected in the traffic split obtained by the algorithm, as in Fig. 5 we see that node 5 uses node 1 for most of its traffic, thus obtaining smaller delay. This is again Wardrop equilibrium where the higher delay path is not used (the small $+ve$ value of traffic on route $5 \rightarrow 4 \rightarrow 0$ is imposed by the algorithm to ensure that all the alternatives are probed often enough to cope up with a change in the network).

3) Note the small value of delay on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ even for moderate (or, very small) load on the system. This is to be compared with the corresponding values shown under the results for open system where the delays on these routes were higher even though the average data sampling rates were significantly smaller. Thus, in comparison with the open system, the closed system provides better performance.
4) Fig. 6 shows that the algorithm is also able to adapt the channel access rates in a distributed fashion. It can be checked that the values of $\alpha_i's$ converged-to by the algorithm indeed are just enough to serve the traffic that is offered to the different nodes.

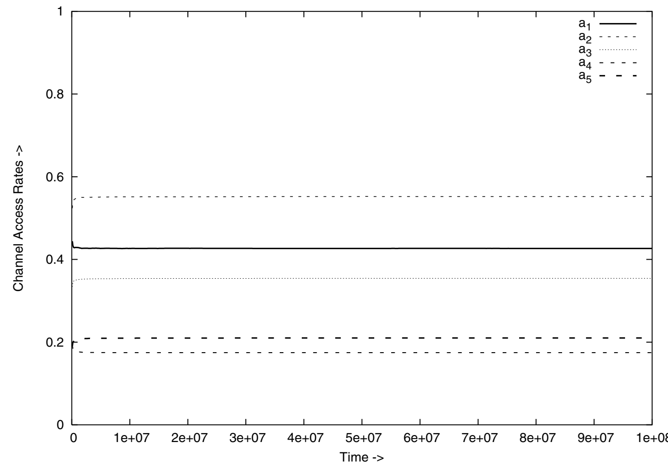

Fig. 6. Convergence of *channel access rates* for closed system.

VI. Conclusions and Future Work

For wireless sensor networks with random channel access, we proposed a data sampling approach that guarantees a *long term* data sampling rate while minimizing the end-to-end delays. Simulation results show that performance of this scheme is better than the traditional layered architecture. We also saw that the proposed scheme does not require tedious parameter tuning as is the case for the layered architecture. We also proposed a learning algorithm, applicable to both the open as well as the closed system, to achieve Wardrop equilibrium for the end-to-end delays incurred on different routes from sensor nodes to the fusion center. For the closed system, this algorithm also *adapted* the *channel access rates* of the sensor nodes.

We are now working on modifications of the algorithm to make it converge to an *efficient* equilibrium. We are also tempted to perform the network *lifetime* analysis as a result of a very fast convergence to Wardrop equilibrium.

References

[1] V. S. Borkar and P. R. Kumar, "Dynamic Cesaro-Wardrop equilibration in networks," IEEE Transactions on Automatic Control, 48(3):282-296, March 2003.
[2] Jr. R. L. Hamilton and H.-C. Yu, "Optimal routing in multihop packet radio networks," In IEEE INFOCOM Conference Proceedings, San Francisco, pp. 389-396, June 1990.
[3] J. G. Wardrop. Some theoratical aspects of road traffic research. In Proceedings Inst. Civil Eng., Part 2, 1952, pp. 325-378.
[4] V. S. Borkar. Stochastic approximation with two time scales. System Control Letters, 29:291-294, 1996.
[5] V. S. Borkar and S. P. Meyn. The O.D.E method for convergence of stochastic approximation and reinforcement learning. SIAM Journal of Control and Optimization, Vol 38, pp. 447-469, 2000.
[6] V. S. Borkar and D. Manjunath. Charge based control of diffserv-like queues. Automatica, Vol. 40, No. 12, December 2004.
[7] M. Farukh Munir, Arzad A. Kherani, F. Filali, On stability and sampling schemes for wireless sensor networks. 5th IEEE/IFIP WiOpt Conference Proceedings, April 16-20, 2007, Limassol, Cyprus.
[8] http://www.tinyos.net/