*Institut Eurécom
Department of Mobile Communications
2229, route des Crêtes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-07-191

Achieving Cesaro-Wardrop Equilibrium in Wireless Sensor Networks

March 12th, 2007

Muhammad Farukh Munir*, Arzad Alam Kherani**, and Fethi Filali*
**Indian Institute of Technology Delhi, New Delhi India.

Tel : (+33) 4 93 00 82 03
Fax : (+33) 4 93 00 82 00
Email : {Muhammad-Farukh.Munir, Fethi.Filali}@eurecom.fr,
alam@cse.iitd.ernet.in

[1]Institut Eurécom's research is partially supported by its industrial members: BMW Group Company, Bouygues Télécom, Fondation d'entreprise Groupe Cegetel, Cisco Systems, France Télécom, Hitachi Europe, SFR, Sharp, ST Microelectronics, Swisscom, Texas Instruments, Thales.

Achieving Cesaro-Wardrop Equilibrium in Wireless Sensor Networks

Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali

Abstract

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 the 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.

Contents

List of Figures

1 Introduction

Wireless Sensor networks is an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. WSNs are becoming increasingly important in recent years due to their ability to detect and convey real-time, in-situ information for many civilian and military applications. Such networks consist of large number of distributed sensor nodes that organize themselves into a multihop wireless network. Each node has one or more sensors, embedded processors, and low-power radios, and is normally battery operated. Typically, these nodes coordinate to perform a common task.

We propose an adaptive and distributed routing scheme for a general class of wireless sensor networks. The objective of our scheme is to achieve Cesaro Wardrop equilibrium, an extension of the notion of Wardrop equilibria that first appeared in [4] in the context of transportation networks. The notion is defined in Equation (3) later in this paper. Our algorithm is actually an adaptation of the algorithm proposed in [1] to the case of wireless sensor networks. In the algorithm of [1], each source uses a two time-scale stochastic approximation algorithm. Difference in the two algorithms are:

1. In wireless sensor networks 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]).

In this paper, we consider a static wireless sensor network 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 packets per second, 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 (assumed to be a part of the network[1]). 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 node discriminates between its own packets and the packets to be forwarded (thus allowing for the model of [3] which considers an Ad Hoc network so that nodes give priority to transmission of their own packets or 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 $\phi$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.

The organization of this work is as follows. In Section 2, we detail the different data collection mechanisms. We propose a distributed routing algorithm in Section 3. Numerical results from TinyOS simulations are presented in Section 4. In Section 5, we briefly conclude the work and outline the future directions.

2 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 work, we consider two possibilities of data collection mechanism:

2.1 Layered System

This is the traditional slotted Aloha based system with a layered architecture where the application layer (sampling process in case of sensor networks) does not directly interact with the lower layers (the random access MAC in our example).

[1]Conceptually, we can assume that this fusion center is also a sensor node, which has 0 sampling rate.

In Section 2.4, we will see the issues with stability in the sensor networks that use the slotted Aloha like random access mechanism for channel access with a sampling process without any communication with the MAC layer. 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 in Section 2.4). Thus, there is an extra dimension that needs to be optimally controlled.

2.2 Closed System

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 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 to some other node. 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.

2.3 Applications for Closed System

The closed scheme is meant to be used in applications where a sensor network 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 to some of its neighbors).

2.4 An Example

Consider a 4-node wireless sensor network shown in Fig. 1. Node 0 is the common destination for all the data generated by the other three sensors, labeled $1, 2, 3$. All the transmission in the network is done only by these sensor nodes; the job of node 0 is to receive data sensed at these sensor nodes. To begin with, we assume that node 3 can not directly transmit to the destination node 0. Node 1 and 2 can communicate with node 0 but not with each other; Node 3 can communicate with both, node 1 and node 2.

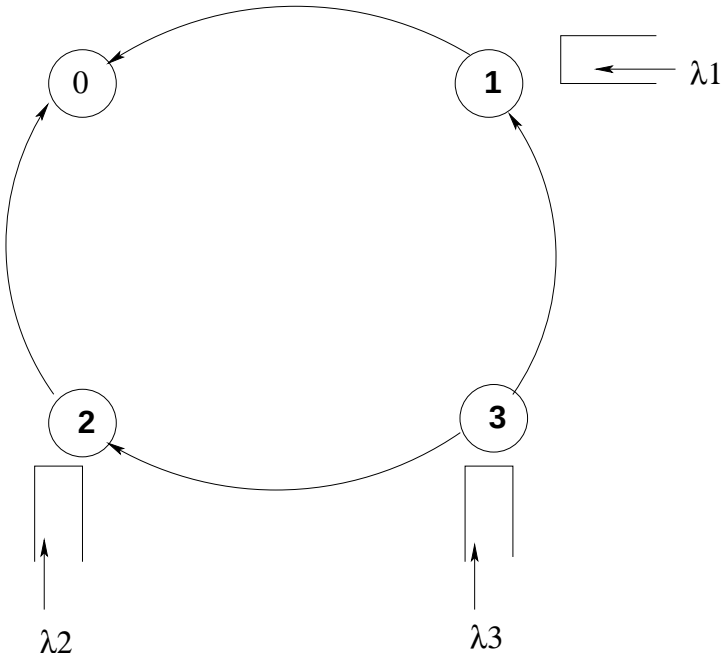

Figure 1: Network Configuration

The time is slotted and the sensor nodes use random access (CSMA/CA like) mechanism for transmission of their data; if node i has a packet to be transmitted,

it attempts a transmission in a slot with some given probability α_i. We will assume that the packet *generation* process at node i is Bernoulli with packet generation probability λ_i[2]

Node 1 and 2 transmit directly to node 0, but one has to decide on the path that packets from node 3 will follow. There are various options for this:

1. either all the packets generated at node 3 will be transmitted to node 1, or to node 2, or

2. for each packet transmitted by node 3, the next hop node is chosen randomly, for example, a packet transmitted from node 3 goes to node 1 with probability 0.3 and to node 2 with probability 0.7.

For this example, we will assume the first option (of course, it is a special case of the second option); we will allow for the second more general option when we come to optimal routing. Traffic splitting method as provided by the second option were also used in PRNs [2]. We will also assume that each packet from a node is attempted transmission till it is successfully received by the intended destination. A transmission is successfully received by a node if it is seeing no other transmission and the node is not transmitting. For cases where one allows for possibility of dropping a packet after it has incurred some number of collisions will not be considered in this work for simplicity; the relevant equations can be found in [3].

Assume that all the packets from node 3 use node 1 to reach node 0. In this case, let π_i denote the steady-state probability that node i has a packet to be transmitted in a slot. We can then write down the following *approximate* equations for the stable system (formal derivation of these equations can be found in [2]).

$$\begin{aligned} \pi_1\alpha_1(1-\pi_2\alpha_2) &= \lambda_1+\lambda_3 \\ \pi_2\alpha_2(1-\pi_1\alpha_1) &= \lambda_2 \\ \pi_3\alpha_3(1-\pi_1\alpha_1) &= \lambda_3 \end{aligned} \tag{1}$$

These equations are approximate because they are derived under a strong decoupling assumption. For stability of all the queues in the network, we need to choose α_i's such that the above system of equations (in π_i's) gives us a solution $(\pi_1,\pi_2,\pi_3)\in[0,1)^3$. The stability condition under which above relations are valid are

[2]Such models were frequently used in context of Packet Radio Network (PRN) literature in the $70'$s and $80'$s, see for example [2]. We will see later that for sensor networks where MAC layer can be allowed to control the application layer, one can achieve better results compared to those in PRNs where application layer operates independently of the MAC layer.

$$\begin{aligned}\alpha_1(1-\alpha_2) &> \lambda_1+\lambda_3\\ \alpha_2(1-\alpha_1) &> \lambda_2\\ \alpha_3(1-\alpha_1) &> \lambda_3\end{aligned}$$

Clearly, for a given sampling rate $\lambda_i, i = 1,2,3$, there will exist many possibilities of the channel access rates that give a stable system. These conditions are actually very different from the one proposed in [2]. In fact, a simple counter example can be given under which the conditions of [2] implies stability, while the system is not stable.

This system is not analytically tractable for the queueing delays. Various approximate analysis can be found in [2] and its references. Because of this reason, the extra degree of freedom that one gets in the parameter α_i is hard to utilize properly as the correct dependence of the system performance (for example, the queueing delays at various nodes) is not known. An instance of this difficulty is that the system of rate balance equations (1) are not valid for all values of α_i. In fact, the discrepancy between the actual system performance and that obtained from using (1) can be as large as 50%. The delay equations provided in [2] and references therein are based on equation (1) and for this reason, these expressions perform poorly for a broad range of parameters α_i[3].

This is clear from the relations (1) which implies that as long as the system is stable, we can solve the rate balance equations (1). Since these equations depend on π_i and α_i only via $\pi_i\alpha_i$, in the stable region this product $\pi_i\alpha_i$ will remain unchanged (w.r.t. changes in α_i). Hence we are tempted to conclude that there is an extra degree of freedom in α_i that can be employed *without changing the end-to-end delays*.

Further, this model was justified in the standard OSI-like model where one did not aim at cross-layer optimization and where the application layer (the sampled voice packets source) was not in control of the MAC layer. If one likes to minimize the expected delay on a node, one way would be to control the arrival of packets from node's own sensing mechanism. One such example that we will be considering (or, proposing) in this work is the following:

A sensor node gets a new packet from the application layer only if it decides to transmit in a slot but finds the transmit queue empty. As is the case with random access, sensor node i decides to attempt a transmission with probability α_i. We will call this system the *closed system* and the first system with layered architecture the *open system*.

[3]We remark here that our present observations are not aimed at questioning the significance of [2] and the related work from PRN literature. Most of these studies never aimed at tuning the parameters α_i, and since they assumed relatively small values of α_i which were fixed *a priori*, most of the time in their work the decoupling approximation leading to equations 1 was good. In our work, however we are trying to get the best system performance, hence need to tune the parameters α_i optimally, so that a correct/accurate analytical model is required for all possible values of α_i's .

Table 1: Node level Delays

Node →	1	2	3
Open system	3.52	2.80	1.45
Closed system	0.56	2.30	2.30

For the Closed System model, the throughput of nodes 2 and 3 are

$$\lambda_2 = \alpha_2 (1 - \alpha_1)$$
$$\lambda_3 = \alpha_3 (1 - \alpha_1)$$

Using these, the throughput of node 1 is

$$\lambda_1 = \alpha_1(1 - \alpha_2) - \lambda_3$$

The stability condition is

$$\alpha_1(1 - \alpha_2) \geq \alpha_3(1 - \alpha_1)$$

The expected number of packets at the three nodes are

$$\begin{aligned} T_1 &= \frac{\rho_0}{(1-\rho)(1-\rho+\rho_0)} \\ T_2 &= \alpha_1 \\ T_3 &= \alpha_1 \end{aligned} \tag{2}$$

Where $\rho_0 = \frac{(1-\alpha_1)\alpha_3+\alpha_1\alpha_2}{\alpha_1(1-\alpha_2)}$ and $\rho = \frac{(1-\alpha_1)\alpha_3}{\alpha_1(1-\alpha_2)}$. The expected delay[4] at each node are easily obtained using Little's Law as $D_i = \frac{T_i}{\lambda_i}$ for $i = 2, 3$ and $D_1 = \frac{T_1}{\lambda_1+\lambda_3}$. It is to be noted that these formula are exact, unlike those in the layered system, where the delay expression available in literature are approximate [2].

The mean node delay at the three nodes in the two systems for $\lambda_i = 0.1$ as obtained from discrete event simulations (for open system) and analysis (for closed system) are shown in Table 1 and 2. The mean delays for the three flows are thus obtained to be:

The mean delays for the closed system were obtained using simple formulae given before in equations (2). For the Open system, since the delay expressions available in literature are approximate, we developed a discrete event simulator to find these delays. The mean delay for the open system was obtained as follows: the simulation was run using different combinations of α_i spanning the stability region of the system. The delay vector provided here is the one which was closest to the origin in terms of Euclidean distance compared to all the other delay vectors

[4]The formal derivation of these expressions are provided in the Appendix.

Table 2: Flow level Delays

Flow →	1	2	3
Open System	3.52	2.80	4.97
Closed System	0.56	2.30	2.86

obtained by varying α_i. Clearly, the flow delay is significantly reduced in the closed system, while using a moderate value of α_i.

Observations from the toy example:

1. The values of α_i for Open system that gives the best performance are very large, thus implying waste of resources due to frequent collisions.

2. The flow delay is significantly reduced in the Closed system, while using a moderate value of α_i.

3. In Open system, one needs to tune the value of α_i in order to get the best delay performance; this may not of much use because the Closed system is giving better results compared to the best result from Open system. Thus, an optimization over α_i in the Open system is not justified. The exact delay expressions are not known. The approximate expressions used in literature are valid only for small values of α_i whereas the optimal point is obtained for large $\alpha_i's$, for which the available approximation has been shown to perform poorly.

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).

3 Routing Algorithms for Different Systems Under Consideration

If the traffic split is not allowed, the objective of the distributed routing algorithm would be to find the shortest delay path between any given source and the fusion center. However, one may allow for traffic split and then try to route the traffic , hoping for a better performance (as the situation without traffic split is a special case of traffic splitting). Under this added freedom of traffic splitting, the routing algorithm is expected to put traffic of a node on those routes for which the delays are smallest and equal. This is what is well known as the Wardrop equilibrium. We propose a stochastic approximation algorithm based distributed algorithm to converge to a Wardrop equilibrium. This algorithm is actually an adaptation of the

algorithm already proposed in [1] to our system for which we can *prove* convergence to Wardrop equilibrium.

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 (see [1] for this definition)

$$\begin{array}{c} \sum_{1 \leq j \leq R: \mathcal{R}_{j,1}=i} \left(\prod_{k=1}^{R_j - 1} \phi \mathcal{R}_{j,k}, \mathcal{R}_{j,k+1} \right) \\ \left(\sum_{k=1}^{R_j - 1} y \mathcal{R}_{j,k}, \mathcal{R}_{j,k+1} \right) = \sum_{k=1}^{R_l - 1} y \mathcal{R}_{l,k} \mathcal{R}_{l,k+1}, \end{array} \tag{3}$$

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.

3.1 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.

$$y^k(n+1) = (1-a) y^k(n) + a Y^k(n). \tag{4}$$

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,

$$\begin{array}{c} \phi_i, \mathcal{R}_{k,2}(n+1) = (1-b) \phi_i, \mathcal{R}_{k,2}(n) + \\ b \left(\sum_{1 \leq l \leq R: \mathcal{R}_{l,1}=i} y^l(n) \phi_i, \mathcal{R}_{l,2}(n) - y^k(n) \right) \end{array} \tag{5}$$

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 [5], 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 [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 [7]. 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.

3.2 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 $n\left(k\right)$ 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,

$$\hat{\lambda}_i^k = \frac{n\left(k\right)}{k} \tag{6}$$

.

$$\alpha_i^{k+1} = max\left\{min\left[\alpha_i^k + c\left(\lambda_i - \hat{\lambda}_i^k\right), 1\right], 0\right\}. \tag{7}$$

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

3.3 Practical Considerations

Here, we will discuss some of the practical aspects of our proposed algorithm. Delay estimation of paths by a node in every slot can be done by having power of the sink so large that it can reach all the sensors in one-hop. Therefore, the sink can acknowledge all the incoming packets so that the sensors will get estimation of the delay incurred by their packets.

4 Implementation Results

We consider a 6-node sensor network shown in Fig. 2. 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. In general, the algorithm is able to converge to a Wardrop equilibrium for *any-scale* random deployment of wireless sensor networks.

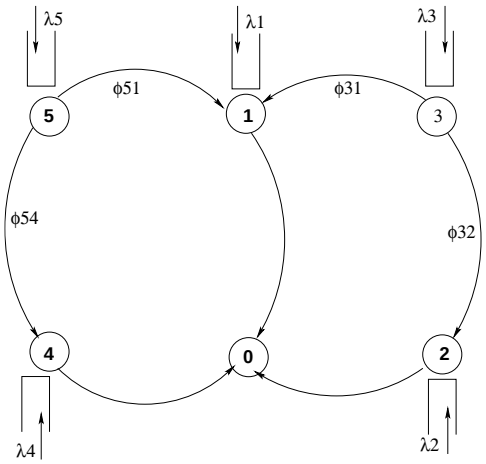

Figure 2: Network Simulated

Apart from a demonstration of the convergence of the proposed algorithm, we will see in this section that the data 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. We have implemented the Open and Closed system as an

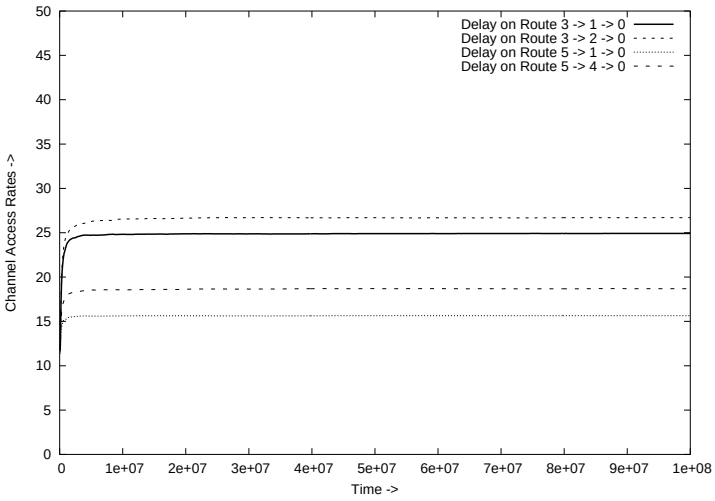

Figure 3: Open System

application layer module in TinyOS [8]. At simulation start up, the nodes learn the network topology and built routes toward the fusion center (sink, node 0). The results presented in this section are the average over several simulation runs.

4.1 Observations from Open System

In Fig. 3 and 4 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 $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$. 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 $\alpha_i \leq 0.2$ for $i = 1, ..., 5$.

1. The delays on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ are very close to each other, with a 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. 4 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 $\alpha_i' s$. In Fig. 4, we also see that convergence to a *load-balanced* regime (equal delays on all the possible routes from a particular source) is violated by changing the $\alpha_i' s$. As we will see later, this

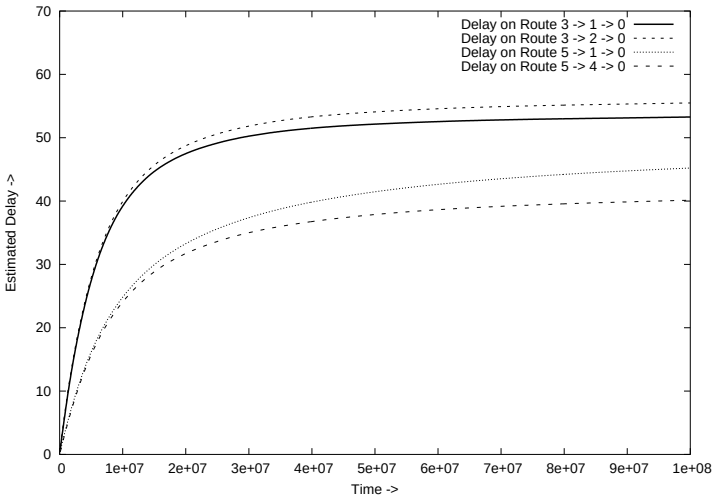

Figure 4: Open System

is not a problem in the closed system because the system adapts its channel acess probabilities to meet the target traffic and there is no need of further tuning this parameter.

4.2 Observations from Closed System

Simulation results for the closed system are presented in Fig. 5, 6, and 7. The data sampling rates were set at $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$. 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 also close to each other, with a fast convergence. This is also reflected in the traffic split obtained by the algorithm, as in Fig. 6 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

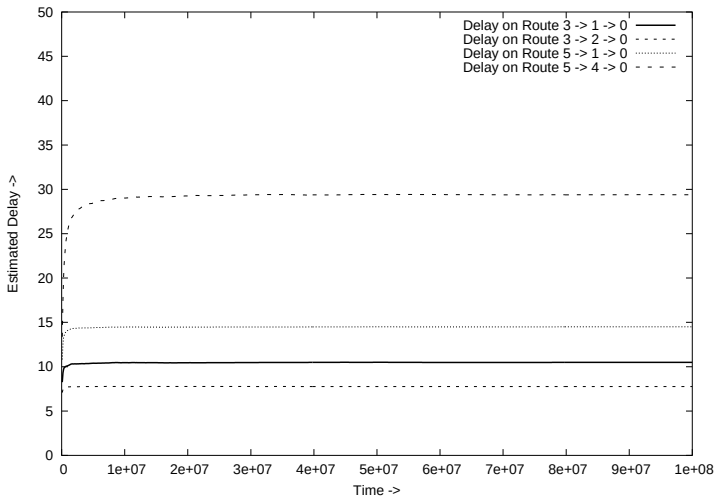

Figure 5: Closed System

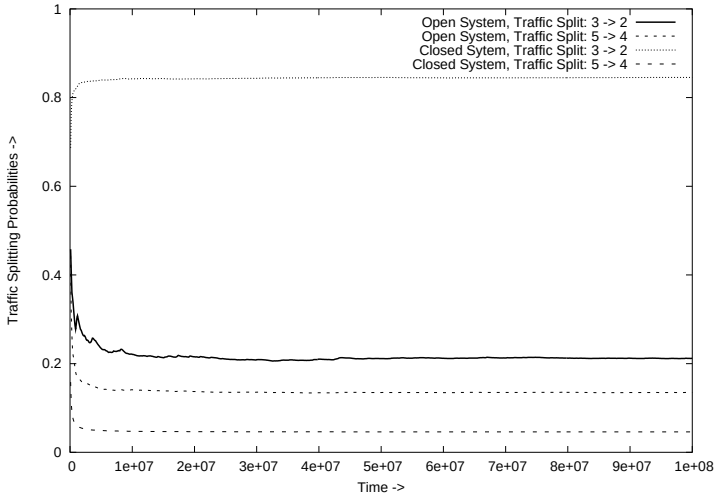

Figure 6: Traffic Split over Differenet Routes

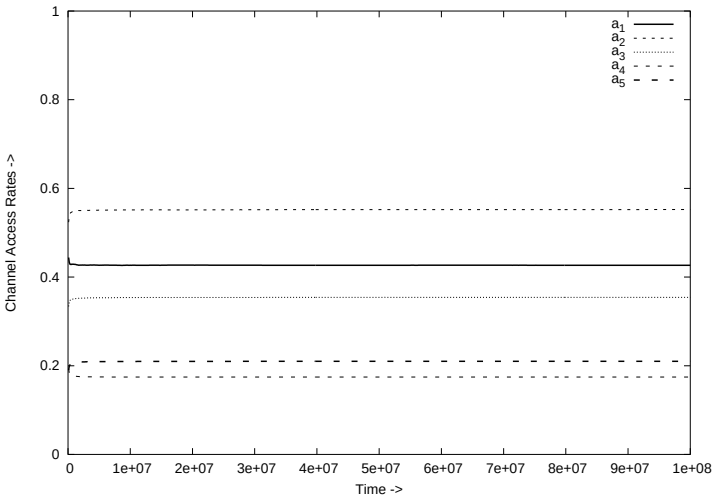

Figure 7: Convergence of Channel Access Rates

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. 7 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 offered to the different nodes.

5 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 where the channel access mechanism is independent of the data sampling process. We also saw that the proposed scheme does not require tedious parameter tuning as is the case for the layered architecture.

We then proposed a learning algorithm, applicable to both the open system 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. Since the objective of the algorithm was only to converge to a Wardrop equilibrium, at this moment it is not able to make a judicious choice among multiple Wardrop

equilibria, if they exist. We are now working on modifications of the algorithm to make it converge to an *efficient* equilibrium.

Appendix

Under *closed system*, the average delay at the transmit queue of node i is
1) When nodes do not have any traffic to forward, $\sum_l \phi_{l,i} = 0$:
Delay at node i = $D_i = \frac{1-s_i}{\alpha_i s_i}$, where

$$s_i = \sum_j \phi_{i,j} (1 - \alpha_j) \prod_{k \in N_j \setminus \{i\}} (1 - \alpha_k) . \tag{8}$$

Proof: if $\sum_l \phi_{l,i} = 0$, then node i has no traffic to be forwarded. The Markov chain of the number of packets in the transmit queue is shown in Fig. 8.

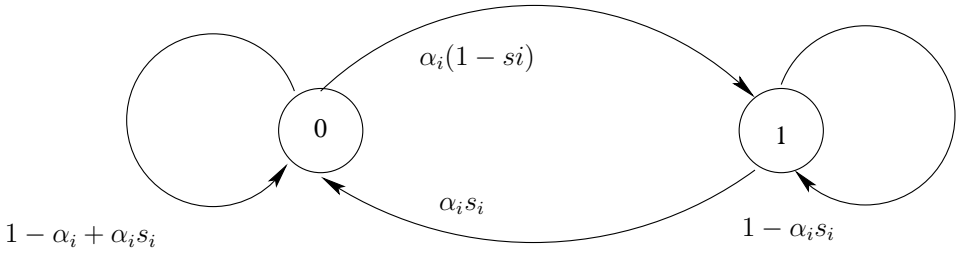

Figure 8: Markov chain for the expected number of packets at node i, case 1: $\sum_l \phi_{l,i} = 0$.

so that, we have the following system of equations

$$\pi_0 (1 - s_i) = \pi_1 s_i \Rightarrow \pi_1 = \frac{1 - s_i}{s_i} \pi_0$$

$$\Rightarrow \pi_0 + \pi_1 = \frac{\pi_0}{s_i} = 1$$

$$\Rightarrow \pi_0 = s_i, \Rightarrow \pi_1 = 1 - s_i$$

Hence expected number of packets in the transmit queue of node i is $1 - s_i$. Using Little's law, the expected delay is

$$D_i = \frac{1 - s_i}{\alpha_i s_i} \tag{9}$$

since the effective arrival rate into node i's queue is $\alpha_i s_i$.

2) The transmit queue of node i can contain more than one packet at a time, $\sum_l \phi_{l,i} > 0$:

$$D_i = \frac{\rho_0}{(1-\rho)(1+\rho_0-\rho)(\psi+\lambda_i)},$$

where

$$\psi_i = (1-\alpha_i)\sum_l \phi_{l,i}\alpha_l \prod_{k\in N_i\setminus\{l\}} (1-\alpha_k). \tag{10}$$

Proof: If $\sum_l \phi_{l,i} > 0$, then the transmit queue of node i can contain more than one packet at a time. The Markov chain of the number of packets in node $i's$ transmit buffer is given in Fig. 9.

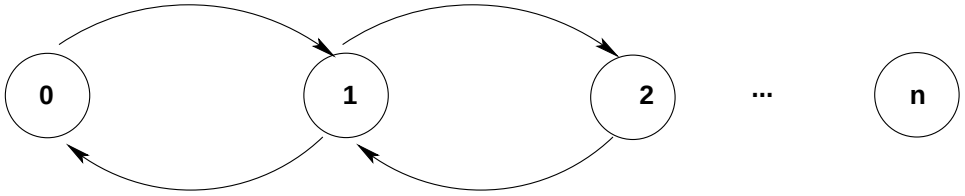

Figure 9: Markov chain for the expected number of packets at node i, case2: $\sum_l \phi_{l,i} > 0$.

where

$$p_{0,1} = \alpha_i(1-s_i) + \psi_i$$

$$p_{n,n+1} = \psi_i,\ for\, n \geq 1$$

$$p_{n,n-1} = \alpha_i s_i,\ for\, n \geq 1$$

we define

$$\rho_0 = \frac{\alpha_i(1-s_i)+\psi_i}{\alpha_i s_i},$$

and $\rho = \frac{\alpha_i s_i}{\psi_i}$. Then $\pi_1 = \rho_0\pi_0$, and $\pi_{n+1} = \rho\pi_n \Rightarrow \rho_0\rho^n\pi_0$ for $n \geq 1$

$$\Rightarrow \pi_0 + \pi_0\sum_{n=1}^{\infty}\rho_0\rho^{n-1} = 1, \Rightarrow \pi_0\left(\frac{1+\rho_0-\rho}{1-\rho}\right) = 1$$

$$\Rightarrow \pi_0 = \frac{1-\rho}{1+\rho_0-\rho},\ \pi_n = \rho_0\rho^{n-1}\pi_0$$

So the expected number of packets in node $i's$ transmit queue is then

$$\sum_{n=1}^{\infty} n\rho_0\rho^{n-1}\pi_0 = \rho_0\pi_0\frac{\partial}{\partial\rho}\sum_{n=1}^{\infty}\rho^n$$

$$= \frac{\rho_0 \pi_0}{(1-\rho)^2}, = \frac{\rho_0}{(1-\rho)(1+\rho_0-\rho)},$$

The expected delay at node $i's$ transmit buffer using Little's law is then

$$\frac{\rho_0}{(1-\rho)(1+\rho_0-\rho)(\psi_i+\lambda_i)} \tag{11}$$

References

[1] V. S. Borkar and P. R. Kumar, "Dynamic Cesaro-Wardrop equilibration in networks," IEEE Transactions on Automatic Control, 48(3):382-296, March 2003.

[2] Jr. R. L. Hamilton and H.-C. Yu, "Optimal routing in multihop packet radio networks," In Proceedings of IEEE Conference on Computer Communications (INFOCOM), San Francisco, pp. 389-396, June 1990.

[3] A. A. Kherani, R. El Azouzi and E. Altman, "Stability-Throughput Trade-off and Routing in Multi-hop Wireless Ad-Hoc Networks," Networking 2006 Conference Proceedings, pp. 25-40, Coimbra, Portugal, may 2006.

[4] J. G. Wardrop. Some theoratical aspects of road traffic research. In Proceedings Inst. Civil Eng., Part 2, 1952, pp. 325-378.

[5] V. S. Borkar. Stochastic approximation with two time scales. System Control Letters, 29:291294, 1996.

[6] 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.

[7] V. S. Borkar and D. Manjunath. Charge based control of diffserv-like queues. Automatica, Vol. 40, No. 12, December 2004.

[8] http://www.tinyos.net/