Data Propagation with Guaranteed Delivery for Mobile Networks

Aslanyan, Hakob; Leone, Pierre; Rolim, Jose

How to cite

ASLANYAN, Hakob, LEONE, Pierre, ROLIM, Jose. Data Propagation with Guaranteed Delivery for Mobile Networks. In: Experimental Algorithms : 9th International Symposium SEA. Naples (Italy). [s.l.] : Springer, 2010. p. 386–397. (Lecture Notes in Computer Science) doi: 10.1007/978-3-642-13193-6_33

This publication URL: https://archive-ouverte.unige.ch//unige:36972
Publication DOI: 10.1007/978-3-642-13193-6_33

Data Propagation with Guaranteed Delivery for Mobile Networks

Hakob Aslanyan, Pierre Leone, and Jose Rolim

Computer Science Department, University of Geneva, Battelle Batiment A, route de Drize 7, 1227 Geneva , Switzerland

Abstract. In this paper, we consider wireless sensor networks where nodes have random and changeable mobility patterns. We study the problem where a particular node, called the base station, collects the data generated by the sensors/nodes. The nodes deliver the data to the base station at the time when they are close enough to the base station to ensure a direct transmission. While the nodes are too far to transmit to the base station, they store the data in a limited capacity internal FIFO queue. In the case where the queue is full, the new generated data are inserted in the queue and the oldest data are lost. In order to ensure, with a high probability, that the base station receives the generated data, the nodes disseminate the generated data in the network. The dissemination process consists in transmitting the data to others mobile nodes which are close enough to ensure a direct transmission. The nodes must control the dissemination process. Indeed, if the nodes send systematically the data to the neighbouring nodes then, the FIFO queues are going to be quickly saturated and the data lost (the dissemination process duplicate the generated data). On the other hand if the nodes do not disseminate the data, the data queued first are prone to be systematically lost if the capacity of the queue is too limited.
We propose a protocol based on the estimate of the delivery probabilities of the data. Each node estimates the delivery probabilities of all the queued data. These probabilities depend on the position of the data in the queue and, on the dissemination process. The lower is the delivery probability the more the nodes disseminate the data to increase the delivery guarantee to the base station. In that way, all the messages get a high probability to be delivered to the base station (higher that some predefined threshold). Experimental validations of the protocol show that the protocol performs well and outperforms an existing protocol. [1]

Keywords: Sensor networks, Mobility, Guaranteed delivery, Data propagation.

1 Introduction

Wireless sensor networks (WSN) are composed of a large number of sensor nodes with sensing, processing and wireless communication capabilities. Usually, the

[1] Partially supported by the ICT Programme of the European Union under contract number ICT-2008-215270 (FRONTS).

nodes are spatially distributed in a given region that they monitor. They use their sensing capabilities to monitor the environment, collecting data like temperature, pressure, vibration, sound, etc. The sensed data (that the nodes generate) have to be delivered to a particular node called the base station.

Usually the nodes are battery powered and it is crucial to limit the energy consumption due to the transmissions in order to increase the operability time of the network (network lifetime). In some settings, the nodes are able to transmit the messages directly to the base station by using wireless transmissions. However, it is known that the energy required to transmit data over distance d is proportional d^α where α is usually in the interval $[2, 4]$ (see [13]). Hence, long-range transmissions are energy-costly. A way to reduce the energy consumption is to use intermediate nodes to convey the data to the base station with multi-hops. There is a large amount of research on energy aware data gathering in wireless sensor networks with static nodes, see for instance [1, 4, 5, 10, 7, 14] and the references therein.

In this paper, we consider the case where the nodes are mobile. Many of the routing protocols for wireless sensor networks with static nodes use information about the network topology. One of the first works on data gathering with mobile WSN is presented in [8] that considers the case where the nodes have a reduced mobility pattern and the base station is mobile. The protocol presented in [15] is similar to the one we present in the present paper: Each node locally calculates a delivery probability and decides whether the data are forwarded. However, it is hard to relate the computed estimate with ours and then to proceed to fair comparison at this stage. In [11] the authors present a data gathering protocol for networks where the position of each node is a known function of time. Finally, in [9] the authors present a protocol for networks with randomly moving nodes with different mobility patterns. The nodes are able to change their mobility patterns during the time. The authors define a mobility level index that captures the node speed, dislocation and mobility changes. Based on this index, the authors suggest to evaluate the probability that a given node will deliver data to the base station.

The main idea of the algorithm presented in this paper is to transmit (diffuse) to many nodes a data m generated by the sensor nodes, in such a way that the probability that at least one of the nodes will get close enough to the base station and delivers the data is larger than a predefined threshold. We consider that the nodes have a limited memory and after getting their memory full, they need to drop data for saving newly generated ones. The nodes manage the memory as a FIFO queue. The main advantage of our algorithm is that the nodes do not use information about their positions; they also do not need to know where the base station is located. The nodes take the decisions whether to forward the data to another node or not by using only the count of the previously (successfully) delivered and dropped data. We proceed to the simulation of the algorithm and, we show the effectiveness of our protocol. We also compare the performance with the algorithm presented in [9]. Both algorithms consider that the nodes are randomly moving with varying mobility patterns and with limited memory.

We compare the performance of both algorithms in terms of data delivery rate, average data delivery delay and average number of sent messages per node.

The rest of the paper is organized as follows. We describe the proposed algorithm in Section 2. The theoretical validation is described in Section 3 while we present the experimental validation in Section 4.

2 Description of the Algorithm

We consider that the memory capacity of the nodes is limited. The nodes convey data to the base station coming from two sources. The first type of data, called generated data, are the data that the nodes acquire with their sensing devices. The second type of data, called received data, are the data that are transmitted by others nodes, i.e. disseminated. Thus, after spending some time away from the base station the generated and received data might saturate the memory of some nodes. In this case, the nodes no longer accept received data. However, the node inserts the generated data in the FIFO queue and the data in the head of the queue are lost.

Let us assume that each node i knows the probability $p_i(m)$ that the data m will be successfully delivered to the base station. We point out that this probability depends on the position in the queue where the data are first inserted. Because the position of the data might change with time, we assume that this information is attached to the data m. Then, $q_i(m) = 1-p_i(m)$ is the probability that the node i will not deliver the data m to the base station. If the data m in some way appear on nodes $j_1, \ldots, j_k$, the probability that at least one of those nodes will deliver m to the base station is

$$P_m = 1 - \prod_{i=1}^{k} q_{j_i}(m). \tag{1}$$

We call (1) the delivery probability of data m, and consequently

$$Q_m = \prod_{i=1}^{k} q_{j_i}(m), \tag{2}$$

is the probability that the data m are not delivered.

The main goal of our algorithm is to diffuse the data in the network in such a way that the delivery probability P_m satisfies $P_m \geq d$ for some predefined threshold d (for example we put $d = 0.993$ in our simulations). Or equivalently, $Q_m \leq 1 - d$.

The discussion above shows that if we are able to compute the delivery probability $p_i(m)$, then the diffusion process ensures that the delivery probability is larger than d.

The nodes manage the incoming data (generated or transmitted) in a FIFO queue. The new accepted data are stored at the end of the queue. Once the

memory is full, a node drops the data from the beginning of the queue to make space for the new generated ones. In this case, no new forthcoming received data are accepted. Data m have a supplementary field where the probability Q_m that the data will not be delivered by the nodes to the base station is stored . Newly generated data before being saved to the node memory have $Q_m = 1$. When node i accepts (generated or received from other nodes) data and stores it in the queue, the probability is updated with $Q_m \cdot q_i(m)$ to take into account the probability that the data will be delivered. Although it is not explicitly denoted, the probability $q_i(m)$ depends on the index where the data are inserted in the queue.

In a second phase, the node i proceeds to the diffusion of the data in the network to ensure that the probability of delivery is large enough. If the new probability that m is not delivered satisfies $Q_m \geq 1-d$ and the node encounters another node j then it transmits the data to j. When node j accepts the data, i marks the data as *diffused* and stops the diffusion process. Node j updates the probability Q_m by $Q_m \cdot q_j(m)$ and stores the data in its queue. Node j diffuses the data further if $Q_m \geq 1-d$.

From the point of view of the node j that is requested to convey the data, the diffusion process consists in: 1. j refuses the data if its queue is full 2. else, accepts the data, updates the probability of delivery and diffuses the data further if $Q_m \geq 1-d$.

Finally when a node meets the base station it forwards all the data from its queue. The node does not remove the delivered data from the queue but, simply keeps them to prevent the multiple acceptance of already delivered data. However, the node inserts new data in the queue as if it was empty by ignoring the already delivered data.

The algorithm that we present in the preceding section uses the probabilities $p_i(m)$ that data m will be delivered to the base station. These probabilities depend on many parameters such as: The index where the data are inserted in the queue, the size of the queue, the mobility pattern and the size of the area covered by the mobile nodes. In order to ensure the flexibility and robustness of the protocol we suggest that the nodes estimate themselves these probabilities. Basically, we propose that the nodes use two counters $C_1[k]$ and $C_2[k]$ per queue's entry k. The counter $C_1[k]$ counts the number of data inserted in position k that are delivered by the node to the base station (the value of these counters depend with the time t but we do not introduce this dependency in order to simplify the notation). The counter $C_2[k]$ counts the total number of data inserted in the queue at position k. We then suggest to estimate the probability of delivery with the estimation

$$p_i(m) \approx \frac{C_1[k]}{C_2[k]}. \tag{3}$$

On the left side of Figure 1, we can observe the time the nodes needs to estimate a suitable delivery probability. We observe that the nodes improve the estimates in order to provide a nearly 100% data delivery rate.

Although the purpose of this paper is to provide evidence that the protocol is suitable and a full theoretical analysis of the performance is beyond the scope of this paper, we provide some theoretical evidence here.

3 Theoretical Analysis of the Performance of the Algorithm

We first notice that the mobility pattern of a node is independent of the data generated and received. The random mobility patterns that we consider are proposed in [9] in a similar setting that ours. In [3] the authors classify such random mobility patterns as Random Direction Mobility Patterns since the nodes choose a direction and a travel time repetitively. The aim of such mobility patterns is to make the distribution of the nodes as even as possible in the covered area as well as to ensure that the encounters between mobile nodes are as constant as possible. Indeed, the mean number of neighboring nodes is rather constant, compared, for instance, to the Random Waypoint Mobility Model, see [3].

We use this property and assume that for a given node the expected number of received data on a time span T is μT. Alternatively, we may define $\mu = \lim_{t\to\infty} E(\#received\ data\ in\ [0,t])/t$, and prove that the limit exists by using the stationarity of the nodes' motion. In particular it is independent of the position of the nodes.

The expected number of generated data is also assumed to evolve linearly with time and we denote λT this number. This corresponds to the situation where the nodes collect data repetitively in a deterministic way or in a random but stationary way.

In the following, we discuss how to prove that the estimates (3) converge (see equation (5)). In order to ensure the convergence, it is necessary that that the number of data generated and received by a node, denoted M_n, during the travel time does not depend on the time (time homogeneous) [2]. Although this has to be proved formally, the discussion above shows that this is a reasonable assumption.

Instead of considering the probability $p_i(m)$ that a node i delivers the data m, we consider an averaged value p. This is equivalent to consider the complete set of data and compute the average probability of delivery. Equivalently, we replace the value $p_i(m)$ by an average value p given that the nodes' encounters as well as the index where the data are queued are random. Given the probability p, data m are diffused an expected number α times, ensuring that $(1-p)^{\alpha+1} < 1-d$ ($\alpha = \alpha(p) = log(1-d)/log(1-p) - 1$). Notice that we implicitly assume that the probabilities of delivery are independent of the nodes and the index of the queue.

Each time a node transmits the data to the base station, the node updates the probability of delivery using (3). We denote by p_n the n-th estimate. Let us denote by N_n the total number of data received by the node at the time step n. N_n is the value of C_2 in (3) where we removed the dependencies in k and in time. By the definition of p_n ($p_n = C_1/C_2$), C_1 is then given by $N_n p_n$. Between

the time steps n and $n+1$ the node travels during a time T and has to convey M_n generated and received data. Then, the new estimate p_{n+1} is given by

$$p_{n+1} = \frac{1}{N_n + M_n}\Big(N_n p_n + (M \wedge M_n)\Big)^2.$$

The value M_n is the total number of data collected by the node between the time steps n and $n+1$, $N_n p_n$ is the expected total number of data delivered at time step n and $M \wedge M_n$ is the total number of data delivered to the base station at the time step $n+1$.

After some algebraic manipulations using that $N_n \to \infty$ we get that

$$p_{n+1} \approx p_n - \frac{1}{N_n}\Big(M_n p_n - (M \wedge M_n)\Big). \tag{4}$$

This last equation shows how the estimate of $p = \lim_{n\to\infty} p_n$ evolves. Consider first that the queue is never full, i.e. $M \wedge M_n = M_n,\ \forall n$. In this case, p_n increases up to the time when $p_n = 1$. The behaviour is correct since no data are lost and then, the probability of delivery is 1. On the other case, if some data are lost, i.e. $M \wedge M_n = M$, p_n might converge towards the value ensuring that $p_{n+1} = p_n$. One can prove that under some mild assumptions, we have

$$p = \lim_{n\to\infty} p_n = \frac{E\big(M \wedge M_n\big)}{E\big(M_n\big)}. \tag{5}$$

This last equation shows that the estimates p_n are converging to the right limit since the right side of the equation is the fraction of data that are delivered to the total number of data received and generated by the node.

The rate of convergence of the estimate (3) is difficult to compute. However, we observe on the left of Figure 1 that after a simulation period of $50'000$ seconds the estimate are good enough to provide nearly the maximum delivery guarantee. In the conditions of the simulations, this is the time corresponding to going back to the base station 25 times in average.

With the definition of the parameters λ and α provided in the beginning of the section, we obtain that $E(M_n) = T(\lambda + \mu)$, with T is the expected travel time. Notice that we depart from our implementation of the algorithm since the expression for $E(M_n)$ given here counts the received data even if the queue is full. Using (5), we observe that once the convergence occurs, the nodes can estimate the value of T by using only the statistics related to the queue occupation, λ and μ with

$$T = \frac{E\big(M \wedge M_n\big)}{p(\lambda + \mu)}. \tag{6}$$

Notice that we also expect that $\lambda + \mu = \lambda(1+\alpha)$ since the expected total number of data diffused in the network is $\alpha\lambda$, these data have to be carried by the nodes

[2] $a \wedge b$ is the minimum of a and b, recall that M is the capacity of the queue.

and we assume that the diffusion process distribute the data uniformly among the nodes.

The analysis we have proposed is likely to be rooted in some formal framework. Indeed, we postulate the existence of the constant λ, μ and, we conjecture that this existence can be asserted by using a renewal argument [6]. The renewal theory framework is natural in our setting, since each time a node gets by the base station corresponds to a renewal. In our short investigations, we assume that the estimate p_n is more or less similar for all nodes since we define α as a function of p_n. However, from the numerical experiments that we conducted, it appears that the probability of delivery depends on the mobility pattern. The 'mean-field' analysis that we propose here is prone to be more accurate in the case where the mobility patterns of all the nodes are the same. On the other case, the value we obtain is an averaged value. Moreover, the analysis of the convergence of the estimate (3) can be conducted with the ODE method [12, 2].

We point out that the estimate (6) counts the data received after the queue is full. In our implementation of the algorithm we do not accept data from others nodes while the queue is full, only generated data are inserted in the queue.

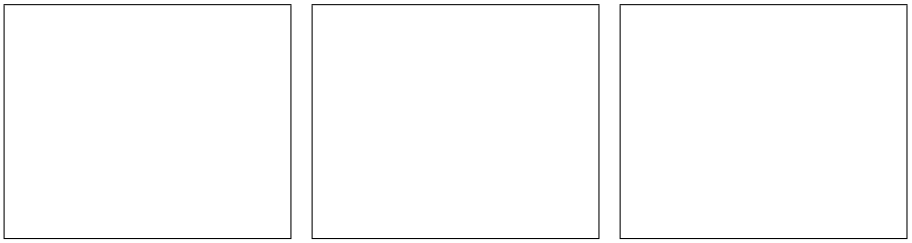

Fig. 1. Time evolution of the algorithm performance with the complex mobility pattern and 20, 80, 300 and 400 nodes. From left to right: The data delivery probability, the average message delay and the average number of messages sent per node. Axis X is a time scale in 1000 seconds.

In the next section we validate experimentally our protocol where nodes use (3) for the estimation of the delivery probability.

4 Experimental Validations

In Figure 1 we present the time evolution of some parameters. Under our simulation conditions the average travel time is about 2000 seconds (not very dependent on the simple or complex mobility pattern). The figure on the left shows how the data delivery rate evolves. We observe that after a period of $50'000$ seconds, the behaviour stabilizes and the network delivers the data with the required guarantees. In the simulation conditions, the period of $50'000$ seconds corresponds to going back an average of 25 times. This means that by applying formula (3) 25 times, we obtain some estimates that are sufficiently accurate to ensure the

delivery guarantees. The figure on the center shows the average delay. We observe that the delay is much lower, about 1000 seconds. This means that the diffusion process is really participating efficiently to convey the data to the base station. On the right of Figure 1 we display the average total number of data sent by node. We first observe that this increases linearly with time. This supports our assumption that the network dynamics is stationary. Moreover, the $40'000$ transmissions are due to an average of $10'000$ generated data and the diffusion of an average of $15'000$ data. Because the average travel time is about 2000 seconds, each node delivers about 125 data to the base station. This shows that on average the network can support the load of conveying the generated data and, that the network is also able to adapt to the period where more data are produced than in average. We suspect that the conditions of the simulations are close to the limit, in the sense that increasing the rate of generated data might lead to a decrease in the data delivery rate.

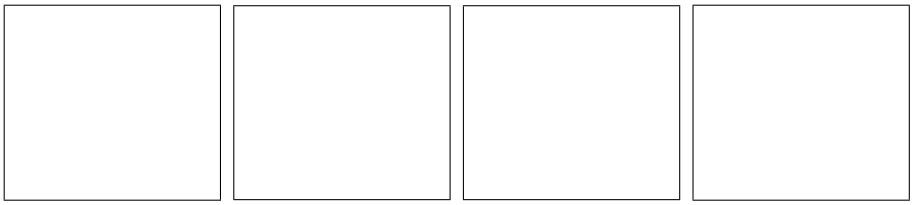

Fig. 2. Network connectivity at random time. Left to right 20 nodes, 80 nodes, 300 nodes, 400 nodes.

We proceed to a set of simulations with different network configurations in order to evaluate the performance of the protocol that we propose in this article. We also proceed to a set of simulations with the same set parameters to the mobility level based protocol (local adapt with random neighbor selection) presented in [9] in order to compare the performance. Actually, both protocols are comparable since they consider nodes with random motions and limited memories. To simplify the presentation, we use the well defined mobility patterns introduced in [9]. The four simple mobility patterns defined are *Working Mobility*, *Walking Mobility*, *Biking Mobility* and *Vehicular Mobility*. These mobility patterns are similar to the motion of a human who is working in his office, walking outside, biking or driving. Figure 3 presents some traces of the motions of the above-defined simple mobility patterns. For each mobility pattern, a node selects a direction and a speed and moves a random time in the direction at the given speed. Basically, the speeds range makes the difference between the various mobility patterns. Using these simple mobility patterns, we define more complex ones where a node changes its mobility pattern by passing from one pattern to another one with some probability. These complex mobility patterns are easy to present with the transition graphs on the Figures 4 and 5. Each vertex of a graph corresponds to a simple mobility pattern, and two vertices are

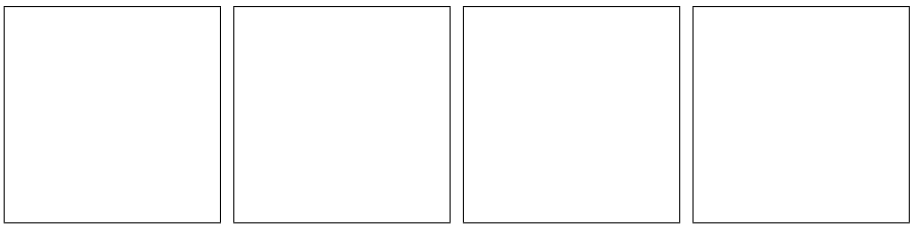

Fig. 3. Motions of simple mobility patterns. Left to right *Working Mobility*, *Walking Mobility*, *Biking Mobility*, *Vehicular Mobility*.

connected by a directed edge, on the top of which is written the probability of passing from on pattern to the other one. $C1 - C4$ complex mobility patterns, presented on Figures 4 and 5 are similar to the ones in [9] and detailed information about the mobility patterns can be found therein. In our simulations all the nodes have the same size of memory, which is enough to accommodate 128 messages, the transmission range of nodes and base station is $70m$ and network is a $1000 \times 1000m^2$ square. Also, each node in average generates one message per 40 seconds ($0.025msg/second$).

We present two sets simulations. In the first, we assign each simple mobility pattern to 1/4 of total number of nodes in network. In the second round, we use the complex mobility patterns in the same portions. Every round contains four simulations with different numbers of nodes in networks 20, 80, 300 and 400, in total eight different simulations. In Figure 2 we show the network connectivity at random time for different numbers of nodes. For each of eight simulations, we simulate the protocol for 400.000 seconds (111 hours) and, we chose the required delivery rate $d = 0.993$ and use (3) for the estimation of the delivery probability.

Fig. 4. Transition graphs of complex mobility patterns used in our simulations (in M_{stop} state node has no motion). On the left *C1 Mobility* on the right *C2 Mobility*.

We compare the performance of our algorithm with the one presented in [9]. We consider three criteria. The first is the data delivery rate, which is the percent of delivered data to the base station. The second is average data delivery

Fig. 5. Transition graphs of complex mobility patterns used in our simulations (in M_{stop} state node has no motion). On the left *C3 Mobility* on the right *C4 Mobility*.

delay which is the average time the data are delivered to base station after being generated. And as the main part of energy goes for data transmissions, we compare the average number of sent data per node which will roughly represent the energy consumption of the protocols. In Figure 6 the delivery rate comparison of protocols are presented, respectively for networks where nodes have simple and complex mobility patterns. We observe that both protocols have stable delivery rates, according to number of nodes in network. And in all eight cases our protocol ensures the requested delivery rates. The comparisons of the average

Fig. 6. Data delivery rate comparison. On the left *simple mobility* on the right *complex mobility*.

data delivery delay of both protocols are presented in Figure 7. Here we observe that as the number of nodes composing the network increases, the data delivery delay tends to a constant. The delivery delay decreases as the number of nodes increases. The algorithm proposed in [9] behaves similarly and the delivery delay is shorter for this algorithm than for ours. It is the only criterion for which that happens.

Figure 8 presents the protocol comparisons in terms of average sent data per node, we observe that the success of mobility level based protocol in dense networks in terms of average message delivery delay is due to the high number of sent data (replication). However, this requires a larger amount of energy. We

Fig. 7. Average data delivery delay comparison. On the left *simple mobility* on the right *complex mobility*.

observe that our algorithm ensures that the number of data sent does not increase as the number of nodes becomes large. Indeed, we observe that the number of data sent tends to a constant. We observe that if the number of nodes in the

Fig. 8. Average sent data per node comparison. On the left *simple mobility* on the right *complex mobility*.

network is not large enough, the diffusion process does not manage to provide the guaranteed delivery of data. This is due to the fact that the nodes do not encounter others nodes. However, our experimental validations show that with 20 nodes, see left of Figure 2, we do not manage to ensure that $Q_m < 1-d$. However, the performance are still valuable since the algorithm ensures 97.84% and 96.82% of data delivery for respectively simple and complex mobility patterns. Figure 8 shows that with 20 nodes the number of data sent is small and, this confirms that the nodes' encounters are not sufficiently frequent. This has also an impact on the data delivery delay that is larger than for networks with more nodes.

References

1. Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks. *IEEE Wireless Communications*, 11(6):6–28, 2004.

2. V.S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.
3. T. Camp, J. Boleng, and V. Davies. A survey of mobility models for for ad hoc network research. *Wireless communications & Mobile Computing*, 2(3):531–541, 2002.
4. Joseph C. Dagher, Michael W. Marcellin, and Mark A. Neifeld. A theory for maximizing the lifetime of sensor networks. *IEEE Transactions on Communications*, 55(2):323–332, 2007.
5. Arvind Giridhar and P. R. Kumar. Maximizing the functional lifetime of sensor networks. In *IPSN*, pages 5–12, 2005.
6. D.P. Heyman and M.J. Sobel. *Stochastic Models in Operations Research, volume I*. Dover publications, Inc., 1982.
7. Aubin Jarry, Pierre Leone, Olivier Powell, and José D. P. Rolim. An optimal data propagation algorithm for maximizing the lifespan of sensor networks. In *DCOSS*, pages 405–421, 2006.
8. P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experience with zebranet. *10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLO02)*, 2002.
9. Athanasios Kinalis and Sotiris Nikoletseas. Adaptive redundancy for data propagation exploiting dynamic sensory mobility. *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 149–156, 2008.
10. Pierre Leone, Sotiris E. Nikoletseas, and José D. P. Rolim. An adaptive blind algorithm for energy balanced data propagation in wireless sensors networks. In *DCOSS*, pages 35–48, 2005.
11. C. Lui and J. Wu. Scalable routing in delay tolerant networks. *Mobihoc*, 2007.
12. S. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2008.
13. K. Pahlavan and A. Levesque. *Wireless Information Networks*. John Wiley and Sons, 1995.
14. Olivier Powell, Pierre Leone, and José D. P. Rolim. Energy optimal data propagation in wireless sensor networks. *J. Parallel Distrib. Comput.*, 67(3):302–317, 2007.
15. Yu Wang and Hongyi Wu. Dft-msn: The delay/fault-tolerant mobile sensor network for pervasive information gathering. *INFOCOM*, 2006.