EDGM: Energy Efficient Data Gathering with Data Mules in Wireless Sensor Networks

Nour Brinis
CRISTAL laboratory
National School of Computer Science
University of Manouba, 2010 Tunisia
brinis.nour@gmail.com

Pascale Minet
INRIA Rocquencourt
78153 Le Chesnay cedex, France
pascale.minet@inria.fr

Leila Azouz Saidane
CRISTAL laboratory
National School of Computer Science
University of Manouba, 2010 Tunisia
leila.saidane@ensi.rnu.tn

***Abstract*—This paper deals with prolonging the lifetime of a wireless sensor network by exploiting mobility. We consider data gathering applications where each event occurring in the monitored area must be reported to the sink. The target applications are assumed to be delay tolerant. We define the network lifetime as the time until coverage loss. We propose an energy efficient strategy to collect data from sensor nodes, using data mules. This strategy has the advantage of not requiring network connectivity. Data mules are responsible to carry data to the sink, to schedule sensor nodes activities and to replace energy constrained nodes. They take advantage of energy harvesting, from a generating power terminal, in order to achieve their mission. We simulate the proposed data collection strategy with the NS2 simulator and derive the network lifetime, the energy consumption of sensor nodes and the rate of collected data meeting a given delay. Furthermore, this study leads us to determine the optimal number of data mules needed to meet a given delivery delay deadline.**

Keywords-wireless sensor networks; energy efficiency; data mule; node activity scheduling; delay tolerant network

I. CONTEXT AND MOTIVATION

Wireless Sensor Networks (WSNs) play an important role in different applications, such as environment monitoring, health treatment, space exploration and others [1]. A WSN is composed of a large number of sensor nodes. Each one is characterized by three basic capabilities: sensing, wireless communication and computation. These sensor nodes are usually battery-operated. So, their power must be used very sparingly in order to fulfill the underlined application requirements before batteries depletion. Furthermore, each sensor application requires specific quality of service: delivery delay deadline, data accuracy, etc. Hence, minimizing energy consumption, while meeting required performance constraints of the target application, is a very important challenge to prolong network lifetime.

Sensor nodes are usually densely deployed in order to ensure the total connectivity and coverage of the interest area. However, connectivity may be lost quickly because some nodes, such that located in the proximity of the sink, consume more energy for relaying traffic originated from other farther nodes. Also, the variability of the sensors distance to the sink involves a variability of the data delivery delay. Thus, delivering data to the sink by multi-hop forwardings may lead to connectivity loss in addition to an unfair data delivery delay. Moreover, a complete coverage of an area implies connectivity among the nodes only if the radio range is at least twice of the sensing range [2]. However, if the radio range is too large as compared to the sensing range, the network may be subject to excessive radio interference although its connectivity is ensured [2]. That is why, solutions that do not require connectivity must be investigated in order to preserve data delivery to the sink.

In this paper, we focus on the problem of delivering data to the sink while conserving sensor nodes energy, in a WSN totally covered and not necessarily fully connected. Our proposed strategy is based on the use of data mules for gathering data from sensor nodes that are assumed to be static. A data mule is a kind of mobile robot able to communicate with sensor nodes and to carry the generated traffic to the sink. Our solution targets especially delay tolerant applications, requiring a long network lifetime. We define the network lifetime as the time of the first coverage hole creation. we propose a gathering strategy that extends clearly the network lifetime. Indeed, the data mule arrivals are predictable. So, static sensor nodes wake up at well-defined periods, without need to deplete energy for listening data mules polling messages. In addition, static sensor nodes transmit data only at single hop ranges. Our solution combines the data gathering with a maintaining coverage strategy. For this purpose, constrained energy sensor nodes may be replaced by other redundant nodes in order to maintain the whole network coverage.

The performance evaluation of our solution allows us to obtain the rate of delivered data meeting a given delivery delay and the required number of data mules that guarantee this deadline.

The paper is organized as follows. In Section II, we briefly present the state of the art related to data gathering strategies. Afterwards, we define our data collection scheme by describing sensor nodes and data mules behaviors in Section III. In Section IV, we evaluate the proposed solution and compare it with a double range data gathering strategy in terms of energy consumption and data delivery delay. Finally, we conclude the paper and give some directions for our future works in Section V.

II. RELATED WORK

Exploiting mobility for data gathering and routing has been studied in many papers, using various methods. These solutions aim at extending the network lifetime by conserving sensor nodes energy. They can be classified according to the network mobility degree into Networks with mobile sensors and Networks with additional mobile agents.

In mobile WSNs, where all sensor nodes have motion capabilities, the connectivity between mobile nodes is poor, and thus it is difficult to guarantee an end-to-end connections from sensor nodes to the sinks. The main issue is to detect the occasional connectivity between mobile nodes in order to transmit sensed data to the sink. This problem has been treated in ZebraNet [3] project. This project relies on equipping zebras with sensor nodes in order to record the animal position or other relevant sensor readings. For this purpose, zebras wander randomly in the area and send data to the sink when they arrive to its transmission range. Due to their random movement, delay latency can not be bounded and in worst case data can never been send to the sink when the tagged zebra remains far away. So, in order to reduce the data delivery delay, a history-based routing approach has been proposed in [4]. Each sensor node records its past success rate of directly transmitting data packets to the sink. When a sensor meets another sensor, the former transmits data packets to the latter if the latter has a higher success rate.

In static WSNs, having mobile sinks, sinks move towards static sensor nodes in order to collect sensed data. This approach was studied in [5][6]. Otherwise, Additional agents move around static sensor nodes to gather data and carry them to static sinks. In both cases, mobile nodes should gather data via short range communication in order to minimize the energy consumption, while assuring a fairly data delivery to the sink. The data gathering strategies can be classified into: direct-contact data collection and cluster based data collection.

Many applications, especially these deployed for statistics measurement, are more sensitive in terms of network lifetime than that in terms of delivery delay: they are delay tolerant. For this kind of applications, direct-contact data collection is more suitable. The problem has been dealt with in [7] using animal-based mobile collectors. In that case, mobile collectors wander randomly in the interest area to gather data from static sensor nodes. Energy consumption at sensor side is only due to mobile collector discovery and subsequent data transmission. Assume that a mobile collector periodically broadcasts a beacon message while moving. It is very expensive to detect the mobile collector arrival by keeping listening to the beacon message. Moreover, because of the random movement of mobile collectors, the static sensor nodes are not ensured to deliver their sensed data within a bounded delay. Studies in [8] show that, if the mobile collectors move along regular trajectory, then sensors can predict their arrival and so, the network lifetime is improved. One of the regular gathering trajectories that guarantee a single hop data gathering with extreme energy saving is that the mobile collector gathers data packets by sequentially visiting each sensor. This problem is referred in sparse sensor networks to the Travelling salesman problem TSP. In order to shorter the tour length while conserving a single hop transmission, data collectors visit the transmission range of each node in order to gather data. For this purpose, an alternative solution of TSP called TSP neighboring was been proposed in [9]. The above mentioned strategies reduce greatly the energy consumption for data transmissions, however, the data latency increases because of the tour length.

In order to shorten the data delivery latency, most of the previous work relay on introducing multi-hop relays via clustering definition. These strategies aim at retrieving a trade off between network lifetime and delivery time. In these approaches, a mobile collector moves around a subset of sensors, cluster heads, to gather data by local multi-hop communication. By this way, the tour length is shortened and thus the data delivery latency decreases. the main encountered issue is the selection of the appropriate clusters according to a given criteria. Studies in [10] propose a cluster definition according to the definition of the connectivity islands (partitions) of the network. Then, the cluster head is chosen in the middle of the partition and having the maximum residual energy. Studies in [11] presents a bounded relay hop data gathering strategy. An algorithm, called SPT-DCA, selects appropriate nodes that will aggregate captured data transfered by bounded hop transmissions. then, the mobile collector visits the selected nodes to gather data, using the TSP heuristic. SPT-DCA is based on a prior Spanning Tree resolution of the whole network graph. Then, the appropriate cluster head nodes are selected according to a fixed relay hop bound.

Most of the previous data gathering studies deal with extending the network lifetime in a sparsely sensor network. They assume that all the sensors are working simultaneously during the whole process of data collection. Thus, the coverage problem was rarely addressed in the data gathering solutions. Two issues are considered, in this paper, in order to improve the network lifetime: a single hop gathering strategy using data mules, combined with a mechanism for the replacement of some critical nodes before depleting their energy. The goal of our solution is to determine the optimal number of data mules required to extend the covered network lifetime while meeting a given data delivery delay deadline. In the following sections, we detail our proposal.

III. EDGM: DATA COLLECTION SCHEME

Our solution, called EDGM for Energy efficient Data Gathering with data Mules, is characterized by:

- All sensor nodes in the network are assumed to be static. They are denoted *SN*.
- Each sensor node has two independent components: sensing and communication units. Both units are powered from the same limited source of power (battery).
- The data mule or mobile node, denoted *DM*, is able to recharge itself at a generating energy terminal, called energy terminal. We assume that this energy terminal is located at the proximity of the sink.
- Each sensor node and each *DM* are able to compute their residual energy.
- Each sensor node and each *DM* have a memory sufficient to store the collected data.
- The transmission range of each sensor node is denoted r_c. We suppose that all nodes have the same communication range. Otherwise, r_c denodes the minimal transmission range among all deployed nodes.
- The transmission range of each data mule is denoted *R*. The constraint that must be met by *R* is given by Inequation 1 in Section III-C1.
- For simplification purpose, we assume that recharging a *DM* is sufficient to allow it to move, communicate and collect data before turning back to the energy terminal.

A. Principle of the solution

We now explain how each mobile node collects the sensed data. The goal of each sensor node is restricted to capture data and transmit them to the sink, if this node is located in its transmission range, or to a *DM*, when it is one hop away. In other words, sensor nodes do not relay the traffic generated by other nodes. Two main issues must be solved: the trajectory of the *DMs* and the behavior of the sensor nodes. At the proximity of each static node, a *DM* stops at a Break Point, denoted *BPoint*, for a defined Break Period, denoted *BPeriod*, and sends a *HELLO* message. When it arrives at the proximity of the sink and the energy terminal, a *DM* sends the collected data to the sink and recharges itself. Each sensor node *SN* is associated to a unique *BPoint*. The break point positions are defined in Section 3.3. Each *SN* is awake only during its break period to send its captured data. Then, it turns off its communication unit. During the break period, since several sensor nodes are authorized to send data to the *DM*, collisions may occur. A deterministic medium access scheduling is set by the *DM*, in order to avoid the collision problem. More precisely, each node starts sending its sensed data when it receives an invitation delivered by a *DM* through an *INVIT* message which contains the medium access scheduling. Figure 1 summarizes the data collection strategy. During the first round, each *SN* receives the information needed to synchronize itself with the *DMs* present in the considered area. Once it is synchronized, it keeps awake only during a *DM* break period in its associated *BPoint*. During this time, it is authorized to send its collected data through an *INVIT* message. Detailed presentation of the sensor nodes and the *DM* behaviors is given in Section III-B and III-C respectively.

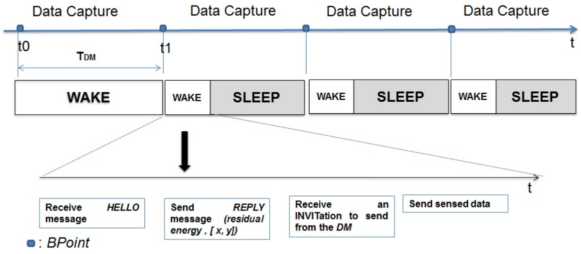

Figure 1. Sensor activities.

B. Sensor Node Behavior

The energy consumed by a node depends on its state. We distinguish the four following states, presented in ascending order of energy consumption:

- *Off*: all sensor units are turned off,
- *Sleep*: only the sensing unit remains active,
- *Receive*: all sensor units are active and the communication unit is receiving a message,
- *Transmit*: all sensor units are active and the communication unit is transmitting a message.

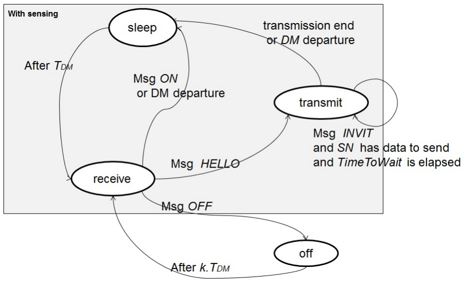

Figure 2. Sensor activities scheduling model.

During the first round of each *DM*, each sensor node remains awake in order to memorize the *DM* round duration, denoted T_{DM}. T_{DM} represents the period between the first and the second received *HELLO* message from the same DM (see for instance t_1 - t_0 in Figure 1). Any sensor node, that is not redundant, will be awake, during a break period, starting when the *DM* stops at its associated break point. Each T_{DM} period, *SN* switches to the *receive* state and processes the message received from *DM*, as described in Figure 2:

- If an *ON* message is received: *SN* switches to the *sleep* state and will switch to the *receive* state for the next arrival of *DM*.

- If an *OFF* message is received: *SN* switches to the *off* state for the *k* following rounds, with *k* a positive integer.
- If a *HELLO* message is received: *SN* switches to the *transmit* state and sends a *REPLY* message.
- If an *INVIT* message is received: *SN* analyses the received message in order to determine the medium access scheduling. It waits its turn, for a *TimeToWait* period, before beginning its data transmission. When this time is elapsed, *SN* starts sending data for a maximum predefined period (*PeriodToSend*). The *TimeToWait* and *PeriodToSend* values are included in the *INVIT* message.

Once *SN* is in the *transmit* state and after a random period, it sends a REPLY message in which it specifies its residual energy and its location. Since each *SN* is static, it sends its location information only in the first round. After that and in all *DMs* rounds, it waits for an *INVIT* message to send its data. It switches to *sleep* state at the *DM* departure or when it ends its data transmission.

Once *SN* is in the *sleep* state, it wakes up after T_{DM} and it switches to the *receive* state.

From the *off* state, a redundant *SN* switches to the *receive* state after $k.T_{DM}$ time units.

C. Data Mule Behavior

In this section, we describe the *DM* behavior. We first define the *DM* trajectory. After that, we present the *DM* behavior at each break point, and the communication between *DMs*. Notice that all *DMs* follow the same trajectory, but with different starting points.

1) DM trajectory : Three conditions must be satisfied for the choice of the *DM* trajectory:

- C1: all static nodes are explored;
- C2: the trajectory length, before turning back to the sink, is minimized;
- C3: the number of break points is minimized.

This problem is reduced to a 2-dimension geometry problem: how to tile a rectangular area *(X,Y)* with a minimum number of disks of radius r_c where each disk center represents a break point of *DMs*? This problem has been solved in [12]. The optimal disks positions are such that the tops of equilateral triangles of edge length $\sqrt{3}.r_c$, are the disk centers. By this way, we ensure the exploration of all static sensor nodes by visiting a break point in their transmission range (Conditions C1 and C3 are satisfied). So, knowing the r_c value, each DM computes the next breakpoint location. In order to minimize the delivery delay needed to transfer gathered data to the sink node, *DM* follows the trajectory visiting first only the break points at even lines, and then that at odd lines, as illustrated in Figure 3 (Condition C2 is satisfied). Besides, when a *DM* meets another *DM*, by receiving its *HELLO* message, the *DM* that moves towards the sink collects all data gathered by the other *DM*. By this way, we reduce the data delivery delay. For this purpose, we use the following assumption:

$$R \geq \sqrt{3}.r_c \tag{1}$$

The first location of *DMs* are chosen to ensure fairness between all sensor nodes in terms of delivery delay. So, the initial positions of the *DMs* are chosen as follows:

Let *n* be the number of *DMs* in the rectangular area *(X,Y)*, if *n* is even, two *DMs* are set each $\frac{2.Y}{n}$. Otherwise, we set two *DMs* each $\frac{2.Y}{n+1}$. These two *DMs* will move in opposite directions. The remaining *DMs* are set at the first *BPoint* of the extremity lines of the trajectory.

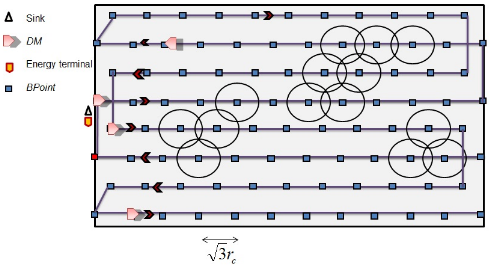

Figure 3. *DM* trajectory.

2) DM behavior at a break point: Stopping at a specified *BPoint*, a *DM* acts as the leader for all the sensors associated to this *BPoint*. When it receives the *REPLY* message from different sensor nodes, it decides whether the sender node is redundant or not, it replaces the not redundant node having critical energy level and it schedules the transmission of the active not redundant nodes. With the knowledge of the *SNs* location sent in the *REPLY* message, *DM* decides whether the sender node is redundant through the algorithm presented in [13]: knowing the positions of all active neighboring nodes, *DM* examines if any semicircle within the interest area of the sender node is empty. In that case, the sender node is not redundant because a portion of its area is covered only by it. In addition, when the residual energy level incorporated in the *REPLY* message is critical, the *DM* sends an *ON* message to the neighboring nodes of the sender node one by one, and each time it checks if the energy constrained node become redundant. In that case, it stops the replacement algorithm; the sender node is replaced. Moreover, using the residual energy sent in each *REPLY* message, *DM* specifies, in the *INVIT* message, the time for each node to wait before starting transmission (*TimeToWait*) and the transmission duration (*PeriodToSend*). Sensors having less residual energy are invited to transmit first. Knowing the number of sensor nodes associated with this *BPoint*, *DM* assigns a fair duration to each node for sending its sensed data.

Table I. Simulation parameters.

Network configuration	Network area	(300m X 300m)
	Number of data mules	1 to 4
	Range (*SN*)	50m
	Range (*DM*)	100m
	Range (sink)	50m
	Bandwidth	2Mbps
	Mac protocol	802.11
	DM(s) speed	1m/s
	BPeriod	15s
	k	3
Traffic parameters	Type	Periodic Data
	Packet size	256 bits
	Throughput	16 Kbps
	DM(s) Buffer size	100 packets
Energy model	Type	Battery
	Initial Energy(*SN*)	15 Joules
	Transmit power	0.36 Watt
	Receive power	0.24 Watt
	Sensing power	0.015 Watt

IV. PERFORMANCE EVALUATION

In this section, we present simulation results to evaluate the efficiency of the proposed EDGM strategy. We carried out several simulations using NS2 simulator and selected modules of the pre-implemented MannaSim project [14]. MannaSim extends the NS2 simulator by introducing WSN specifities: battery energy resource, data generator framework...

A. Evaluation metrics

Our simulation analysis emphasizes on the following performance metrics:

- The energy consumption rate of a node *i*, denoted *ConsumedEnergy*$_i$, is defined by (2) where *InitEnergy*$_i$ denotes the initial energy of node *i* and *Energy*$_i$ is the residual energy of node *i*.

$$ConsumedEnergy_i = \frac{InitEnergy_i - Energy_i}{InitEnergy_i} * 100 \qquad (2)$$

- The average data delivery delay, denoted *AvgDeliveryDelay*, is defined as the sum of packet delivery delays divided by the number of received packets (see (3)):

$$AvgDeliveryDelay = \frac{Sum\_of\_all\_Pkt\_Delays}{Number\_of\_Received\_Pkts} \qquad (3)$$

- The rate of packets, received before a given delay deadline, among the total number of received packets, denoted *MeetDDDRate*, is defined by (4). The Data Delivery Deadline (*DDD*) is the duration elapsed from data generation to data reception by the sink:

$$MeetDDDRate = \frac{Received\_Pkts\_Meeting\_DDD}{Total\_Received\_Pkts} \qquad (4)$$

B. Simulation results

Table I summarizes the simulation scenarii parameters.

We suppose that each node has a transmission range equal to its sensing range, denoted *Range* in Table I.

Firstly, we consider a topology, where 2 *DMs* are used, 16 sensor nodes are deployed to ensure coverage, and 134 redundant sensor nodes are deployed randomly. Notice that the node having identifier 0 is the sink and the nodes 1 and 2 are the *DMs*. Nodes 3 to 18 constitute the 16 nodes needed to ensure coverage. Nodes, having an identifier strictly higher than 18, are initially redundant. Figure 4 visualizes the useful and the total energy consumption rate versus the sensor node identifiers. The useful energy refers to the amount of energy really required to capture and transmit the monitored events. We suppose that the near death energy is 2 Joules. So, all nodes consuming more than 86% of their energy tend to be out of service and the appropriate nodes have been selected to cover the coverage hole before its creation. We observe that energy consumption is clearly minimized for redundant nodes that are mostly kept in off state, starting from the second round. Some nodes, such as 31, 38, 54 and 69, waste more energy than other initially redundant nodes because they have been selected to replace energy constrained nodes. We also notice that the gap between the useful and the total energy is not too significant, due to the frequent turning off of the radio component of each node.

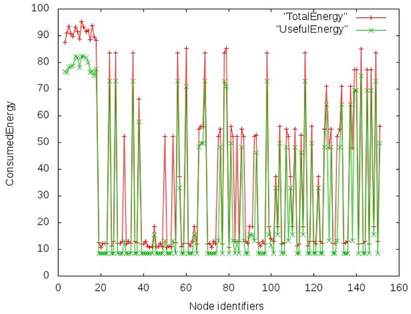

Figure 4. energy consumption rate.

Figure 5 depicts the network lifetime when the total number of nodes varies from 20 to 150 to represent different node density. For each new measurement, we add randomly the appropriate number of nodes on those already deployed. We notice that less *DMs* used leads to longer network lifetime since static sensors can turn frequently to powe-saving mode. It was also shown that more redundant nodes are deployed, latter is the first coverage hole creation. This is reasonable because more redundant sensors would provide more opportunities to replace the nodes having critical energy level. However, when the number of used *DMs* is larger, the number of attempts to replace nodes exhausting their energy increases and so, the replacement success rate increases. Thus, the ability to replace a node has more effect on the network lifetime when the number of redundant nodes is larger.

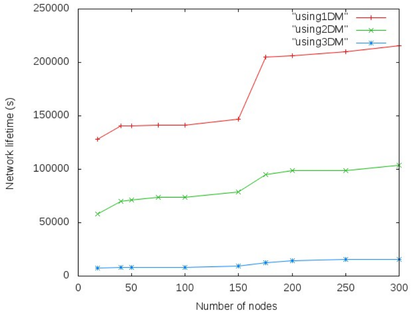

Figure 5. Network lifetime.

Now, we conduct similar simulations with the same configuration, but using 300 static nodes, each one having an initial energy value 30 Joules.

Figure 6 shows the average delivery delay when the square sensing field varies from 300m to 800m, for different number of *DMs*. Shorter data gathering time appears using more data mules on a small sensing field since the tour length is shorter. In addition, the effect of exchanging data between encountered *DMs* appears clearly when the network area is larger. However, the data delivery delay gain become more acute as the number of *DMs* increases, this is due to the concurrent use of multiple *DMs* and the more frequently *DMs* meetings during their tours.

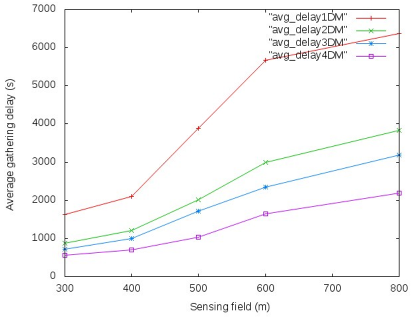

Figure 6. Average data delivery delay.

Table II presents the average delivery delay, denoted *Avg-delay*, and the corresponding consumed energy, denoted *Avg-energy*, on a square area of side length 600m. The consumed energy is the average of the amount of energy consumed in each round of the DMs, starting from the second round. the SPT-DCA strategy uses one mobile collector. We consider that the relay hop bound is 2 hops. We notice that SPT-DCA offers a better data delivery delay in spite of a significant consumed energy. This is due to the local double hop transmission range. Some nodes will relay trafic generated by other sensors. SPT-DCA strategy demonstrates a data delivery delay quite higher of the EDGM strategy using 2 DMs, wile consuming about 18% more energy.

Table II. Simulation results.

Data gathering scheme	Avg-energy	Avg-delay(s)
SPT-DCA	29%	3478.35
EDGM (using 1 DM)	07%	5668,20
EDGM (using 2 DMs)	11%	2995,63
EDGM (using 3 DMs)	12%	2345,19
EDGM (using 4 DMs)	17%	1647,99

Figure 7 plots the rate of packets, meeting a given delivery delay, among all the received packets, on a square area of side length 600m. Through this result, we can determine the number of *DMs* needed to meet a given deadline. For example, for an application having $DDD = 5200s$, using two *DMs* is acceptable to guarantee data delivery to the sink before the deadline. However, using this number of *DMs* is unacceptable for an application having $DDD = 2000s$, this means that some data may arrive to the sink after the deadline. In this case, four *DMs* are required.

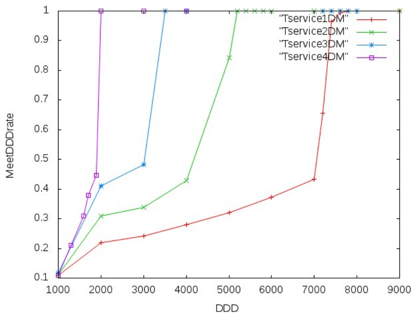

Figure 7. Rate of received packets meeting a given *DDD*.

V. CONCLUSION AND FUTURE WORK

Energy efficient data collection strategies in wireless sensor networks constitute a challenging research domain. In this paper, we proposed a data collection strategy, using data mules. We target especially delay tolerant monitoring applications, requiring a long lifetime. The originality of our contribution consists of joining the coverage and data collection problems. Previous data gathering studies have mainly focused on sparsely deployed sensor networks. Our strategy aims at preserving the coverage property of the sensor network. Indeed, only selected nodes, that are mandatory to ensure coverage, capture data and send them to a single hop node. To validate the proposed solution, we conducted extensive simulations and analyzed the network lifetime and the rate of packets meeting a given delivery delay deadline. Hence, we determine the number of data mules needed to meet a given delivery delay. Larger is the monitoring area leads to more required data mules. Therefore, our future work consists of exploiting the eventual fully connected partition of the sensor network to retrieve a tradeoff between the delivery delay deadline and the number of required data mules. In addition, we are developing an analytical model for predicting the network lifetime obtained by our proposal.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, *Wireless sensor networks: A survey*. Computer Networks, vol. 38, no. 4, March 2002.

[2] H. Zhang, J. C. Hou, *Maintaining sensing coverage and connectivity in large sensor networks*. Technical Report UIUCDCS-R, June 2003.

[3] The ZebraNet Wildlife Tracker. *http://www.princeton.edu/mrm/zebranet.html* [retrieved: April, 2011].

[4] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. *Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet*. In Proc. of conference ASPLOS, 2002.

[5] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, *Exploiting sink mobility for maximizing sensor networks lifetime*. in Proc. 38th HICSS, Big Island, HI, Jan. 2005.

[6] Sh. R. Gandham, M. Dawande, R. Prakash and S. Venkatesan. *Energy-Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations*. Proceedings of IEEE Globecom, December 2003.

[7] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, *Exploiting mobility for energy efficient data collection in wireless sensor networks*. Mobile Netw. Appl. (MONET), vol. 11, no. 3, pp. 327-339, Jun. 2006.

[8] A. Chakrabarti, A. Sabharwal, and B. Aazhang. *Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks*. In Proc. of IPSN, vol. 2634 of LNCS, pp. 129-145, 2003.

[9] S. Nesamony, M. K. Vairamuthu, and M. E. Orlowska.*On Optimal Route of a Calibrating Mobile Sink in a Wireless Sensor Network*. In Proc. of INSS, pp. 61-64, 2007.

[10] M. B. Soares et al. *Hybrid Mobile Robot Navigational Strategy for Efficient Data Collection in Sparsely Deployed Sensor Networks*. In Proc. of IROS, pp. 2833-2838, 2007.

[11] Miao Zhao and Yuanyuan Yang.*Bounded relay hop mobile data gathering in wireless sensor networks*. Mobile Adhoc and Sensor Systems MASS '09, 2009.

[12] R. E. Tarjan. *Depth-First Search and Linear Graph Algorithms*. SIAM Journal on Computing, pp. 146-160, 1972.

[13] Z. Butler and D. Rus, *Event-based motion control for mobile-sensor networks*. IEEE Pervasive Computing, vol. 2, no. 4, pp. 34-42, 2003.

[14] Mannasim. *http://www.mannasim.dcc.ufmg.br* [retrieved: Nov, 2011].

ISBN: 978-1-61208-207-3