Reliable and Energy-efficient Data Collection in Sparse Sensor Networks with Mobile Elements

Giuseppe Anastasi[a], Marco Conti[b], Mario Di Francesco[*,a]

[a] *Dept. of Information Engineering, University of Pisa, Italy*
[b] *CNR-IIT, National Research Council, Italy*

Abstract

Sparse wireless sensor networks (WSNs) are emerging as an effective solution for a wide range of applications, especially for environmental monitoring. In many scenarios, a moderate number of sparsely deployed nodes can be sufficient to get the required information about the sensed phenomenon. To this end, special mobile elements, i.e. mobile data collectors (MDCs), can be used to get data sampled by sensor nodes. In this paper we present an analytical evaluation of the data collection performance in sparse WSNs with MDCs. Our main contribution is the definition of a flexible model which can derive the total energy consumption for each message correctly transferred by sensors to the MDC. The obtained energy expenditure for data transfer also accounts for the overhead due to the MDC detection when sensor nodes operate with a low duty cycle. The results show that a low duty cycle is convenient and allows a significant amount of correctly received messages, especially when the MDC moves with a low speed. When the MDC moves fast, depending on its mobility pattern, a low duty cycle may not always be the most energy efficient option.

Key words: sparse wireless sensor networks, mobile data collectors, reliable data transfer, analytical model

*Corresponding author. Contact address: via Diotisalvi 2, 56126 Pisa (Italy). Phone: +39 050 2217 468. Fax: +39 050 2217 600.

Email addresses: giuseppe.anastasi@iet.unipi.it (Giuseppe Anastasi), marco.conti@iit.cnr.it (Marco Conti), mario.difrancesco@iet.unipi.it (Mario Di Francesco)

1. Introduction

Wireless sensor networks (WSNs) have become an enabling technology for a wide range of applications. The traditional WSN architecture consists in a large number of sensor nodes which are densely deployed over an area of interest. Sensor nodes sample data from their surrounding environment, process them locally and send the results to a data collection point, usually a sink node or an Access Point (AP). The communication between the sensors and the data collection point is ad hoc multi-hop, which is possible due to the network density. More recently, a different WSN architecture has been introduced for application scenarios where fine-grained sensing is not required. For example, many environmental monitoring applications [16] – such as monitoring of weather conditions in large areas, air quality in urban scenarios and so on – can tolerate a low spatial resolution. In this case, nodes are sparsely deployed over the sensing field. As the number of nodes is moderate or low, in contrast with traditional solutions, the costs are reduced. However, since the network is sparse, neighboring nodes are far away from each other, so that they cannot communicate together directly nor through multi-hop paths and a different data gathering scheme is required.

In sparse sensor networks, data collection can be accomplished by means of *mobile data collectors* (MDCs). MDCs are special mobile nodes which are responsible for data gathering. They are assumed to be powerful in terms of data storage and processing capabilities, and not energy constrained, in the sense that their energy source can be replaced or recharged easily. An MDC can be either a Mobile Sink (MS) or a Mobile Relay (MR). Both of them move throughout the WSN and collect data from sensor nodes. An MS is a mobile node which is also the endpoint of data collection [8, 23]. An MR, instead, is a node which carries data from sensors to the sink node or an infra-structured AP [12]. Depending on the application scenario, MDCs may be either part of the external environment [6, 21] (e.g., buses, cabs, or walking people), or part of the network infrastructure [13, 15] (e.g., mobile robots).

The communication between an MDC and sensor nodes takes place in two different phases. First, sensor nodes have to discover the presence of the MDC in their communication range. Then, they can transfer collected data to the MDC while satisfying certain reliability constraints, if required. Different from MDCs, sensor nodes have a limited energy budget, so that both discovery and data transfer should be energy efficient in order to prolong the network lifetime [3]. As the radio component is usually the major source of energy consumption, the overall activity of the radio should be minimized. To this end, a duty cycle approach can be used, so that sensors alternate sleep and active periods. However, the effects of the duty cycle have to be properly investigated: if sensor nodes are sleeping when the MDC comes, they cannot detect it neither transmit data, so that they are only wasting their energy.

In this paper we consider the joint impact of discovery and data transfer for reliable and energy efficient data collection in sparse WSNs with MDCs. Reliable data collection is required in many application scenarios. In many cases, compression techniques are used to reduce the amount of data to be transmitted, and, hence, also the energy consumption [7]. Compression filters the temporal correlation among data. Therefore, the nearly entire amount of compressed data must be available at the sink node (or access point) for the decompression to take place. Reliable data collection is an essential requirement also in applications where the acquired data is used to build a model of the observed phenomenon [3].

The major contribution of this paper is a detailed and realistic model for deriving the performance of the overall data collection process. The proposed methodology is general, so that it can be adapted to different discovery and data transfer protocols, and does not depend on the mobility pattern of the MDC. To the purposes of our analysis, we consider a discovery scheme based on periodic wakeups and an ARQ-based data transfer protocol. Finally, we derive the performance of data collection in terms of both throughput (i.e. average number of messages correctly transferred to the MDC) and energy efficiency (i.e. total energy spent per successfully transferred message) at each contact.

The results obtained show that, in general, a low duty cycle provides a better energy efficiency, especially if the contact time is large enough to allow the reliable transfer of a significant amount of data. However, when the contact time is limited, a very low duty cycle is not convenient as the energy saved during discovery is overcome by the decrease in the number of messages successfully transferred.

The rest of the paper is organized as follows. Section 2 presents an overview of the relevant literature in the field. Section 3 introduces the system model and the related assumptions. Section 4 describes the discovery and data transfer protocols considered in the analysis. Section 5 and 6 present the analytical model for the discovery and the data transfer phases, respectively. Section 7 discusses the obtained results. Finally, section 8 concludes the paper.

2. Related work

Many different papers have addressed the issues of data collection using MDCs. In the context of opportunistic networks, the well known message ferrying approach has been proposed in [25]. Power management has been addressed by [14], where a general framework for energy conservation is introduced. The proposed solution, which can also exploit knowledge about the mobility pattern of the MDC, is evaluated in terms of energy efficiency and delivery performance. However, as the proposed solution is devised for opportunistic networks, it cannot be used straightforwardly in the scenario considered in this paper.

Indeed, many solutions have also been conceived specifically for WSNs. While many papers focus on the mobility of the MDC [10, 11], some works actually address the problem of energy efficient data collection from the sensor nodes' standpoint. For example, [15] considers a periodic wakeup scheme for discovery and a stop-and-wait protocol for data transfer. A stop-and-wait protocol for data transfer is also used in [22], where the MDC is assumed to be controllable. A different solution is investigated in [6], under the assumption that the MDC has a completely predictable mobility. The above mentioned solutions, however, have only been analyzed with simulations, while in this paper we address the problem analytically. In addition, the solution proposed here is flexible enough to support different protocols and mobility patterns of the MDC.

A few papers approach the problem of data collection in WSNs from an analytical point of view. Among them, [23] considers a controllable mobile sink and derives a linear programming formulation to find the optimal way for the sink to visit sensor nodes. In [8] a similar approach is proposed, where multiple sinks are employed. In both cases, it is analytically shown that using a mobile sink leads to a network lifetime much longer than with a static sink. However, these solutions assume that the sensor network is dense enough so that the sink can be reached through a multi-hop path. These solutions are not suitable, however, for very sparse sensor networks, as in the scenario considered in this work.

In the different context of mobile relays, [21] considers MDCs which are not controllable but move randomly, and models the success rate of messages arriving at the access point. However, [21] focuses on sensors' buffer requirements rather than on their energy consumption. Under the same scenario, [12] introduces a more detailed formulation, which considers both the discovery and the data transfer phases of data collection. Furthermore, it evaluates different mobility patterns of the MDC and supports sensor nodes operating with a duty cycle during discovery. Although discussing the probability of data reception at the access point, both [21] and [12] assume an ideal channel and no specific data transfer protocol, so that their findings are mostly affected by buffering constraints. Instead, we explicitly consider data transfer – in addition to discovery – for reliable data collection. In addition, we take the message loss into account by using a model derived from real measurements.

The problem of reliable and energy efficient data collection has also been addressed in [5], where an adaptive and window-based ARQ transmission scheme is evaluated under a realistic message loss model derived from real measurements [4]. In detail, [5] analytically shows that the proposed scheme achieves not only a better throughput, but also a higher energy efficiency than a simple stop-and-wait protocol. A performance evaluation of data collection in the context of sparse WNSs has been carried out in [1] as well. However, both papers have several limitations. First, they only present simulation results. Second, they only consider a specific scenario, either where the static node has only a limited amount of data to transmit [5] or it is continuously generating data [1]. In addition, [5] does not consider the effect of discovery on the subsequent data transfer phase, as it focuses only on data transfer. On the contrary, in this paper we present an analytical model which jointly considers discovery and data transfer for deriving the overall energy efficiency. Furthermore, our model is able to characterize both cases where the sensor has a limited or a (potentially) unlimited number of messages to send to the MDC.

Finally, this paper significantly extends our previous work in [2]. In particular, we have detailed the analysis of both the discovery and the data transfer phases by considering additional parameters and performance metrics. Furthermore, we have also analyzed the scenario where sensor nodes have a limited number of data to send. Moreover, we have surveyed the main approaches proposed for MDC discovery and reliable data transfer, and motivated our choices for the analysis.

3. System model

In this section we will introduce the reference network scenario which is depicted in Figure 1. Specifically, we will consider a single MDC and assume that the network is sparse so that, at any time, the MDC can communicate with at most one static node. We will also assume that the MDC moves along a linear path at a fixed vertical distance (D_y) from the static node, at a constant speed v.

Data collection takes place only during a *contact*, i.e. when the static node and the MDC can reach each other. Furthermore, the area within the radio transmission range R_{tx} of the static node is called *contact area*[1], while the

[1] Depicted with a circular shape in Figure 1 only for convenience.

overall time spent by the MDC inside the contact area is called *contact time*, and is referred to as c_{max}. During a contact, messages exchanged between the MDC and the static node experience a certain message loss. We denote by $p(t)$ the probability that a message transmitted at time t is lost, and assume as $t = 0$ the instant at which the MDC enters the contact area. Any message transmitted when the static node and the MDC are not in contact is assumed to get lost, so that $p(t)$ is defined only within the contact area.

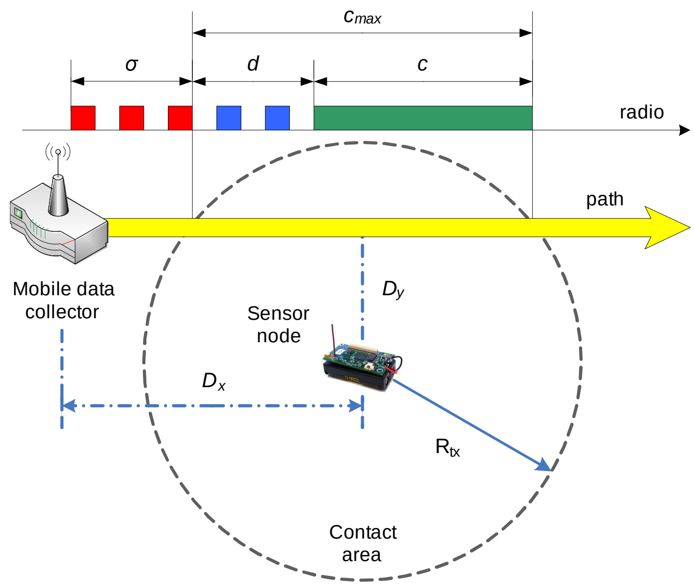

Figure 1: Reference scenario

The overall data collection process can be split into three main phases. Figure 2 shows the state diagram of the static sensor node [14]. As MDC arrivals are generally unpredictable, the static node initially performs a discovery phase for the timely detection of the MDC. Indeed, the successful MDC detection by the static sensor is not immediate, but requires a certain amount of time, called *discovery time*, and denoted as d in Figure 1. Upon detecting the MDC, the static node switches from the discovery state to the data transfer state, and starts transmitting data to the MDC. As a result of the discovery process, the static node cannot exploit the whole available contact time for data transfer.

The portion of the contact time which can be actually used for subsequent data transfer is called *residual contact time* and is referred to as c. After the end of the data transfer phase, the static node may switch to the discovery state again in order to detect the next MDC passage. However, if the MDC has a (even partially) predictable mobility, the static node can exploit this knowledge to further reduce its energy consumption [14]. In this case, the static node can go to a sleep state until the next expected arrival of the MDC. In any case, the static sensor may be awake also when the MDC is out of reach. The amount of time spent by the static node in the discovery state while the MDC has not yet entered the contact area is called *waiting time*, and is indicated with σ in Figure 1.

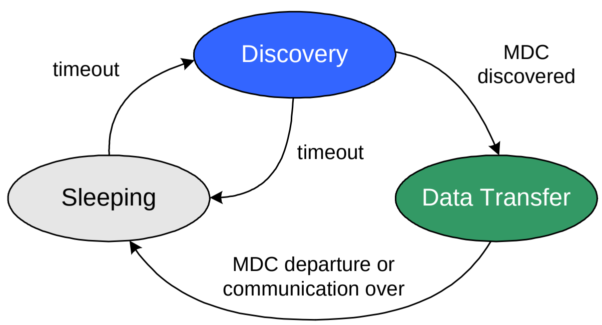

Figure 2: State diagram for the static node during data collection

In principle, different discovery and data transfer protocols can be used for data collection in the scenario introduced above. For the subsequent analysis, however, it is necessary to consider specific protocols for both discovery and data transfer. These protocols will be described in the following section.

4. Data collection protocols

In the next subsections we will briefly survey the most significant schemes proposed in the literature for MDC discovery and reliable data transfer, respectively. In addition, we will introduce the specific protocols considered in the analysis, and the motivations behind their choice.

4.1. Discovery protocol

MDC discovery schemes can be broadly classified as *synchronous*, *on-demand*, and *asynchronous*, schemes.

Synchronous schemes assume that sensor nodes know exactly when the MDC will enter the contact area, and can thus wakeup at pre-defined times [6]. Obviously, such approaches require that the mobility of the MDC is accurately known in advance. However, this assumption is rather strong, unless the motion of the MDC can be controlled. As we are assuming here that the MDC is not controllable, a synchronous scheme cannot be used in our scenario.

On-demand schemes are based on the idea that the static node should wakeup only when the MDC is able to communicate with it. The main challenge associated with such an approach is how to inform the sleeping node that the MDC is in the contact area. To this end, two main approaches can be used. In the first, nodes use multiple radios [19, 24]. A long-range and high-power radio is used for data communication, while a low-range low-power radio is used for awaking nodes. A different approach exploits a radio triggered activation, similarly to Radio Frequency IDentification (RFID) systems [9]. In this case the MDC sends wakeup messages (or signals) which have enough energy to trigger the activation of the static sensor node. On-demand schemes are appealing in the context of sparse WSNs, because they are able to significantly reduce the energy consumption of sensor nodes. In addition, they allow a very timely detection of the MDC. However, they have some major disadvantages. For instance, they have a very short coverage range, limited to a dozen meters in most cases. This can be a very limiting factor in a large number of applications – e.g., in the context of urban monitoring – as a short range contact between the MDC and static sensors may not always be possible. In addition, they require special hardware support, which is not available on currently off-the-shelf commercial platforms. For these reasons we did not consider an on-demand scheme for the discovery protocol.

Finally, asynchronous schemes define sleep/wakeup patterns whose properties ensure that nodes are able to communicate without explicitly agreeing on their activation instants. There are several variants of asynchronous schemes, mainly based on two different categories. In the first, the MDC sends periodic discovery messages, while the static node periodically wakes up and listens for advertisements for a short time. If it does not detect any discovery message it can return to sleep, otherwise it can start transferring data to the MDC [20]. A similar approach consists in replacing the stream of discovery messages with a single long discovery message or tone. In this case the listening time of receivers can be very short, provided that the duration of the discovery message is at least equal to the listening period. Such a scheme has been extensively used in the context of traditional (dense) WSNs [17, 24], where it has been proved to be efficient and robust. However, as it requires using long preambles, it is not very appealing in the specific scenario we are considering. In fact, as the MDC is moving, any delay in the detection of discovery messages can compromise the subsequent data transfer phase, as it reduces the residual contact time. Nevertheless, it is not possible for the MDC to emit a continuous tone, as the static sensor would never be able to access the channel in this case, unless multiple radios are used. Hence, we considered an asynchronous protocol based on a stream of discovery messages, which is described below.

To advertise its presence in the surrounding area, the MDC periodically sends special messages called *beacons* (Figure 3(a)). The duration of a beacon message is equal to T_{BD}, and subsequent beacons are spaced by a *beacon period*, indicated with T_B. In order to save energy during the discovery phase, the static node operates with a duty cycle δ, defined by the active time T_{ON} and the sleep time T_{OFF}, i.e. $\delta = T_{ON}/(T_{ON} + T_{OFF})$. The activity time of the static node is set to $T_{ON} = T_B + T_{BD}$, so that the node is allowed to receive a complete beacon during its active time[2], provided that it wakes up when the MDC is in the contact area. On the other hand, the sleep time T_{OFF} is set to a value which enforces the desired duty cycle δ.

[2]Recall that in our scheme we are using messages instead of tones.

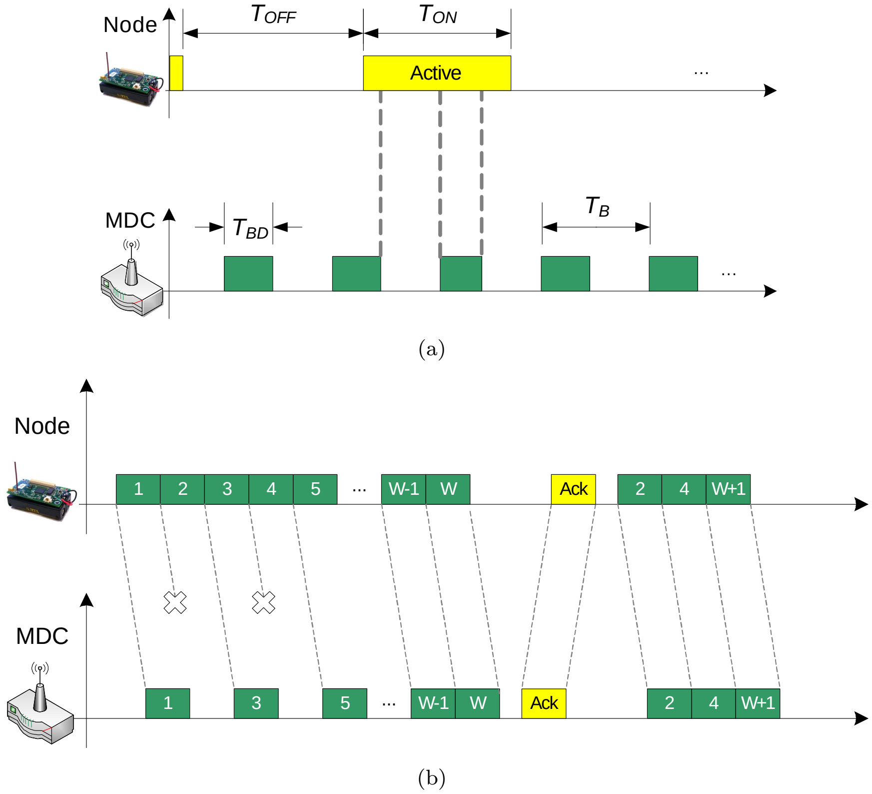

Figure 3: Discovery (a) and data transfer (b) protocols used for data collection

4.2. Data transfer protocol

There are also many different alternatives as for the data transfer protocol. Most common approaches are based on *Automatic Repeat reQuest* (ARQ) schemes, especially in the context of sparse WSNs [15, 21, 22]. ARQ basically consists in a retransmission strategy based on acknowledgements and timeouts. Data sent by the source node has to be explicitly acknowledged by the receiver before a timeout expires. If the acknowledgement message is not received in time, the sender retransmits its data. Also in this case there are several variants. In the simplest case – i.e. Stop-and-Wait – the source simply sends a single message at once, and waits for the corresponding acknowledgement, like in [15, 22]. More advanced techniques involve sending more than one message at once, so that the acknowledgement message can embed information about the whole group of messages (Go-back-N) or even single messages in the group being correctly received (Selective Retransmission) [5].

A different approach consists in using *Erasure Coding* (EC) instead of retransmissions. In this case, transmitted data include additional redundant messages, so that the original information can be recovered even when a number of messages is lost [18]. However, EC schemes are not very convenient in this context. This is because the static node needs a (periodic) feedback from the MDC to decide whether it is still in contact or not. Hence, the advantage of using Erasure Coding – i.e. that they do not require an acknowledgement message – cannot be fully exploited. In addition, an EC scheme is more useful in a context where multiple MDCs are simultaneously present in the same contact area, which is rather unlikely in the context of sparse WSNs we are considering. Due to these motivations, we decided not to use an EC-based data transfer scheme. Instead, we considered an ARQ-based protocol with selective retransmission, which is outlined below.

Upon receiving a beacon from the MDC, the static node enters the data transfer state. While in this state, the static node remains always active to exploit the residual contact time as much as possible. On the other hand, the MDC enters the data transfer phase as soon as it receives the first message sent by the static node, and stops beacon transmissions. The communication scheme adopted during the data transfer phase is based on Automatic Repeat reQuest (ARQ). The static node splits buffered data into messages, which are transmitted in groups (windows). The number of messages contained in a window, i.e. the *window size*, is assumed to be fixed and known both at the sender and at the receiver. The static node sends messages in a window back to back, then waits for an acknowledgement sent back by the MDC (Figure 3(b)). The acknowledgement message contains a mask indicating which individual messages of the corresponding window have been correctly received by the MDC. If the acknowledgement message is lost, the static nodes simply retransmits the previous window as is. Otherwise, the static node flushes the messages which have been correctly received by the MDC and fetches the new messages to transmit from its local buffer.

The end of the data transfer phase depends on the adopted communication paradigm. If the static sensor has just a limited amount of buffered data – which can be transmitted in a fraction of the contact time – it simply goes to sleep when it has already sent all data in its buffer. Otherwise, i.e. when it has always data to send, the static node uses all the available contact time and goes to sleep only when the MDC is not reachable any more. However, the static node generally cannot know when the MDC leaves the contact area, for instance because it cannot derive the residual contact time a priori[3]. In practice, the static node assumes that the MDC has exited the contact area when it misses N_{ack} consecutive acknowledgements.

5. Discovery phase analysis

In this section we will develop an analytical model for the discovery phase, while in the next section we will analyze the data transfer phase. For convenience, we summarized the main symbols used throughout the paper in Table 1. The purpose of the discovery phase analysis is to derive the distribution of the discovery time and, thus, the residual contact time as well. The analysis is split in two main phases. First, the state of the static node (i.e. ON or OFF) over time is derived, by keeping in consideration the duty cycle. Second, the beacon reception process is modeled, i.e. the state transitions of the static node are characterized, on the basis of the probability that a beacon sent by the MDC at a given instant will be correctly received by the static sensor.

With the help[4] of Figure 4 we introduce the framework for the subsequent

[3]Recall that, in this context, the acknowledgement message is used not only for implementing a retransmission strategy, but also as an indication of the MDC presence in the contact area.

[4]For convenience, we have plotted a sample message loss function $p(t)$, whose trend is representative for the scenario under consideration. In fact, at first the message loss decreases with time, until a minimum is achieved near the middle of the contact time. This is because, after the MDC has entered the contact area, it moves closer to the sensor node. Specifically, the MDC reaches the minimum distance from the sensor node when it is at the middle point

Table 1: Summary of the main symbols used throughout the paper

Symbol	**Description**
v	Speed of the MDC
c_{max}	Contact time
c	Residual contact time
d, D	Discovery time and the associated random variable
σ	Waiting time
T_B	Beacon period
T_{BD}	Beacon duration
T_{ON}	Active time
T_{OFF}	Sleep time
δ	Duty cycle
$s(t)$	Radio state at time t
$r(t)$	Residual time in the current radio state at time t
$X(k)$	State probability vector at the k-th beacon transmission
N	Maximum number of beacons within a contact
T_s	Message slot duration
w	Window size
$p(t)$	Message loss probability at time t
$N(t)$	Total number of messages received by the MDC during the window starting at time t
$R(t)$	Total number of messages acknowledged by the MDC at the end of the window starting at time t
R	Total number of messages acknowledged by the MDC during a contact
W	Number of windows contained in the residual contact time
$L(q)$	Latency for tranferring q messages
P_{rx}	Power consumption of the radio in receive mode
P_{sl}	Power consumption of the radio in sleep mode
P_{tx}	Power consumption of the radio in transmit mode
N_{ack}	Number of consecutive acks to be missed by the sensor to consider the MDC as out of range
$\overline{E}_{dt,r}$	Average total energy per message when the sensor has always data to send
$\overline{E}_{dt,q}$	Average total energy per message when the sensor has only q messages to send

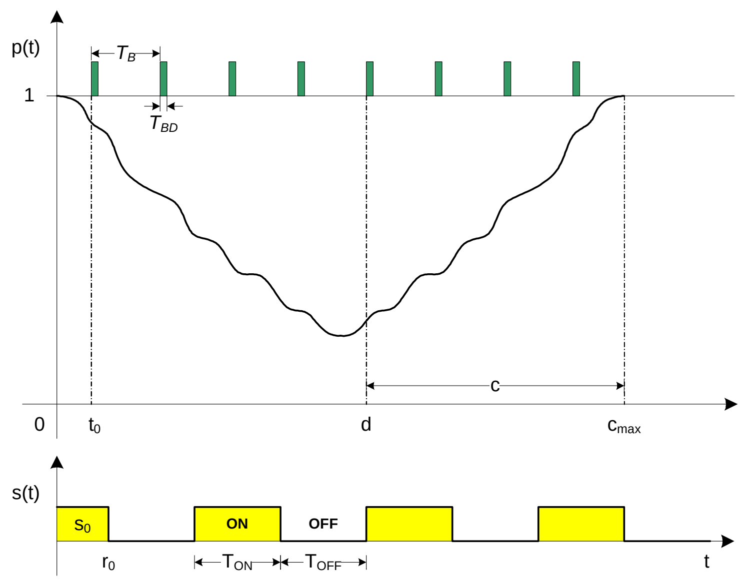

Figure 4: Example of beacon discovery process

analysis. As beacon transmissions do not depend on when the MDC enters the contact area, the initial beacon transmission within the contact area is generally affected by a random offset (with respect to the beginning of the contact time). In detail, the time instant at which the MDC transmits the first beacon while in the contact area is denoted as t_0. As beacon transmissions are periodic and start at t_0, the actual instants of subsequent beacon transmissions can be expressed as $t_n = t_0 + n \cdot T_B$, with $n \in [1, N-1]$ where $N = \lceil c_{max}/T_B \rceil$ is the maximum number of beacons the MDC can send while in the contact area. Therefore, if the MDC is discovered by means of the m-th beacon, the discovery time is $d = d_m(t_0) = t_0 + m \cdot T_B$, and the corresponding residual contact time is $c = c_m(t_0) = c_{max} - d = c_{max} - d_m(t_0)$.

The state of the static node at a given instant is completely specified by its composite state (s, r) where s denotes the radio state, i.e. ON or OFF, and

of the path traversed within the contact area (see Figure 1). After the minimum, the message loss probability starts to increase with time until it becomes equal to one. Obviously, this opposite trend is due to the MDC moving away from the sensor node and, at a point, going out of the contact area.

r represents the residual time, i.e. the amount of time the node will remain in the same state s. The initial state of the static node at the time $t = 0$ is referred to as (s_0, r_0). Let us denote by $s(t)$ and $r(t)$ the radio state and the residual time, respectively, at a generic time t. Because of the duty cycle, both $s(t)$ and $r(t)$ evolve in a deterministic way. In detail, the radio state of the static node is periodic, with period equal $T_{ON} + T_{OFF}$. We focus now on the radio state $s(t_n)$ of the static node at beacon transmission times (t_n). As $s(t)$ is periodic, it is sufficient to investigate the remainder of the ratio between the beacon transmission times and the period of the duty cycle. By comparing this remainder against the initial residual state s_0 and the initial residual time r_0, it is possible to derive $s(t_n)$ (see Appendix A for a detailed derivation). Specifically, it is

$$\underset{s_0=\mathrm{ON}}{s(t_n)} = \begin{cases} \mathrm{ON} & \text{if } 0 \leq t'_n < r_0 \\ \mathrm{OFF} & \text{if } r_0 \leq t'_n < r_0 + T_{OFF} \\ \mathrm{ON} & \text{if } r_0 + T_{OFF} \leq t'_n < T_{ON} + T_{OFF} \end{cases} \tag{1}$$

$$\underset{s_0=\mathrm{OFF}}{s(t_n)} = \begin{cases} \mathrm{OFF} & \text{if } 0 \leq t'_n < r_0 \\ \mathrm{ON} & \text{if } r_0 \leq t'_n < r_0 + T_{ON} \\ \mathrm{OFF} & \text{if } r_0 + T_{ON} \leq t'_n < T_{ON} + T_{OFF} \end{cases} \tag{2}$$

where $t'_n = t_n \mod (T_{ON} + T_{OFF})$. Similarly, we can also derive the residual time $r(t_n)$

$$\underset{s_0=\mathrm{ON}}{r(t_n)} = \begin{cases} r_0 - t'_n & \text{if } 0 \leq t'_n < r_0 \\ T_{OFF} + r_0 - t'_n & \text{if } r_0 \leq t'_n < r_0 + T_{OFF} \\ T_{ON} + T_{OFF} + r_0 - t'_n & \text{if } r_0 + T_{OFF} \leq t'_n < T_{ON} + T_{OFF} \end{cases} \tag{3}$$

$$\underset{s_0=\mathrm{OFF}}{r(t_n)} = \begin{cases} r_0 - t'_n & \text{if } 0 \leq t'_n < r_0 \\ T_{ON} + r_0 - t'_n & \text{if } r_0 \leq t'_n < r_0 + T_{ON} \\ T_{ON} + T_{OFF} + r_0 - t'_n & \text{if } r_0 + T_{ON} \leq t'_n < T_{ON} + T_{OFF} \end{cases} \tag{4}$$

Once the duty-cycled state of the static node has been fully characterized, we have to model the actual beacon reception process. To this end we introduce the state representation illustrated in Figure 5, where the states B_i, $i \in [0, N + 1]$,

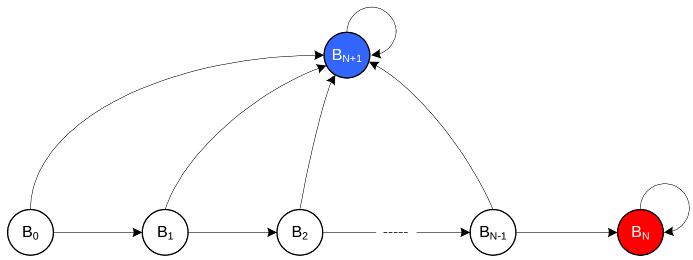

Figure 5: Beacon states

refer to the static node at beacon transmission times. In detail, B_0 is the initial state of the static node, where the MDC has not yet transmitted the first beacon while in the contact area. B_j is entered by the static node after missing the first j beacons sent by the MDC, where $j \in [1, N-1]$. B_N is entered when the static node has not detected the MDC presence at all, because it has not received any of the beacons. Finally, B_{N+1} is entered when the static node has successfully received a beacon. When it is in a state B_k, with $k \in [0, N-1]$, the static node can only move to the state B_{k+1} or to the state B_{N+1} if it has lost or got a beacon, respectively. Note that B_N and B_{N+1} are absorbing states. In addition, the state of the static node is completely specified by its current state.

Now, we can derive a joint characterization of the radio state of the static node and the beacon reception process. For simplicity, time has been discretized in slots with duration Δ, so that the whole process can be modeled as a discrete time Markov chain. For the sake of clarity, in the following we will not explicitly refer to time-dependent parameters by their actual discretized values, unless otherwise specified. The transition matrix H corresponding to the beacon reception process can be thus written as follows

$$\mathbf{H} = \begin{pmatrix} 0 & H_{01} & 0 & \cdots & 0 & H_{0,N+1} \\ 0 & 0 & H_{12} & & 0 & H_{1,N+1} \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & H_{N-1,N} & H_{N-1,N+1} \\ 0 & 0 & 0 & \cdots & H_{NN} & 0 \\ 0 & 0 & 0 & \cdots & 0 & H_{N+1,N+1} \end{pmatrix}$$

where the H_{kl} are sub-blocks denoting the transition probability from the state B_k to the state B_l. Note that the state B_0 is evaluated at time $t = 0$, while state B_i with $i \in [1, N]$ is evaluated at the i-th beacon transmission time, i.e. t_i. In addition to the state B related to the beacon reception, the H_{kl} blocks also keep track of the radio state of the static node. In detail, the elements of the H_{kl} block can be expressed as

$$h^{kl}_{(s_i,r_i),(s_j,r_j)} = \mathbb{P}\left\{B_l, (s_j, r_j) \,|\, B_k, (s_i, r_i)\right\}$$

Since the state of the static node is deterministic, the only admissible transitions are those specified by the state equations (1-4), i.e. between the generic state (s_i, r_i) and the corresponding state (s^*_j, r^*_j) such that $s^*_j = s(t_k)$ and $r^*_j = r(t_k)$. Specifically, the transition probabilities are as follows

$$h^{kl}_{(s_i,r_i),(s^*_j,r^*_j)} = \begin{cases} 1 & \text{if } s^*_j = \text{OFF and } B_l \neq B_{N+1} \\ 0 & \text{if } s^*_j = \text{OFF and } B_l = B_{N+1} \\ p(t_k) & \text{if } s^*_j = \text{ON and } B_l \neq B_{N+1} \\ 1 - p(t_k) & \text{if } s^*_j = \text{ON and } B_l = B_{N+1} \end{cases}$$

The above probabilities can be justified as follows, assuming to be in the state B_k.

- If the radio will be OFF during the next beacon transmission (at time t_k), then the static node will miss the beacon for sure, so that it can only enter a state different from B_{N+1} (i.e. $B_l = B_{k+1}$).

- Otherwise, the static node will be active during the next beacon transmission time. The static node will miss the beacon with a probability $p(t_k)$, thus moving to the state $B_l = B_{k+1}$. Conversely, it will correctly receive the beacon with a probability $1-p(t_k)$ thus entering the state $B_l = B_{N+1}$.

Let $\mathbf{X}^{(0)}$ be the initial state probability vector of the static node and $\mathbf{X}^{(k)}$ the state probability vector associated to the time of the k-th beacon transmission, with $k \in [1, N-1]$,

$$
\begin{aligned}
\mathbf{X}^{(k)} &= \begin{pmatrix} X_0^{(k)} & X_1^{(k)} & \cdots & X_{N-1}^{(k)} & X_N^{(k)} & X_{N+1}^{(k)} \end{pmatrix} \\
\mathbf{X}^{(0)} &= \begin{pmatrix} X_0^{(0)} & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}
\end{aligned}
$$

where only the $X_0^{(0)}$ component of the initial state vector is not zero, as when the MDC enters the contact area the static node is waiting for the first beacon to be sent. By definition of discrete time Markov chain, it follows that

$$
\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} \cdot \mathbf{H} \quad \text{for } k = 0, 1, 2, \ldots, N-1 \tag{5}
$$

Note that the X_{N+1}^{k} component of the state vector represents the cumulative probability of the MDC discovery after k beacon transmissions. Hence, the p.m.f. of the discovery time r.v. D, i.e. $d(m, t_0) = \mathbb{P}\{D(t) = m\}$, can be derived as

$$
d(m, t_0) = \begin{cases} X_{N+1}^{(0)} & \text{if } m = t_0 \\ X_{N+1}^{(k)} - X_{N+1}^{(k-1)} & \text{if } m = t_k,\ k \in [1, N-1] \\ 0 & \text{otherwise} \end{cases} \tag{6}
$$

It may be worthwhile pointing out that $\mathbf{X}^{(0)}$ and $\mathbf{H}$ can be rewritten in a more compact form. Specifically, the non-zero sub-blocks of $\mathbf{X}^{(k)}$ for $k = 0, 1, 2, \ldots, N-1$ can be derived as follows

$$
X_{k+1}^{(k+1)} = X_k^{(k)} \cdot H_{k,k+1} \tag{7}
$$

$$
X_D^{(k+1)} = X_k^{(k)} \cdot H_{k,N+1} + X_{N+1}^{(k)} \cdot H_{N+1,N+1} \tag{8}
$$

To derive the p.m.f. of the discovery time by (5) we need to know the initial state probabilities for $\mathbf{X}^{(0)}$. As both beacon transmissions and activations of the static node are periodic and independent, it is reasonable to assume that the initial radio state and the initial residual time are uniformly distributed along all possible values. Hence, for both radio states and independent from the residual time, the initial probability is $\Delta/(T_{ON} + T_{OFF})$, where Δ is duration of a discretized time slot.

All the above discussion assumes a certain initial beacon transmission time t_0. To properly characterize the discovery time, the Equation (6) must be evaluated for all possible values of t_0. Again, as both beacon transmissions and activations of the static node are periodic and independent, we will assume that all possible values of t_0 are uniformly distributed in the range $0 \leq t_0 < T_B$. Note that t_0 has been disctretized into $\hat{t}_0 \in \mathcal{T} \equiv \{0, \Delta, \ldots, n_{t_0} \cdot \Delta\}$, where $n_{t_0} = \lfloor T_B/\Delta \rfloor$ is the maximum number of discretized time slots Δ which fit into $[0, T_B)$. Hence, the p.m.f. $d(m)$ of the discovery time per contact is

$$d(m) = \sum_{\hat{t}_0 \in \mathcal{T}} d(m, \hat{t}_0) \cdot \mathbb{P}\left\{\hat{t}_0\right\} = \frac{\Delta}{T_B} \sum_{\hat{t}_0 \in \mathcal{T}} d(m, \hat{t}_0)$$

6. Data transfer phase analysis

In this section we will derive the amount of messages correctly transferred by the static node to the MDC. Recall that the static node enters the data transfer phase after a successful beacon reception. Since this depends on the discovery time, we will make use of the p.m.f. $d(m)$ obtained in the previous section to derive the number of correctly transferred messages.

As anticipated in Section 4, while in the data transfe state, the static sensor is always on, and uses an ARQ-based communication protocol for data transfer. In the following, we will assume that both data and acknowledgement messages have a fixed duration T_s, referred to as *message slot*. In addition, we will assume a window size of w messages.

We focus now on a single window starting at the generic time t (see Figure 6). As the message loss changes with the distance between the MDC and the static sensor, every message will experience its own loss probability. However, we will assume that the message loss is constant during a message slot, i.e. that the i-th message in the window starting at time t will experience a message loss probability $p(t + i \cdot T_s)$. This is reasonable, given the short duration of the message slot.

Let's denote by $N(i, t)$ the r.v. denoting the number of messages successfully received by the MDC in a given slot i of the window starting at time t. Clearly,

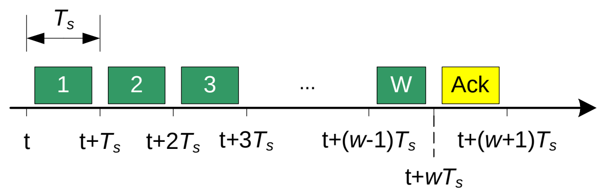

Figure 6: ARQ-based communication for data transfer

the p.m.f. of $N(i,t)$ is $n(i,t,m) = \mathbb{P}\{N(i,t) = m\}$, i.e.

$$n(i,t,m) = \begin{cases} 1 - p(t + i \cdot T_s) & \text{if } m = 1 \\ p(t + i \cdot T_s) & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Hence, the total number of messages received by the MDC within a window, i.e. $N(t)$, is the sum of the $N(i,t)$ r.v.s

$$N(t) = \sum_{i=0}^{w-1} N(i,t)$$

so that its p.m.f. is the convolution of the single p.m.f.s, i.e. $n(t,m) = \otimes_{i=0}^{w-1} n(i,t,m)$.

Furthermore, we denote by $R(t)$ the r.v. representing the number of messages correctly transferred to the MDC when an ARQ-based mechanism is used. So, in this case $R(t)$ represents the number of messages acknowledged by the MDC. In the following, we will consider a selective retransmission scheme, where acknowledgements notify the sensor node which messages sent in the last window have been correctly received by the MDC. Hence, the reception of the acknowledgement has to be accounted as well, so that the messages within a window are correctly transferred if they are successfully received by the MDC and the corresponding acknowledgement is not lost. We denote by $A(t)$ the r.v. indicating the number of acknowledgements correctly received by the MDC for the corresponding window starting at time t. Hence, the p.m.f. of $A(t)$ is $a(t,m) = \mathbb{P}\{A(t) = m\}$, i.e.

$$a(t,m) = \begin{cases} 1 - p(t + w \cdot T_s) & \text{if } m = 1 \\ p(t + w \cdot T_s) & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

hence $R(t) = N(t) \cdot A(t)$ and, being them indepedent, we have that

$$\mathbb{E}\left[R(t)\right] = \mathbb{E}\left[N(t)\right] \cdot \mathbb{E}\left[A(t)\right] = \sum_{i=0}^{w-1} [1 - p(t + i \cdot T_s)] \cdot [1 - p(t + w \cdot T_s)] \tag{11}$$

6.1. Joint discovery and data transfer

The foregoing discussion focuses on a single communication window. To get the number of messages transferred during the whole contact time, we have to characterize both the starting time t of the first window and the total number W of windows actually available in the residual contact time. Hence, the total number of messages correctly transferred during a contact is

$$R = \sum_{i=0}^{W} R(t + i \cdot (w+1) \cdot T_s)|_{t=D} \tag{12}$$

If D is the r.v. denoting the discovery time, whose p.m.f. has been derived in the previous section, clearly the start time of the first communication window is $t = D$. In addition, the number of windows in the residual contact time is

$$W = \left\lfloor \frac{c_{max} - D}{(w+1) \cdot T_s} \right\rfloor$$

under the assumption that the static node can exploit all the residual contact time for data transfer[5]. Thus, the number of messages successfully transferred can be expressed as the r.v. R which depends only on the discovery time D.

In the discussion above, we have implicitly considered the case in which the sensor has always data to send, i.e. the data transfer spans over the whole residual contact time. In many cases, however, the static sensor may have a limited number of data to transmit – for example because it has already aggregated sensed data into a small number of messages [7]. In such conditions, the static sensor may use only a limited part of the residual contact contact time to transfer all buffered data. To this end, it is worth considering the amount of time needed to transfer buffered data, referred to as *data transfer latency*. Let

[5]We verified by simulation that with an appropriate setting of N_{ack} is possible to exploit the residual contact time almost completely (see Section 7.2 for the details).

R_j be the number of messages correctly transferred after j windows, where the first window starts at time t. It is

$$R_j = \sum_{i=0}^{j-1} R(t + i \cdot (w+1) \cdot T_s)|_{t=D} \tag{13}$$

Then, the probability $l_j(q)$ of receiving q messages after j windows is, for $j > 0$

$$\mathbb{P}\{l_j(q)\} = \mathbb{P}\{R_j \geq q\} - \sum_{k=0}^{j-1} \mathbb{P}\{l_k(q)\} \tag{14}$$

i.e. the probability of receiving at least q messages after j windows, given that the same amount of data has not yet been transferred in the previous $j-1$ windows (in those cases the associated data transfer latency is exactly l_k each). Clearly, $\mathbb{P}\{l_0(q)\} = 0$, so that Equation (14) can be recursively solved. Once we have the resulting probabilities, we can immediately derive the data transfer latency r.v. $L(q,t)$ for transfers which start at time t. Finally, we can get the data transfer delay per contact as $L(q) = L(q,t)|_{t=D}$, where D is the r.v. of the discovery time.

6.1.1. Energy efficiency

In this section we will derive the total energy consumed by the static node per message successfully delivered to the MDC, on the average. This metric provides an indication of the energy efficiency for the overall data collection process. The energy consumed in a given radio state is calculated as $P_{rs} \cdot T_{rs}$, where P_{rs} is the power drained in the state rs while T_{rs} is the time spent in the same state. As possible radio states we consider rx for receive, tx for transmit and sl for sleep (i.e. shutdown). In addition, we assume that the power consumption when the radio is idle (i.e. it is monitoring the channel) is the same as in the receive state. As the energy efficiency depends on both discovery and data transfer, we derive first the energy consumption for the discovery phase, and then its joint effect on the subsequent data transfer.

Since the MDC arrival may be unknown a priori, the static node may spend a waiting time σ in addition to the discovery time (see Figure 1). Hence, the average energy spent during the discovery phase is

$$\overline{E}_{disc} = P_{sl} \cdot (\sigma + \mathbb{E}[D]) \cdot (1-\delta) + P_{rx} \cdot (\sigma + \mathbb{E}[D]) \cdot \delta$$

where the first term accounts for the energy spent in the sleep state, while the second one accounts for the time spent in the active state before the correct reception of the beacon message.

On the other side, the average energy spent for data transfer depends on the actual communication paradigm. When the static sensor has always data to send and data transfer takes the entire residual contact time, we have

$$\overline{E}_{dt,r} = \left(\frac{\mathbb{E}[c_{max} - D]}{w+1} + \mathbb{P}\{D\} \cdot \frac{N_{ack}}{2} \cdot T_s \right) \cdot (w \cdot P_{tx} + P_{rx}) \tag{15}$$

The first part of the equation denotes the average number of windows in the residual contact time plus the average number of windows wasted after the end of the contact. In fact, the static node can detect the loss of N_{ack} consecutive acknowledgements either before or after the (nominal) end of the contact with the MDC. Hence, the static node can remain awake for at most N_{ack} windows after the MDC has exited the contact area. We are using $N_{ack}/2$ in Equation 15 under the assumption that the static node remains awake – after the contact time – for a number of windows which is uniformly distributed in $[0, N_{ack}]$. Note that the wasted windows have to be considered only when the contact actually occurs, hence the related term has to be multiplied by the probability that the MDC is correctly detected (i.e. $\mathbb{P}\{D\} = X_N^{(N-1)}$). The second term in Equation (15), instead, denotes the amount of power spent during each window – the static sensor transmits w messages and, then, waits for the related acknowledgement.

On the contrary, when the amount of data is limited, the static sensor remains active only for the time needed to successfully transfer q messages, i.e. the data transfer latency $L(q)$. Hence, we have

$$\overline{E}_{dt,q} \quad = \quad \frac{\mathbb{E}[L(q)]}{w+1} \cdot (w \cdot P_{tx} + P_{rx})$$

Finally, the average total energy consumed by the static sensor per each message correctly transferred to the MDC for the two different cases is, respectively

$$\begin{aligned}\overline{E}_r^{msg} &= \frac{1}{\mathbb{E}\left[R\right]} \cdot (\overline{E}_{disc} + \overline{E}_{dt,r}) \\ \overline{E}_q^{msg} &= \frac{1}{q} \cdot (\overline{E}_{disc} + \overline{E}_{dt,q})\end{aligned}$$

7. Results

In this section we will use the analytical formulas derived in the previous sections to perform an integrated performance analysis of the overall data collection process. To this end, we will consider the following performance metrics.

- *Residual contact ratio*, defined as the average of the ratio between the residual contact time and the contact time

$$\eta = \mathbb{E}\left[\frac{c_{max} - D}{c_{max}}\right]$$

- *Contact miss ratio*, defined as the fraction of MDC passages not detected by the static sensor (i.e. $\mathbb{P}\{N\} = X_{N+1}^{(N-1)}$).
- *Throughput*, defined as the average number of messages (or bytes) correctly transferred to the MDC at each contact (i.e. $\mathbb{E}\left[R\right]$).
- *Probability of bulk reception*, defined as the probability that the whole amount q of messages is successfully transferred to the MDC in a single contact (as defined in Equation (14)).
- *Average data transfer latency*, defined as the average time needed by the static sensor to successfully transfer an amount q of messages to the MDC in a single contact (i.e. $\mathbb{E}\left[L(q)\right]$).
- *Energy consumption per byte*, defined as the mean energy spent by the static sensor per each message (or byte) correctly transferred to the MDC (i.e. $\overline{E}_{msg}$ as defined in Subsection 6.1.1).

In our analysis we used the same message loss model considered in [5]. It was derived from experimental data measured under the same scenario introduced

Table 2: Interpolated message loss coefficients as functions of the MDC speed ($D_y = 15$m)

Coefficient	$v = 3.6$ km/h	$v = 40$ km/h
a_0	0.133	0.4492
a_1(s^{-1})	0	0
a_2(s^{-2})	0.000138	0.0077
c_{max}(s)	158.53	16.915

in Section 3 [4]. Specifically, we used a polynomial interpolation of the message loss in the form

$$p(t) = a_2 \cdot \left(t - \frac{c_{max}}{2}\right)^2 + a_1 \cdot \left(t - \frac{c_{max}}{2}\right) + a_0 \tag{16}$$

Equation (16) holds only within the contact area, i.e. for $0 < t < c_{max}$. For other values of t, $p(t)$ is assumed to be equal to one, as outside of the contact area any transmitted message is lost. To derive the coefficients in Equation (16) – reported in Table 2 for different MDC speeds v and for a vertical distance $D_y = 15$ m – we used the same methodology described in [5].

We evaluated the model derived in Sections 5 and 6 and validated the analytical results with a discrete event simulator written in C. The simulator implements the discovery and the data transfer protocols presented in Section 4. More specifically, the simulator operates under the same scenario already introduced in Section 3 and uses the trace-driven realistic message loss model in [4]. Note that, as it implements the full-featured data transfer protocol, the simulator is able to obtain the actual end of the contact, resulting from the loss of N_{ack} consecutive acknowledgements, while the model assumes that the residual contact time is completely exploited for data transfer.

To derive confidence intervals we used the replication method with a 90% confidence level. In all experiments we performed 10 replicas, each consisting of 10.000 MDC passages. The obtained confidence intervals are always very low (below 1%) and are thus omitted. In the following, we will show both analytical and simulation results. However, unless stated otherwise, we will refer to the analytical results. Table 3 shows the parameter settings for both analysis and simulation.

Table 3: Parameters used for analysis

Parameter	**Value**
Transmit power (0 dBm)	49.5 mW
Receive (idle) power	28.8 mW
Sleep power	0.6 μW
Message payload size	24 bytes
Message slot size	15 ms
T_B	100 ms
T_{BD}	9.3 ms

7.1. Discovery phase

In this section we will evaluate the performance of the discovery protocol, in terms of residual contact ratio and missed contacts. Obviously, the effectiveness of the discovery protocol strictly depends on the beacon period T_B, which must be larger than or equal to the beacon duration T_{BD} (9.3 ms in our scenario). In addition, the discovery process is clearly influenced by the duty cycle used by the static sensor.

Figures 7 and 8 show the residual contact ratio for three different beacon periods and several duty cycles, when the MDC moves at 3.6 km/h and 40 km/h, respectively[6]. To validate our analytical model we also performed some simulation experiments under the same conditions. The comparison between simulation and analytical results shows that our model is very accurate.

The results in Figure 7 clearly show that the static node quickly discovers the MDC when it moves slowly (i.e. at 3.6 km/h). As expected, for a fixed duty cycle, the residual contact ratio decreases when T_B increases. Also, for a given beacon period, the residual contact ratio decreases with the duty cycle. However, most of the contact time can be effectively used for communication even when the duty cycle is very low. The results are different, instead, when the speed of the MDC is high (i.e. 40 km/h). From Figure 8 we can see that the residual contact ratio drops sharply when the duty cycle changes from 10%

[6]We got results similar to the 3.6 km/h scenario also when the speed is 20 km/h, so we omitted them for the sake of space

to 0.5%, especially for large values of T_B. Intuitively, this behavior can be explained as follows. For the same beacon period, a lower duty cycle involves longer delays in the detection of the MDC. When the MDC moves fast, the time it remains in the contact area is lower – in this scenario the average contact time is about 17s – and, hence, a late discovery results in a small amount of time left for data transfer.

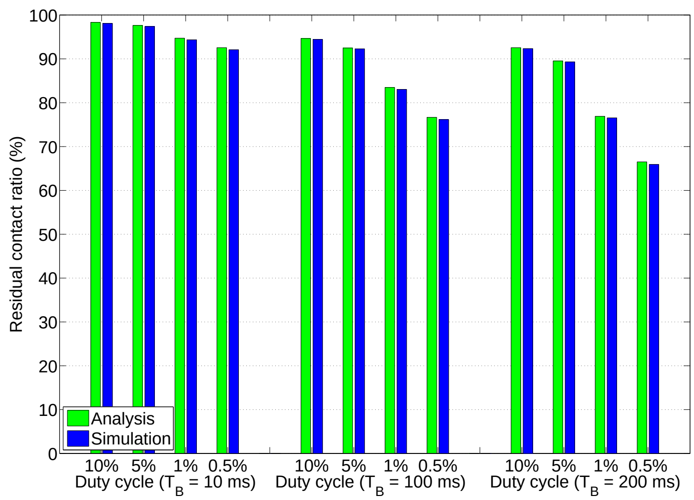

Figure 7: Residual contact ratio as a function of the beacon period for $v = 3.6$ km/h

In addition, there are cases where the passage of the MDC is not detected at all, i.e. the contact is missed. This happens because, when the contact time is short, the number of beacons emitted by the MDC within the contact area is low. As a consequence, the static sensor has a low probability to correctly receive a beacon, especially if it is using a low duty cycle. To quantify this effect, we measured the fraction of MDC passages which are not detected by the static node. In the 3.6 km/h scenario we observed a non negligible number of missed contacts only when the beacon period is 200 ms and the duty cycle is 0.5%. Instead, the percentage of missed contacts is very high in the 40 km/h scenario, as shown in Figure 9. The static node misses contacts even with a 10% duty cycle when T_B is large. In detail, when the duty cycle is 1% the contact miss ratio is over 40% for T_B equal to 100 and 200 ms. The results are even worse

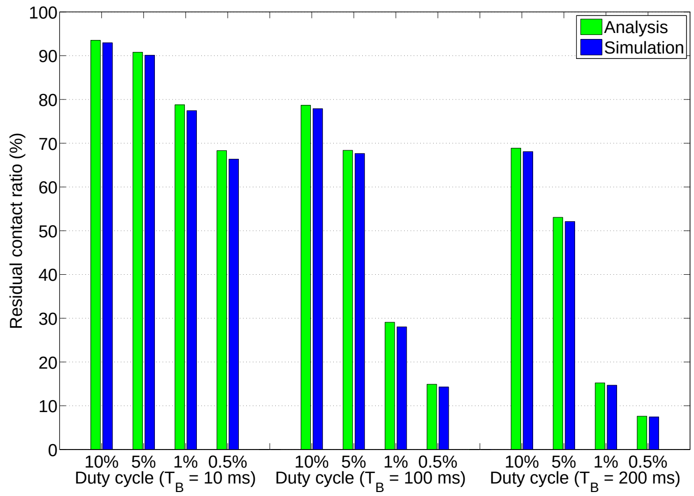

Figure 8: Residual contact ratio as a function of the beacon period for v = 40 km/h

for the 0.5% duty cycle. This is because – when the duty cycle is so low – the sleep time becomes comparable to the entire contact time. As a consequence, the chance that the static node does not detect the passage of the MDC at all – because it has not correctly received any beacon during its active time – is much higher than in the other cases.

From the above results it emerges that a low T_B value would be preferable as this increases the frequency of beacon transmission and, hence, the probability of a timely beacon discovery. On the other hand, the T_B value must be larger than the beacon duration T_{BD}, which depends on the specific sensor platform that is used (T_{BD}=9.3 ms in our case). In addition, when T_B is too low (e.g., 10 ms) most of the beacon period is occupied by the beacon transmission and, hence, the static node might not be able to reply promptly to the MDC after receiving a beacon. Based on these remarks, in all the subsequent analysis we will use T_B = 100 ms, which is realistic in the context of implementing the discovery protocol for a Mote platform [1].

7.2. Data transfer

In this section we will evaluate the performance of the ARQ-based data transfer protocol in terms of throughput and data transfer latency. It is worth

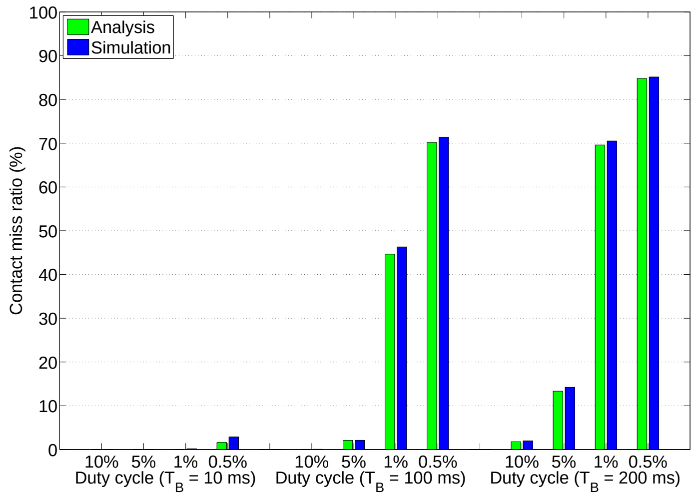

Figure 9: Contact miss ratio as a function of the beacon period for $v = 40$ km/h

recalling that the following results have been obtained for the actual residual contact time, so that they account for the effects of the discovery phase as well.

We start with considering the throughput – in terms of messages acknowledged by the MDC – for the scenario in which the static node sends data during all the residual contact time. Figure 10 and Figure 11 show the throughput achieved when the MDC moves at different speeds. Note that our analysis assumes that the data transfer phase takes all the residual contact time while, in practice, the static sensor assumes that the MDC has exited the contact area after missing N_{ack} consecutive acknowledgements. To evaluate the impact the above-mentioned assumption we used simulation. From the above figures we can see that our results are very accurate. There is only a slight difference between analytical and simulation results, which is higher for low values of the window size. This is because the time after which the sensor assumes that the MDC is out of reach is $(w + 1) \cdot N_{ack} \cdot T_s$, i.e., it increases with the window size (for a fixed value of N_{ack}). Therefore, the probability of an incorrect decision (the MDC is assumed to be out of range, while it is still in) is higher when the window size is smaller. Such incorrect decisions typically occur when the MDC is discovered very quickly, i.e. when the message loss is still high. This also suggest that the value of N_{ack} should be tailored to the specific scenario, which depends on the speed of the MDC. For instance, we used $N_{ack} = 25$ and $N_{ack} = 10$ for the 3.6 km/h and the 40 km/h scenarios, respectively.

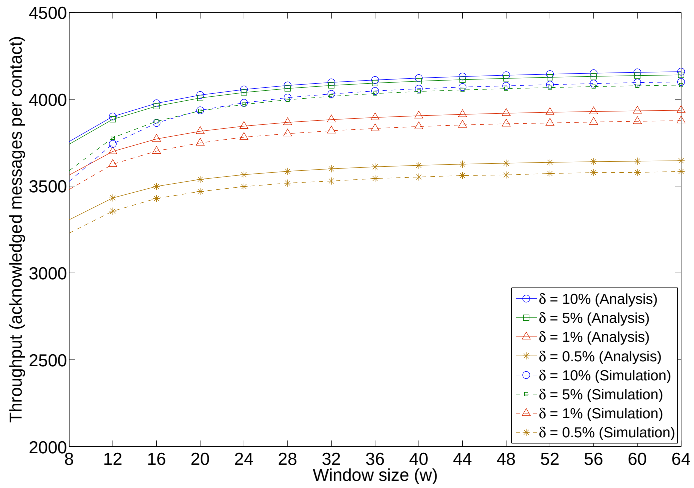

Figure 10: Throughput as a function of the window size for $v = 3.6$ km/h

Figure 10 shows the throughput as a function of the window size when $v = 3.6$ km/h. We can see that the throughput increases with the window size for all considered duty cycles, so that the maximum is obtained with the largest considered window size of 64 messages. This is due to the acknowledgement overhead, which decreases as the window size increases. Specifically, the throughput reaches over 4000 messages per contact, corresponding to about 100 kB of data, for the 10% duty cycle. Similar results are also obtained with the lower 5% and 1% duty cycles. Such results can be explained on the basis of the residual contact ratios, which are similar for the different duty cycles, i.e. the residual contact time is not reduced significantly when a low duty cycle is used. The same is not true for the 0.5% duty cycle, which actually experiences a lower, but still reasonable, throughput of about 3500 messages per contact (nearly 88 kB).

Figure 11 shows the throughput as a function of the window size when $v = 40$ km/h. The throughput has a different trend in this case. It first increases

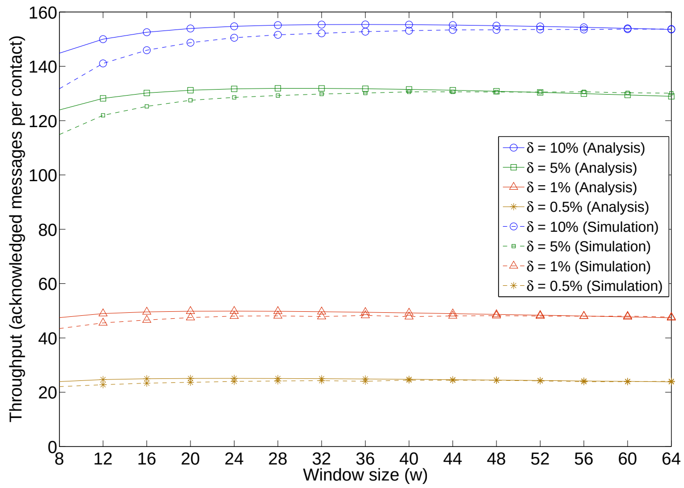

Figure 11: Throughput as a function of the window size for $v = 40$ km/h

when the window size is low, then decreases after a point which depends on the duty cycle. In addition, the obtained throughput changes significantly with the duty cycle. In fact, while the 10% and the 5% duty cycles both achieve a similar throughput over 100 messages per contact (around 3 kB), the 1% duty cycle gets only 50 messages (1.2 kB) per contact. The lowest 0.5% duty cycle even obtains a throughput of 25 messages per contact (0.6 kB), which is rather low, but may be enough for certain applications.

We consider now the scenario where the static sensor has a limited number of messages to send during a contact, referred to as bulk. Figure 12 shows the probability of successful bulk reception when the MDC moves at 40 km/h and the window size is 32. We can see that it is very high – more than 90% – for a wide range of bulk sizes, when the duty cycle is over 5%. However, it drops below 50% when the duty cycle is low (e.g., 1% and 0.5%). This is strictly related to the average throughput achievable in the different cases (see Figure 11). Specifically, some values of the bulk size are over the average throughput achieved for the same duty cycle in the scenario where the sensor node transmits data during the overall residual contact time. In these cases, therefore, there is a very low probability of bulk reception, as the probability decreases significantly

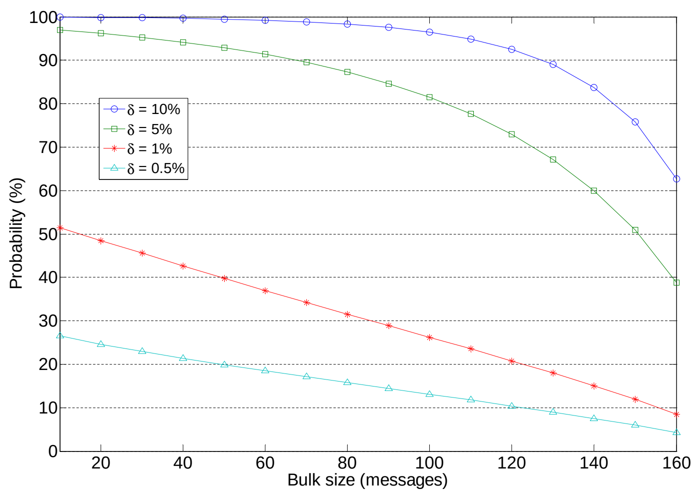

Figure 12: Probability of bulk reception as a function of the number of messages for $v = 40$ km/h and $w = 32$

when the bulk size increases.

Figure 13 shows that the average bulk transfer latency – besides increasing with the bulk size, which is quite obvious – also increases with the duty cycle. This counter intuitive result can be explained by considering that higher duty cycles allow a shorter discovery time. This implies that the data transfer phase starts earlier, i.e., when the packet loss is higher (see Figure 4). Thus, under the same bulk size, higher duty cycles take a longer time to transfer messages because of the worse message loss. However, if we take into account both the discovery time and the data transfer latency, the total time needed to transfer a bulk decreases when the duty cycle increases, as expected.

7.3. Energy efficiency

In this section we will evaluate the energy efficiency of data collection. It should be noted that the considered energy consumption accounts for both discovery and data transfer, so that it fully characterizes the overall data collection process. While in the previous sections we have considered only what happens within the contact area, in the following we take into account also the time spent by the static sensor on waiting for the MDC to enter the communication range.

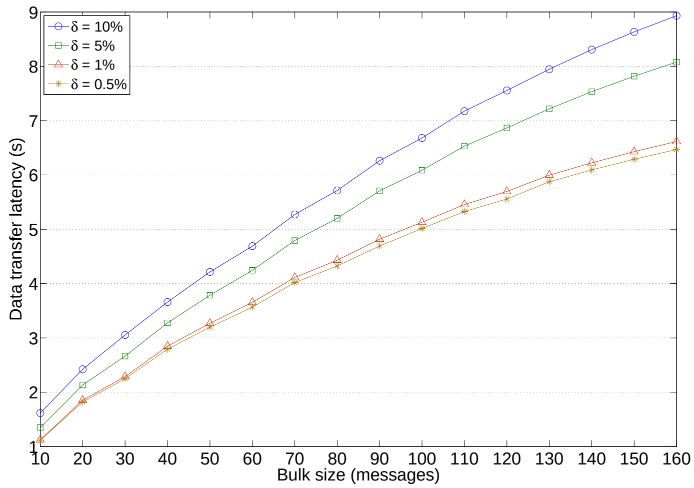

Figure 13: Data transfer latency as a function of the number of messages for $v = 40$ km/h and $w = 32$

To this end, we measured the average energy spent per acknowledged message as a function of the waiting time. In the following, we will limit our discussion to the scenario in which the static sensor sends data during the whole residual contact time, as this is the worst case scenario.

Figure 14 shows the average energy consumption per acknowledged message when the MDC moves at $v = 40$ km/h and the window size is 32. Clearly the energy consumption increases with the waiting time, but a very low duty cycle is not necessarily the most convenient option. In fact, when the average waiting time is below 25 s (i.e. the MDC arrival can be predicted with rather good accuracy), the best option is the 10% duty cycle. Instead, when the waiting time is between 25 s and 200 s, the most convenient duty cycle is 5%. From later on, i.e. when the MDC presence is very difficult to estimate, the best duty cycle is 1%. In addition, the 0.5% duty cycle always gets a higher average energy consumption than the 1% duty cycle. These results are in contrast to the expected behavior, i.e. that the energy consumption decreases with the duty cycle, as it happens in the scenario where the MDC moves at 3.6 km/h (we have omitted the correspondent figure for the sake of space).

Actually, these results can be explained as follows. Low duty cycles may

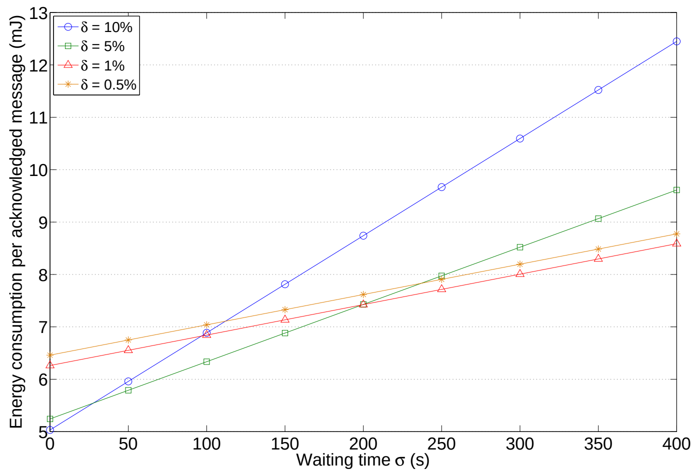

Figure 14: Energy efficiency as a function of the waiting time for $v = 40$ km/h

delay the MDC detection, leading to lower residual contact times. In addition, they may also produce high contact miss ratios, so that the energy spent during discovery is simply wasted, as the sensor node cannot transmit any data. In the considered scenario, where the MDC moves with $v = 40$ km/h, the contact time and the residual contact time are short. When the waiting time is low, i.e. when the sensor knows the MDC arrival times with a good accuracy, the energy overhead due to discovery is negligible, because the sensor spends most of its active time during data transfer. Instead, the advantages of low duty cycles become relevant when the waiting time is high. In this case, the sensor may spend a significant amount of time by looking for beacons when the MDC is out of the contact area. Thus, a low duty cycle reduces the activity of the sensor during the waiting time, which is the highest share of the overall energy consumption. Regarding the 0.5% duty cycle, it is always unsuitable in this scenario, because the energy gain due to the lower activity time is thwarted by the decrease in the throughput (see Figure 11).

8. Conclusions

In this paper we have developed an analytical model of the overall data collection process in sparse sensor networks with mobile data collectors (MDCs). The model is flexible enough to incorporate different discovery and data transfer protocols. We limited our discussion to a discovery algorithm where the MDC sends periodic advertisements and the sensors follow an asynchronous scheme based on a low duty cycle. In addition, we considered an ARQ communication protocol with selective retransmission for reliable data transfer. Our findings show that low duty cycles can be actually used for a large class of environmental monitoring applications. Surprisingly, a low duty cycle may not always be the most energy efficient option, depending on a number of different factors such as the speed and the mobility pattern of the MDC.

This work could be improved along different directions. For example, the model proposed in this paper could be extended to the case of multiple MDCs. In addition, the findings of our analysis could be used as a basis for the definition of adaptive data collection protocols, which are capable to tailor the operating parameters to the actual conditions (i.e. knowledge about the mobility pattern of the MDC, buffer constraints, residual energy of sensor nodes etc.).

Acknowledgements

Work funded partially by the European Commission under the FP6-2005-NEST-PATH MEMORY project, and partially by the Italian Ministry for Education and Scientific Research (MIUR) under the FIRB ArtDeco and PRIN WiSe DeMon projects.

References

[1] G. Anastasi, M. Conti, and M. Di Francesco. Data collection in sensor networks with Data Mules: an integrated simulation analysis. In *Proc. of IEEE ISCC 2008*, 2008.

[2] G. Anastasi, M. Conti, and M. Di Francesco. An analytical study of reliable and energy-efficient data collection in sparse sensor networks with mobile elements. In *Proc. of EWSN 2009*, 2009.

[3] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3), May 2009.

[4] G. Anastasi, M. Conti, E. Gregori, C. Spagoni, and G. Valente. Motes sensor networks in dynamic scenarios. *International Journal of Ubiquitous Computing and Intelligence*, 1(1), April 2007.

[5] G. Anastasi, M. Conti, E. Monaldi, and A. Passarella. An adaptive data-transfer protocol for sensor networks with Data Mules. In *Proc. of IEEE WoWMoM 2007*, 2007.

[6] A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In *Proc. of IPSN 2003*, 2003.

[7] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 2(2):70–87, April 2007.

[8] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Proc. of IEEE GLOBECOM 2003*, 2003.

[9] L. Gu and J. Stankovic. Radio-triggered wake-up for wireless sensor networks. *Real-Time Systems Journal*, 29:157–182, 2005.

[10] Y. Gu, D. Bozdag, and E. Ekici. Mobile element based differentiated message delivery in wireless sensor networks. In *Proc. of IEEE WoWMoM 2006*, 2006.

[11] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C. Lee. Partitioning based mobile element scheduling in wireless sensor networks. In *Proc. IEEE SECON 2005*, pages 386–395, 2005.

[12] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *ACM/Springer Mobile Networks and Applications*, 11(3):327–339, June 2006.

[13] D. Jea, A. Somasundra, and M. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Proc. of IEEE DCOSS 2005*, 2005.

[14] H. Jun, M. Ammar, and E. Zegura. Power management in delay tolerant networks: A framework and knowledge-based mechanisms. In *Proc. of IEEE SeCon 2005*, 2005.

[15] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *Proc. of ACM Mobisys 2004*, 2004.

[16] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of ACM WSNA 2002*, 2002.

[17] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for sensor networks. In *Proc. of ACM SenSys '04, November 3-5*, 2004.

[18] L. Rizzo. Effective erasure codes for reliable computer communication protocols. *SIGCOMM Comput. Commun. Rev.*, 27(2):24–36, 1997.

[19] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. B. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.

[20] C. Schurgers, V. Tsiatsis, and M. B. Srivastava. Stem: Topology management for energy efficient sensor networks. In *Proc. of IEEE Aerospace Conference '02, Big Sky, MT, March 10-15*, 2002.

[21] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *Proc. IEEE SNPA 2003*, 2003.

[22] A. Somasundara, A. Kansal, D. Jea, D. Estrin, and M. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 5(8), 2006.

[23] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *Proc. HICSS 2005*, 2005.

[24] X. Yang and N. Vaidya. A wakeup scheme for sensor networks : Achieving balance between energy saving and end-to-end delay. In *Proc. of IEEE RTAS '04, pp. 19-26*, 2004.

[25] W. Zhao and M. Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *Proc. IEEE FTDCS 2003*, May 2003.

A. Derivation of state equations

As already mentioned in Section 5, the state of the static sensor at a generic time t is defined by two functions, $s(t)$ and $r(t)$. In detail, with $s(t)$ we denote the radio state of the static sensor, i.e. if it is ON of OFF at a given time t. In addition, with $r(t)$ we denote the residual time, i.e. the additional time that the node will spend in the current radio state $s(t)$.

We start deriving the evolution of the radio state. Recall from Section 3 that the static node is using a duty cycle δ during the discovery process, so that it remains awake for a time T_{ON} and goes to sleep for a time T_{OFF} periodically.

For convenience, let's define the following rectangular function to represent the radio state

$$x(t) = \begin{cases} ON & \text{if } 0 \leq t < T_{ON} \\ OFF & \text{if } t \geq T_{ON} \end{cases}$$

As we can see from Figure 15, the radio state of the static node can be expressed

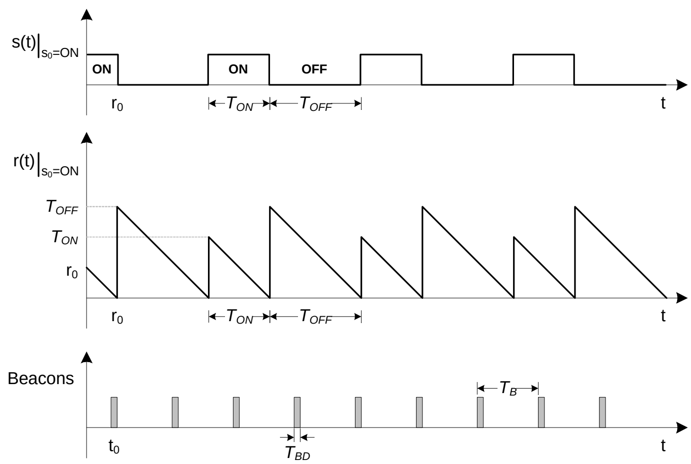

Figure 15: Detail of the state of the static sensor during discovery

as a series of rectangular pulses. Note that $s(t)$ depends on the initial conditions, i.e. the radio state and the corresponding residual time of when the MDC enters the communication range of the static node. In detail, we can express $s(t)$ as follows

$$\begin{aligned} \underset{s_0=ON}{s(t)} &= u(t) \cdot x(t + (T_{ON} - r_0)) \\ &+ \sum_{k=0}^{\infty} x(t - (r_0 + T_{OFF} + k(T_{ON} + T_{OFF}))) \qquad (17) \\ \underset{s_0=OFF}{s(t)} &= x(t - r_0) + \sum_{k=1}^{\infty} x(t - (r_0 + k(T_{ON} + T_{OFF}))) \qquad (18) \end{aligned}$$

where we denote as $u(t)$ the unit-step function. As the evolution of the radio state is periodic – except for the initial period where $t \leq r_0$ – we can rewrite

Equations (17) and (18) in the equivalent form, $\forall k = 0, 1, 2, \ldots$

$$\underset{s_0=ON}{s(t)} = \begin{cases} ON & \text{if } 0 \leq t < r_0 \\ OFF & \text{if } \begin{array}{l} r_0 + k(T_{ON} + T_{OFF}) \leq t \\ < r_0 + T_{OFF} + k(T_{ON} + T_{OFF}) \end{array} \\ ON & \text{if } \begin{array}{l} r_0 + T_{OFF} + k(T_{ON} + T_{OFF}) \leq t \\ < r_0 + (k+1)(T_{ON} + T_{OFF}) \end{array} \end{cases} \tag{19}$$

$$\underset{s_0=OFF}{s(t)} = \begin{cases} OFF & \text{if } 0 \leq t < r_0 \\ ON & \text{if } \begin{array}{l} r_0 + k(T_{ON} + T_{OFF}) \leq t \\ < r_0 + T_{ON} + k(T_{ON} + T_{OFF}) \end{array} \\ OFF & \text{if } \begin{array}{l} r_0 + T_{ON} + k(T_{ON} + T_{OFF}) \leq t \\ < r_0 + (k+1)(T_{ON} + T_{OFF}) \end{array} \end{cases} \tag{20}$$

More specifically, we can also derive the state of the static sensor at the istants of beacon transmissions, i.e. $t_n = t_0 + n \cdot T_B$, with $n \in [1, N-1]$ where $N = \lceil c_{max}/T_B \rceil$. Specifically, we can write Equations (17) and (18) in the equivalent form

$$\underset{s_0=\text{ON}}{s(t_n)} = \begin{cases} ON & \text{if } 0 \leq t'_n < r_0 \\ OFF & \text{if } r_0 \leq t'_n < r_0 + T_{OFF} \\ ON & \text{if } r_0 + T_{OFF} \leq t'_n < T_{ON} + T_{OFF} \end{cases} \tag{21}$$

$$\underset{s_0=\text{OFF}}{s(t_n)} = \begin{cases} OFF & \text{if } 0 \leq t'_n < r_0 \\ ON & \text{if } r_0 \leq t'_n < r_0 + T_{ON} \\ OFF & \text{if } r_0 + T_{ON} \leq t'_n < T_{ON} + T_{OFF} \end{cases} \tag{22}$$

where $t'_n = t_n \mod (T_{ON} + T_{OFF})$ by definition. Note that Equations (21) and (22) are the same as Equations (1) and (2).

As for the residual state, we follow a similar derivation. In detail, we first define the following triangular functions for convenience

$$w_{ON}(t) = \begin{cases} T_{ON} - t & \text{if } 0 \leq t < T_{ON} \\ 0 & \text{if } t > T_{ON} \end{cases}$$

$$w_{OFF}(t) = \begin{cases} T_{OFF} - t & \text{if } 0 \leq t < T_{OFF} \\ 0 & \text{if } t > T_{OFF} \end{cases}$$

so that we can write $r(t)$ as follows (see Figure (15))

$$\begin{aligned}
\underset{r_0=ON}{r(t)} &= u(t) \cdot w_{ON}(t + (T_{ON} - r_0)) \\
&+ \sum_{k=0}^{\infty} w_{OFF}(t - (r_0 + k(T_{ON} + T_{OFF}))) \\
&+ \sum_{k=0}^{\infty} w_{ON}(t - (r_0 + T_{OFF} + k(T_{ON} + T_{OFF}))) \qquad (23) \\
\underset{s_0=OFF}{r(t)} &= u(t) \cdot w_{OFF}(t + (T_{OFF} - r_0)) \\
&+ \sum_{k=0}^{\infty} w_{ON}(t - (r_0 + k(T_{ON} + T_{OFF}))) \\
&+ \sum_{k=0}^{\infty} w_{OFF}(t - (r_0 + T_{ON} + k(T_{ON} + T_{OFF}))) \qquad (24)
\end{aligned}$$

Then, we rewrite Equations (23) and (24), $\forall k = 0, 1, 2, \ldots,$ as

$$\underset{s_0=\mathrm{ON}}{r(t_n)} = \begin{cases} r_0 - t & \text{if } 0 \le t < r_0 \\ T_{OFF} + r_0 - t & \text{if } \begin{array}{l} r_0 + k(T_{ON} + T_{OFF}) \le t \\ < r_0 + T_{OFF} + k(T_{ON} + T_{OFF}) \end{array} \\ T_{ON} + T_{OFF} + r_0 - t & \text{if } \begin{array}{l} r_0 + T_{OFF} + k(T_{ON} + T_{OFF}) \le t \\ < r_0 + (k+1)(T_{ON} + T_{OFF}) \end{array} \end{cases}$$

$$\underset{s_0=\mathrm{OFF}}{r(t_n)} = \begin{cases} r_0 - t & \text{if } 0 \le t < r_0 \\ T_{ON} + r_0 - t & \text{if } \begin{array}{l} r_0 + k(T_{ON} + T_{OFF}) \le t \\ < r_0 + T_{ON} + k(T_{ON} + T_{OFF}) \end{array} \\ T_{ON} + T_{OFF} + r_0 - t & \text{if } \begin{array}{l} r_0 + T_{ON} + k(T_{ON} + T_{OFF}) \le t \\ < r_0 + (k+1)(T_{ON} + T_{OFF}) \end{array} \end{cases}$$

Finally, we write Equations (23) and (24) in the equivalent form

$$\underset{s_0=\mathrm{ON}}{r(t_n)} = \begin{cases} r_0 - t'_n & \text{if } 0 \le t'_n < r_0 \\ T_{OFF} + r_0 - t'_n & \text{if } r_0 \le t'_n < r_0 + T_{OFF} \\ T_{ON} + T_{OFF} + r_0 - t'_n & \text{if } r_0 + T_{OFF} \le t'_n < T_{ON} + T_{OFF} \end{cases} \qquad (25)$$

$$\underset{s_0=\mathrm{OFF}}{r(t_n)} = \begin{cases} r_0 - t'_n & \text{if } 0 \le t'_n < r_0 \\ T_{ON} + r_0 - t'_n & \text{if } r_0 \le t'_n < r_0 + T_{ON} \\ T_{ON} + T_{OFF} + r_0 - t'_n & \text{if } r_0 + T_{ON} \le t'_n < T_{ON} + T_{OFF} \end{cases} \qquad (26)$$

i.e., in the same form as Equations (3) and (4), respectively.