ARTICLE IN PRESS

Computer Networks xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# Energy efficient and reliable data delivery in urban sensing applications: A performance analysis

Eleonora Borgia [a,*], Giuseppe Anastasi [b], Marco Conti [a]

[a] Institute of Informatics and Telematics, National Research Council (CNR), Italy
[b] Department of Information Engineering, University of Pisa, Italy

ARTICLE INFO

*Article history:*
Received 26 October 2012
Received in revised form 26 June 2013
Accepted 8 July 2013
Available online xxxx

*Keywords:*
Urban sensing
Wireless sensor networks
Mobile sinks
Reliable data delivery
Erasure coding
Analytical model and implementation

ABSTRACT

Urban sensing is an emerging application field for Wireless Sensor Networks (WSNs), where a number of static sensors is sparsely deployed in an urban area to collect environmental information. Data sensed by each sensor are, then, opportunistically transmitted to Mobile Nodes (MNs) that happen to be in contact. In the considered scenario, communications between MNs and sensors require paradigms with a minimal synchronization between devices, extremely fast and energy efficient, especially at the sensor side. To deal with the above issues, in [1] we proposed a hybrid protocol for data delivery from sensors to MNs, named Hybrid Adaptive Interleaved Data Protocol (HI). By combining Erasure Coding (EC) with an Automatic Repeat reQuest (ARQ) scheme, the proposed protocol maximizes the reliability of communications while minimizing the energy consumed by sensors. In this paper, we present an in-depth analysis of the HI performance. We provide an analytical evaluation by defining a flexible model to derive the probability of data delivery and exploiting it to investigate the performance over a wide range of parameters. Moreover, we perform an experimental study to evaluate the HI effectiveness on real sensor platforms. Specifically, we analyze the impact of resource constraints imposed by sensors on data delivery and provide a careful characterization of its actual consumption of resources.



## 1. Introduction

Wireless Sensor Networks (WSNs) play a crucial role in the realization of the Future Internet, which will be truly pervasive, ubiquitous and content-centric with data generated automatically by applications/devices or by users [2,3]. In this scenario, WSNs are a powerful technology to automatically generate content, gather and process the data generated and, thus, to increase the awareness of a certain environment or phenomenon. Their applicability spans over a large variety of domains [4–6]. Relevant applications include, among others, environmental monitoring, surveillance, event detection, intelligent agriculture, health monitoring.

Traditionally, a WSN consists of static and resource-constrained sensor nodes, densely deployed in the sensing area, so that sensed data can be relayed through multi-hop communication and collected by a sink node [7–12]. More recently, there has been an increased interest in the research community to introduce mobile elements in WSNs, to increase their performance [13–15]. Mobile elements can be either regular sensor nodes or special nodes that are exploited to support the data-collection process. In the former case, sensor nodes can be mounted, for instance, on vehicles (e.g., taxis or buses) and exploited for pollution monitoring or surveillance of urban areas [16]. In the latter case, sensor nodes are still static, however, the WSN is enhanced with special mobile data-collector nodes. These special mobile nodes can be either mobile

* Corresponding author. Address: Via G. Moruzzi 1, 56124 Pisa, Italy. Tel.: +39 050 315 2407; fax: +39 050 315 2593.
*E-mail addresses:* eleonora.borgia@iit.cnr.it (E. Borgia), giuseppe.anastasi@iet.unipi.it (G. Anastasi), marco.conti@iit.cnr.it (M. Conti).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.07.025

elements specifically introduced in the sensing field for data collection (e.g., mobile robots), or mobile elements that already exist in the environment and can, thus, be exploited to collect data from sensor nodes. In the latter case, they are also known as Mobile Nodes (MNs), data mules or message ferries [17].

Robots are often sophisticated mobile entities (e.g., Unmanned Aerial Vehicles) that are able to make complex decisions in order to move close to sensor nodes to collect the data. On the other hand, the mobility patterns of MNs cannot be controlled and it is, typically, independent from the WSN operations. Hence, sensor nodes must adapt their behavior when a MN happens to be within their transmission range. Despite these limitations, WSN with MNs represents a cost effective solution for data collection in sensor networks, especially when sensor nodes are sparsely deployed in the environment and, thus, they cannot communicate directly. MNs move in the environment for their own business (e.g., city buses) and are exploited to collect the sensed data from the sensor nodes, without the need to deploy new and costly elements such as mobile robots. Sparse WSNs require a much lower number of nodes than traditional (dense) WSNs. This reduces not only the cost of the WSN, but also contention and collisions. Finally, since MNs can traverse the network to collect data, the energy consumption can be spread more uniformly in the network as well as reduced, therefore contributing to make the environment more sustainable and greener [18]. As a result, the network lifetime can be significantly improved [19,20]. For the above reasons, in this paper we focus on data collection in sparse WSN with MNs.

Sparse sensor networks are particularly appealing for urban applications [21,22]. In this case, a low number of static sensors can be placed in a few strategic points of a city to collect physical information about the environment, e.g., the level of pollutants or allergens in the air. Data generated by sensors can be collected by MNs, which can be either vehicles (e.g., bikes, cars, buses or shuttles) or people carrying a smartphone. MNs can either use the collected information for their own purposes, make them available to other MNs that happen to be in contact with them, or send them to remote user or sink node, through a long-range communication facilities (e.g., GPRS/UMTS).

In such a scenario, data delivery presents several issues. One of them is the limited contact time available to sensors for communicating with MNs. This is especially true when MNs are vehicles moving at a high speed. A different aspect is related to the variable quality of wireless communication. The wireless channel is known to be noisy, especially in urban scenarios where there might be several sources of interferences. Since MNs are mobile, the message loss rate is highly time-varying, and significantly affected by the physical distance between sensors and MNs [21]. As a consequence, communication protocols targeted to data delivery in such a scenario should be reliable, and should also have a limited overhead in order to exploit the short and limited contacts to the full extent. In this context, approaches based on Erasure Coding (EC) have shown to be effective [23]. To deal with all the above issues, the Hybrid Adaptive Interleaved Data Protocol (HI), a hybrid communication protocol for reliable data delivery from sensors to MNs, has been proposed [1]. HI efficiently combines EC with an ARQ scheme such that: (i) reliability of communications is significantly enhanced and (ii) energy consumption at the sensor node is drastically reduced. The simulation analysis carried out in [1] has demonstrated that the proposed hybrid communication protocol outperforms a pure ARQ scheme based on acknowledgments and selective retransmissions, especially when many MNs are simultaneously in contact with the sensor. However, being based on simulation experiments only, the analysis in [1] is not exhaustive because it does neither study the general properties of the hybrid communication protocol, nor its effectiveness when implemented on real resource-constrained sensor nodes. For example, it does not consider that most real sensor platforms currently in use have scarce memory, thus the maximum *stretch factor* (i.e., ratio between the number of redundant and original messages), which has been used to study the potential of EC, may not be feasible.

This paper extends the analysis in [1] along the following two directions: (i) it provides an analytical evaluation of the data delivery process to study the sensitiveness of the protocol performance to the parameters' setting and (ii) it analyzes the effect of resource limitations imposed by real sensor devices. Concerning point (i), the major contribution is the development of an analytical model that characterizes the behavior of the overall data delivery process. The proposed model provides a much more flexible (and quick) tool than simulation models. In fact, the complexity of simulation models, due to the high number of events that need to be handle, does not allow to study in detail the system behavior over a large number of parameters. The analytical model is first validated against simulation results to assess its accuracy, and then used to investigate the general properties of the HI protocol.

Regarding point (ii), the major contribution is to analyze the real effectiveness of the proposed hybrid delivery protocol when implemented on real sensor nodes. Indeed, some performance indices are difficult (if not impossible) to measure through analytical or simulation models. An example is the actual consumption of resources, such as percentage of memory or additional energy used for encoding/decoding. Such quantities, which obviously depend on the specific sensor platform considered, must be necessarily assessed by means of an implementation on real sensor nodes. Therefore, to complete the performance analysis of the HI protocol, in this work we also provide a comprehensive analysis on the use of resources by an experimental analysis.

The obtained results show that, when using the HI protocol, the MN is able to decode the original bundle with high probability, even with low values of stretch factor and duty cycle, or with different mobility patterns followed by the MN. These results are further confirmed by the experimental analysis that shows that HI is a feasible solution, despite the very limited storage and processing resources of commercially available sensor platforms, and it introduces a very low energy consumption. In addition, HI is also very appealing for urban sensing scenarios, where more than one MN can be simultaneously in contact with a static sensor at once.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 introduces the system model. Section 4 describes the proposed HI protocol. Section 5 presents the analytical model, while Section 6 provides a careful discussion of the analytical results. Section 7 describes the experimental environment, while Section 8 presents the experimental results. Finally, Section 9 concludes the paper.

## 2. Related work

Many papers in the literature have addressed data delivery in WSNs with mobile nodes [14]. The idea of exploiting mobile nodes for data delivery to save energy at sensor nodes was originally proposed in [24] and refined later in [25]. The use of mobile nodes was applied to different scenarios and has proven to be effective for energy conservation also in the urban context. For example, in [26] shuttle buses were used for collecting data from several static nodes encountered along their paths. In [21], mobile nodes traveling with different speeds were used to collect useful information emitted by a static sensor node along streets.

For improving the reliability of communication, different approaches have been adopted. On one hand, several proposed solutions rely on Automatic Repeat reQuest (ARQ) schemes [27–30]. However, approaches based on retransmissions have several limitations, including a high overhead for loss detection and retransmission, and are unsuitable in environments characterized by high error rate or frequent link failures, and for broadcast communications. Recently, further solutions have been proposed with the specific goal of reducing the unnecessary retransmissions, and, as a consequence, the congestion produced by retransmissions. A work in this direction is TRCCIT [31], where a localized hybrid acknowledgment scheme (HACK) – i.e., a combination of implicit and explicit ACKs – together with a timer management are used to provide reliability in an adaptive fashion. Similarly, in [32] authors proposed the ERTP protocol, where the reliability is controlled at each hop dynamically by adjusting the maximum number of retransmissions. On the other end, other solutions make use of EC for transmitting data. For example, EC has been extensively exploited for increasing reliability in multi-hop sensor networks [23,33–38]. Specifically, [23] compares EC and ARQ by means of experimental analysis which demonstrates a higher reliability reached by EC as it is able to better tolerate packet losses. Similar topics were also discussed in [35], where authors compared EC and ARQ schemes in terms of packet delivery and average energy consumption by means of analytical models. A theoretical analysis of the use of EC is also provided in [36] but, instead of comparing different schemes, authors focused mainly on optimizing EC. Specifically, by applying genetic algorithms, they aimed at finding the optimal number of encoded messages to be transmitted while minimizing the overall cost of transmission and packet loss. Most recently, an hop-by-hop data transfer based on EC, named RDTS, has been proposed [38]. Unlike other proposals, RTDS has the peculiarity of performing EC at each hop, and (to reduce the computational overhead) the partial coding mechanism is applied. A simulation analysis confirms its advantages in terms of energy consumption, traffic overhead, and network lifetime with respect to another EC-based approach. In general, EC schemes have the advantage of mitigating the feedback implosion problem generated by ARQ. However, they should not be applied to congested network, otherwise adding redundancy only worsen the situation. A natural idea is to combine retransmissions and redundancies together, thus, more recently, some works proposed schemes based on such hybrid solution. Specifically, in [39] authors proposed DTSN, a reliable protocol which supports different grades of reliability. The total reliability is achieved by a Selective Repeat ARQ scheme, while partial reliability is provided with the help of EC strategies. Simulation results reveal advantages in terms of throughput (up to a 40% enhancement) and reliability of communications (up to a 90% improvement), while showing significant energy savings. Another hybrid ARQ scheme has been proposed in [40], where authors showed, by analytical and simulation analyses, consistent performance gain in error resiliency, end-to-end latency and energy consumption.

Note that the aforementioned works refer to scenarios with multi-hop *unicast* communications and exploit data redundancy to increase the delivery probability of each single message to the final destination (which is not guaranteed due to intermittent connectivity between nodes). In this paper we focus on single-hop communication, and refer to bundle-oriented applications, where a number of messages have to be reliably delivered to the destination. In addition, we consider both unicast (i.e., single MN) and multicast (i.e., multiple MNs) communications.

Most existing protocols have been evaluated on the basis of simulation and/or theoretical analysis, while very few works provide an experimental performance evaluation on real sensor platforms (see e.g., [23,37,41]). On the contrary, real implementations are needed to have a more realistic assessment of protocol performance. In addition, a practical validation is useful not only to assess the effect of communication efficiency but also of resource constraints imposed by real sensors. Such aspect has not been carefully investigated in previous research works and has been left as open points in [1]. For example, [23] measured the encoding and decoding speed on MICA2 motes, while [41] provides a short discussion on the decoding time only for the Tmote Sky platform. Instead, the evaluation of the actual resource consumption (i.e., memory usage, encoding parameters, energy consumption for the encoding process) is a key target of this paper.

As mentioned in the Introduction, this paper extends our previous simulations analysis of the HI protocol in [1]. In this paper we develop an analytical model of the HI protocol to investigate the sensitiveness to operating parameters such as bundle size, stretch factor, sensor duty cycle, and mobility pattern followed by MN. In addition, we implement the HI protocol on real sensor nodes to investigate, through experimental measurements, its real performance on resource-constrained sensor nodes.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

## 3. System model

In this paper we refer to a specific class of WSN applications, referred to as *bundle-oriented applications* where static sensors produce a limited amount of data. For instance, such data might consist in a detailed snapshot of some quantity of interest (e.g., the level of pollutants in the air) collected during a time-interval (e.g., the current day or the last few hours). The data stored at the sensors are opportunistically delivered to MNs whenever they happen to be in contact. Specifically, data transfer is accomplished by streaming the messages in the bundle to the MNs until all data have been successfully transferred. MNs may consume data on their own, being the final endpoint of the data transfer, or share them with other users, either directly (via interaction with another mobile user), or indirectly (via long-range connection). A realistic application scenario is represented by sensors located in an urban environment (e.g., along streets, at traffic lights, at bus stops) where MNs are represented by people walking or cars moving around the city.

In this paper we address the case of a MN acting as data consumer and we focus on the direct single-hop communication between the (sensor, MN) pair as it is the most challenging. Indeed, while static sensors are resource-constrained, especially in terms of energy, MNs have higher computational resources and no major energy concerns as their battery can be recharged. The peculiarities of the two endpoints make their communications difficult and extremely challenging, especially in highly dynamic mobile scenarios, requiring to be further investigated. Conversely, interactions characterizing the second case (i.e., those between MNs) are similar to those occuring in Delay Tolerant Networks (DTNs) and therefore they can be treated with the same solutions suitable for those environments [42,43]. A different reasoning should be done when sensed data are sent directly to a remote collection point through a long-range communication. Here, the characteristics of the environment pose a different set of communication problems (e.g., handover, admission control issues, use of directional antennas) whose investigations are out of the scope of this paper.

Our reference network model is illustrated in Fig. 1. We assume that the network is sparse enough so that static sensors cannot communicate each other, but they can only communicate with MNs when moving within their transmission range. Specifically, we assume that static sensors can be in contact with one or more MNs at once, i.e., more than one MN can be in the communication range of the sensor node at the same time. Contacts are unpredictable, and MNs are independent, i.e., there is no coordination between them.

Moreover, in such a scenario, contacts occur infrequently and only for a short time. Hence, they should be exploited as much as possible by the communication protocol in order to deliver data efficiently. In detail, the duration of contacts (i.e., the *contact time*) is very limited, especially when the speed of the MNs is high, or sensor nodes use a power management scheme – e.g., a duty-cycle mechanism [21] – to save energy. In addition, data transfer might be affected by a severe message loss, due to the distance and interferences. This is especially true for scenarios where a large number of elements (including MNs) are simultaneously present in the communication range of the sensors.

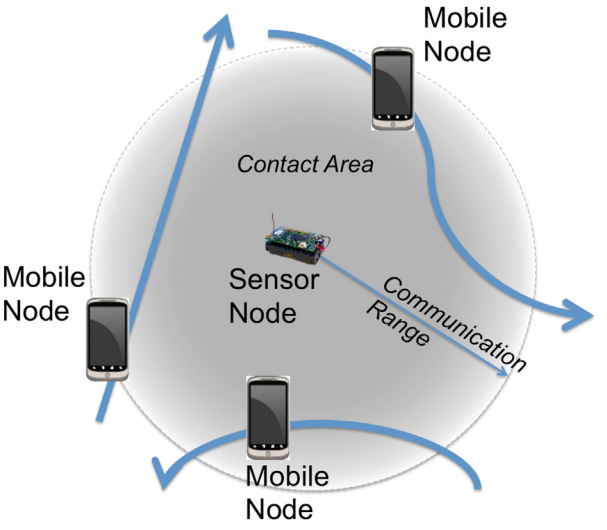


**Fig. 1.** Reference network model.

Another important aspect is that communication between sensor nodes and MNs is possible only when they are in contact, i.e., in the communication range of each other. As the mobility pattern of the MNs is assumed to be random, the static sensor has to: (i) discover the arrival of a MN in the contact area and (ii) detect when the MN has left the contact area, i.e., it is not reachable any more. For MN discovery, low-power discovery protocols can be used [19], typically based on periodic listening [44]. Point (ii) is related to the unknown duration of contacts. Specifically, once a contact has started, the sensor cannot know if a MN is still reachable or not at a given time, unless the MN provides an explicit feedback on its presence in the contact area. In the following we assume that MNs periodically signal their presence in the contact area by sending special *beacon messages* to the static sensors.

## 4. Reliable data delivery

In the scenario described above, the communication is considered successful only if the amount of data available at the sensor is correctly transferred to the MN(s) traveling through the sensor area. To improve the reliability of the data delivery process, the *Hybrid Adaptive Interleaved Communication Protocol* (HI) has been proposed in [1]. The HI protocol is a hybrid communication protocol that relies on a combination of encoding techniques and ARQ schemes. Specifically, in HI the source data is not broadcasted plain but encoded by sensors, i.e., redundant information is added to source data to construct encoded data. Transmissions of encoded data are then regulated through acknowledgments issued by MNs which notify their actual reception state. Finally, MNs start the decoding process

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

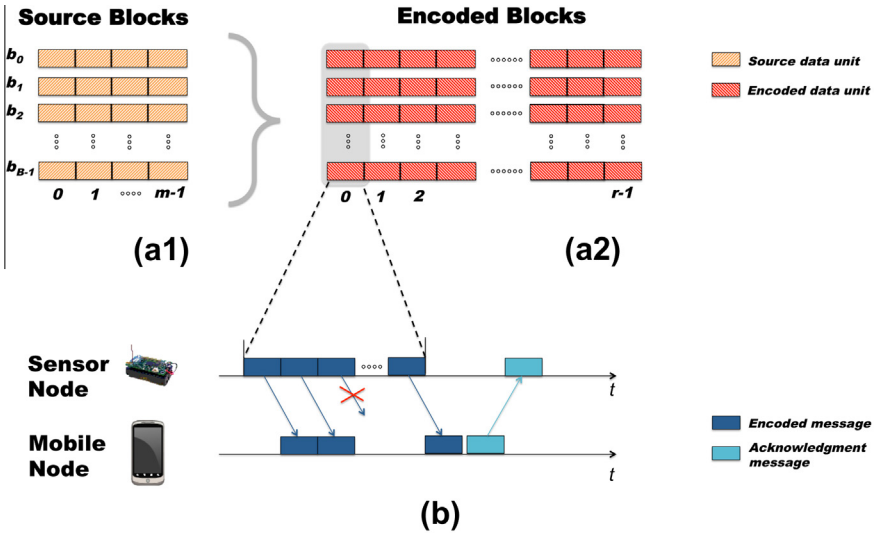


**Fig. 2.** (a and b) Encoding process and (c) HI communication protocol.

when they receive a subset of encoded data so that they may reconstruct the original source data.

In the following we will briefly overview all the three aforementioned phases – namely, *encoding*, *HI communication protocol*, and *decoding* – with a specific focus on sensor-node resource constraints.

### 4.1. Encoding

The *encoding* phase is executed at static sensor nodes, and consists in generating some redundancies to the source bundle before transmission, by applying some EC technique. Specifically, we use Reed-Solomon (RS) codes [45] to encode the source bundle and we incorporate the main optimizations introduced in [46] to reduce its computational complexity, namely (i) computations in the finite fields, (ii) lookup tables, and (iii) systematic codes. Regarding point (iii), assuming that $m$ represents the number of source elements and $r$ the number of encoded elements generated, *systematic codes* are such that encoded elements include a verbatim copy of source elements. Therefore, only $r - m$ elements require encoding. Obviously, the use of systematic codes reduces the computational complexity as well as the memory usage.

Similarly to [46], the source bundle is split into $B$ blocks (i.e., $b_0, b_1, \ldots, b_{B-1}$), each consisting of $m$ data units (see Fig. 2a). This guarantees to keep the $m$ value small, and independent from the bundle size. Each block is then encoded separately, to produce $r$ data units, by applying the RS technique (see Fig. 2b). The ratio between the number of redundant and original messages is named *stretch factor* (i.e., $s_f \triangleq r/m$). Encoding is performed by the sensor node in advance with respect to transmission, whenever the source data are ready. Hence, the sensor node can initiate the communication with a MN as soon as its presence is detected, without having to encode data units on the fly. As a result, the utilization of the (limited) contact time is increased, since no additional encoding overhead occurs during communication.

### 4.2. HI communication protocol

The communication starts when the sensor node has detected the presence of one or more MNs in the contact area. We assume that both sensor nodes and MNs are aware of the encoding parameters, i.e., the number of original messages ($m$) and blocks ($B$) within a bundle, as well as the encoding function.

Reliable communication is accomplished by means of the *Hybrid Adaptive Interleaved Communication Protocol* (HI) [1]. Fig. 2c shows the communication procedure initiated at the sensor side upon discovering the presence of at least one MN. To transmit encoded data units, the sensor node uses an interleaved scheme which consists in scheduling encoded data units picked from distinct (consecutive) blocks rather than sequentially from the same block (the shadow column in the figure). This procedure guarantees more uniform message losses among all blocks, and is independent from the number of blocks and the bundle size. Then, the sensor node encapsulates one encoded data unit into a message of size $b_{msg}$ bytes, and transmits a burst of messages to the MN(s). The MN stores the encoded data units from messages correctly received into its local buffer. In addition, the MN uses the *block identifier* and the *sequence number within the block*[1] from the encoded message header to derive if a sufficient number of messages (i.e., at least $m$ different messages for each block) has been received to decode the original bundle.

Every $T_{ack}$ time, the MN replies with an acknowledgment message[2] (*Ack*) which carries a mask notifying, for each block, how many encoded messages have been correctly received. The sensor node collects all the incoming *Acks* from all MNs that are currently in contact and stores, for each block, the lowest value of messages received

[1] Allowed block identifier values are in the range $[0, B-1]$, while allowed sequence numbers within the block are in the range $[1, r]$.

[2] A contention-based approach is used to avoid collisions between multiple MNs.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

correctly by all the MNs. From the quantities above, the sensor is able to derive for each block if additional data transmissions are required. Specifically, the sensor transmits additional encoded messages for all the blocks for which less than $m$ messages have been received. In order to transmit always fresh and useful encoded messages, the sensor starts from the last message sent but skipping those blocks already completed by all MNs (if any). The process is repeated until the minimum set of encoded messages has been received by all the MNs (i.e., all the block values stored at the sensor node are equal to $m$), or all the MNs are out of the contact area. The latter situation occurs when no *Acks* are received by the sensor within an *end of contact* time interval $T_{eoc}$.

It is worthwhile noting two fundamental features of HI. First, HI is able to dynamically adapt to different levels of message losses experienced by different MNs as number and sequence of transmitted encoded messages are not fixed but variable and depend on loss conditions. Second, *Acks* introduce a very limited overhead as, in such a sparse urban scenario, they are anyway needed as explicit feedback on the MN presence in the contact area.

### 4.3. Decoding

The *decoding* phase is executed at the receiver side (i.e., at the MN) when $m$ distinct encoded messages have been received for each block. The MN decodes the messages and stores the resulting block in its local buffer. Once all $B$ blocks have been correctly decoded, the MN obtains a copy of the original bundle which is thus ready to be used by the application. On the contrary, if the minimum set of encoded messages is not received by the MN, the decoding cannot be performed, and an error message is reported to the application. Note that, for data reconstruction, we adopt similar software optimizations to those used in the encoding process.

## 5. Theoretical analysis

In this section we develop an analytical model for the data delivery phase, and derive the probability for delivering correctly the original bundle to the MN for the scenario depicted in Fig. 3. Without loosing in generality, in our analysis we consider a fixed sensor node and a MN which approaches the sensor by moving at a constant speed $v$ on a linear path at a fixed (vertical) distance from the sensor ($D_y$). Since the MN moves randomly, the sensor is not able to predict when it will enter the contact area. Therefore, the sensor must perform a discovery phase before the communication phase. The discovery process is not instantaneous but it requires a certain amount of time, i.e., the *discovery time* indicated by $d$ in Fig. 3. As a consequence, only a portion of the overall contact time $CT$ can be actually used by the sensor for the data delivery. The contact time yet available for data communication after a MN has been discovered is called *residual contact time*, and is denoted as $rs$ in Fig. 3. During the residual contact time, the sensor broadcasts a source bundle (which has been previously encoded) to the MN.

For the sake of clarity, we divide our analysis in two steps. Initially, we focus on the data delivery process in isolation assuming that it exploits the entire contact time, i.e., it starts at $t = 0$ (Section 5.1). Then, we evaluate the combined effect of the discovery and data delivery processes together, assuming that the first requires a certain amount of time and that the second begins just after the end of the discovery phase (Section 5.2).

The derivation of the analytical model in the latter case is more complex, as it requires the knowledge of the discovery process model. However, deriving the analytical model for the discovery process is beyond the scope of this paper. To this aim, we will consider the analytical model derived in [44] for a similar scenario. Specifically, authors derived the distribution of the discovery time for an asynchronous discovery scheme where the sensor operates with a duty cycle to save energy and the MN periodically sends *beacon messages* to announce its presence (i.e., red rectangles in Fig. 3 represent beacons). Note that the model derived in this section can be applied also to other discovery protocols (e.g., [47]). Therefore, for the model derivation in Section 5.2, we make use of the probability mass function (p.m.f.) of the random variable $D$ denoting the discovery time that has been obtained in [44]. Note that $d(k) = P[D(t) = k]$, where $k$ represents the $k$-th beacon emitted in the contact area[3]. For convenience, we summarize the notation used in the following discussion in Table 1.

### 5.1. Data delivery analysis

In this subsection, before deriving the model of the data delivery process, we introduce the assumptions made in our analysis.[4]

- A1. Time is slotted and each time slot allows the exact transmission of $B$ encoded messages (i.e., an entire column in Fig. 2).
- A2. The transmission of each encoded message takes a fixed duration $T_{msg}$. Consequently, each time slot has a duration $T_B = B \times T_{msg}$. We define $W = \lceil CT/T_B \rceil$ as the number of time slots which fits into the contact time $CT$.
- A3. Message loss is constant during the transmission of an encoded message. This is reasonable due to the short duration of a message transmission. However, different encoded messages are subject to different loss probabilities as message loss changes with time and the distance between MN and the sensor. We assume that message loss follows a Bernoulli distribution, with parameter $p_{i,j}$ ($i \in [0, W-1]$ and $j \in [0, B-1]$) that changes for different messages. Hence, the message of block $b_j$ transmitted in time slot $i$ will suffer a packet loss equal to $p_{i,j}$.

Let $X$ be a random variable (r.v.) denoting the number of encoded messages which are successfully received by the MN during the contact, and let $X_j$ (with $j \in [0, B-1]$) be

[3] Note that beacons are emitted every 100 ms.

[4] The assumptions have been introduced to simplify the presentation and can be easily removed, but this reduces the readability.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

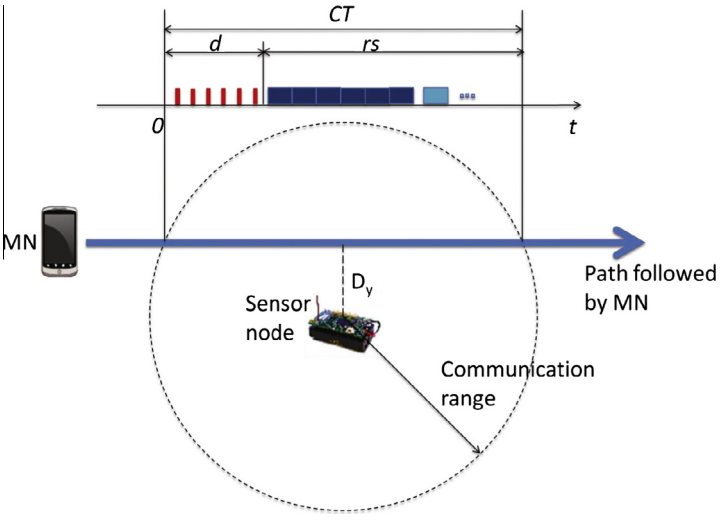


**Fig. 3.** Reference scenario.

**Table 1**
Main symbols used for the theoretical analysis.

| Symbol | Description |
|---|---|
| $m$ | Number of original messages for each block |
| $r$ | Number of encoded messages for each block |
| $B$ | Number of blocks |
| $T_{msg}$ | Time to transmit an encoded message |
| $T_B$ | Time to transmit a window of B encoded messages |
| $W$ | Number of time slots within the contact time |
| $CT$ | Contact time |
| $d$, $D$ | Discovery time and the associated random variable |
| $rs$, $RS$ | Residual contact time and the associated random variable |
| $p_{i,j}$ | Loss probability for the message $j$ at the slot time $i$ |

the r.v. denoting the number of encoded messages successfully received for each block such that $X = X_0 + X_1 + \cdots + X_{B-1}$.

To decode the bundle, a MN has to receive at least $m \cdot B$ distinct encoded messages, and, more specifically, it has to receive at least $m$ distinct encoded messages for each block. Hence:

$$P_{decode} = P[X_0 \geqslant m] \times P[X_1 \geqslant m] \times \ldots \times P[X_{B-1} \geqslant m] = \prod_{j=0}^{B-1} P[X_j \geqslant m] \quad (1)$$

By denoting with $P_{decode_j} = P[X_j \geqslant m]$ the decoding probability for a given row $j$, Eq. (1) can be express as:

$$P_{decode} = \prod_{j=0}^{B-1} P_{decode_j} \quad (2)$$

In the above equation, $P_{decode_j}$ is given by:

$$P_{decode_j} = P[X_j \geqslant m] = \sum_{k=m}^{r} P^*[X_j = k] \quad (3)$$

where $P^*[X_j = k]$ represents the probability of receiving exactly $k$ encoded messages for block $j$ at the end of the contact.

$P^*[X_j = k]$ can be evaluated through a recursive process that starts soon after the end of the discovery process and takes into account all the possible combinations of messages received on all the available time slots. Since we are assuming that the communication phase starts at $t = 0$ (i.e., in the initial slot $i = 0$) and exploits all the W available time slots, $P^*[X_j = k] = P_0^*[X_j = k]$.

The probability $P_0^*[X_j = k]$ can be computed as follows:

$$P_0^*[X_j = k] = \sum_{l_0=0}^{k} \left(P_0[X_j = l_0] \times P_1^*[X_j = k - l_0]\right) \quad (4)$$

i.e., the probability of receiving $k$ messages in $W$ time slots given that $l_0$ messages are received in the first time slot and $k - l_0$ messages are received in the remaining $W - 1$ time slots.

Then, we can express $P_1^*$ in terms of the probability of $P_1$ and $P_2^*$, i.e.:

$$P_1^*[X_j = k_1] = \sum_{l_1=0}^{k_1} \left(P_1[X_j = l_1] \times P_2^*[X_j = k_1 - l_1]\right) \quad (5)$$

By repeating the same procedure for all the time intervals, we obtain the Equation for the last two time slots:

$$P_{W-2}^*[X_j = k_{W-2}] = \sum_{l_{W-2}=0}^{k_{W-2}} \left(P_{W-2}[X_j = l_{W-2}] \times P_{W-1}[X_j = k_{W-2} - l_{W-2}]\right) \quad (6)$$

Finally, Eq. (4) can be recursively solved by taking into account that:

$$P_i[X_j = n] = f(n, p_{i,j}) = \begin{cases} 1 - p_{i,j} & \text{if } n = 1 \\ p_{i,j} & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The above discussion is true in the case of a continuous transmission,[5] i.e., once reached the end of the bundle, the transmission starts from the beginning, thereby taking advantage of all the residual contact time, as assumed by the HI protocol.

An interesting case to analyze is the one in which the bundle is transmitted only once, i.e., once reached the last column, the sensor stops the transmission. In this case we can study the actual percentage of gain that can be achieved through the use of coding techniques in data transmission. Hence, after the transmission of exactly $r$ encoded messages for each block, Eq. (6) becomes:

$$P^*_{r-2}[X_j = k_{r-2}] = \sum_{l_{r-2}=0}^{k_{r-2}} \left(P_{r-2}[X_j = l_{r-2}] \times P_{r-1}[X_j = k_{r-2} - l_{r-2}]\right) \quad (8)$$

### 5.2. Discovery and data delivery analysis

In the previous subsection we have assumed that the data delivery phase begins at $t = 0$. However, as mentioned earlier, in the general case the initial time coincides with the end of the discovery phase. Hereafter, we present the generic model which jointly takes into account the discovery and data delivery phases.

By denoting $D$ the r.v. that represents the discovery time and, consequently, by $RS = CT - D$ the r.v. that represents the residual contact time, the probability of receiving at least $m$ encoded messages for block $j$ can be derived by applying the law of total probability. That is:

$$P_{decode_j} = \sum_{x=0}^{CT} (P[X_j \geqslant m | D = x] \times P[D = x]) \quad (9)$$

where $P[D = x]$ is the probability that the MN is discovered at the instant $x$ (i.e., at the time slot $x = \lfloor D/T_B \rfloor$), whose value can be derived from $d(k)$, while $P[X_j \geqslant m | D = x]$ represents the decoding probability starting from $x$.

Eq. (9) can be rewritten by grouping different contributions according to the ending instant of the discovery phase:

$$P_{decode_j} = \sum_{x \in \mathcal{T}_1} \sum_{k=m}^{r} (P^*[X_j = k | D = x] \times P[D = x]) + \sum_{x \in \mathcal{T}_2} \sum_{k=m}^{r1} (P^*[X_j = k | D = x] \times P[D = x]) \quad (10)$$

where $\mathcal{T}_1 = [0, CT - r \cdot B \cdot T_B]$ and $\mathcal{T}_2 = (CT - r \cdot B \cdot T_B, CT - m \cdot B \cdot T_B]$.

[5] Note that we assume to always transmit all the blocks in the bundle, but if all the MNs within the sensor area receive $m$ distinct messages for a certain block, that block should be skipped. Such assumption has been assumed to simplify the presentation but it can be relaxed with minor changes.

**Table 2**
Parameters for the packet loss model for different values of MN speed (case $D_y$ = 15 m).

| Parameter | $v$ = 3.6 km/h | $v$ = 20 km/h | $v$ = 40 km/h |
|---|---|---|---|
| $a$ | 0.000138 $s^{-2}$ | 0.0028 $s^{-2}$ | 0.0077 $s^{-2}$ |
| $b$ | 0.133 | 0.3828 | 0.4492 |
| $CT$ | 158 s | 30 s | 17 s |

The above equation is composed of two terms: the first takes into account those cases when the residual time is enough to send the whole encoded bundle (i.e., $r$ encoded messages for each block), while the second term considers cases when only a fraction of encoded messages are transmitted in the residual contact time (i.e., $r_1$ with $m \leqslant r_1 < r$).[6]

Eq. (10) can be solved by applying the recursive process to count all the possible combinations of received messages on all (part of) the available time slots during the residual contact time as previously explained. Finally, the decoding probability of the bundle $P_{decode}$ is evaluated by solving Eq. (2) starting from the result of Eq. (10).

## 6. Performance evaluation

In this section we evaluate the performance of the data delivery process referring to the scenario depicted in Fig. 3, and we use the *decoding probability* as our main performance index. The decoding probability ($P_{decode}$) is defined as the probability that a MN can successfully decode the original data bundle (i.e., it correctly receives the minimum amount $m$ of distinct messages for each block in the bundle). A sensitiveness analysis of $P_{decode}$ is then presented. Specifically, we consider the impact of the following parameters: (i) bundle size (i.e., varying both $m$ and $B$ parameters), (ii) stretch factor ($s_f$), (iii) duty cycle at the sensor ($\delta$), and (iv) mobility pattern followed by MN.

To simulate the loss experienced by each message we apply the packet loss model used in [29], which was derived from an extensive experimental analysis carried out in a scenario similar to the one considered here [21]. Specifically, the packet loss is modeled by the following interpolated 2-degree polynomial function:

$$p(t) = \begin{cases} a\left(t - \frac{CT}{2}\right)^2 + b & \text{if } 0 < t < CT \\ 1 & \text{otherwise} \end{cases} \quad (11)$$

where $t$ represents the time elapsed since the initial contact time (i.e., the time when the MN entered the transmission sensing range of the sensor) and $CT$ represents the nominal contact time. Table 2 shows the parameter values, for different speeds of MN and when $D_y$ is equal to 15$m$, that have been derived in [29]. However, as we assume here that time is slotted, we use a discrete-time function $p[n]$ derived from Eq. (11). Specifically, $p[n]$ is obtained

[6] To be precise, Eq. (10) is composed by a third term which considers all the cases in the interval $\mathcal{T}_3 = (CT - m \cdot B \cdot T_B, CT)$. However, its contribution is zero because less than the minimum number of messages needed to decode the bundle (i.e., less than $m \cdot B$ messages) can be received in the residual contact time.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

**Table 3**
Parameters used for the theoretical evaluation.

| Parameter | Value |
|---|---|
| Message payload size ($b_{msg}$) | 89 bytes |
| Total message transmission time ($T_{msg}$) | 17 ms |
| Duty-cycle ($\delta$) | 1%, 5%, 10% |
| MN speed ($v$) | 40 km/h |
| Contact time ($CT$) | 17 s |

by sampling the original continuous-time function with frequency $f = \frac{1}{T_{msg}}$ and starting at $t_0 = \frac{T_{msg}}{2}$. As a result, the following function is obtained:

$$p[n] = p\left(\frac{T_{msg}}{2} + n \cdot T_{msg}\right) \quad \text{with} \quad n \in (0, B \cdot W - 1) \qquad (12)$$

Eq. (12) is aligned with assumption A.3 made in Section 5.1. Indeed, each message experiences a different loss probability but the latter is assumed to be constant during the transmission of each single message with a value equal to the loss experienced at the half of its transmission. Note that this approximation is reasonable and does not affect the accuracy of evaluation as the packet loss difference between the beginning and the end of message transmission is negligible, being at most 0.22%.

In order to verify the accuracy of the analytical model, we also make use of a discrete-event simulator of the HI protocol [1]. Note that the simulation model includes also the discovery process. In all experiments we performed 10 replicas, each consisting of 10,000 contact times. To derive the confidence intervals, we used the independent replication method with a 90% confidence level. However, since the confidence intervals obtained in simulations are very low (less than 1%), they will be omitted for better clarity of the curves.

### 6.1. Model validation

Before presenting the sensitiveness analyzes, we discuss about the validity of the proposed analytical model. Specifically, we focus on the case where the bundle is transmitted only once by the sensor (i.e., once reached the end of the bundle the sensor stops transmitting), hereafter referred to as "Single Bundle Transmission", to study the performance gain of adding redundancy to a data transmission. The values used for the evaluation are shown in Table 3. In addition, Table 4 reports the bundle sizes (in Bytes) considered in the analysis, which have been obtained by varying both the number of messages $m$ and the number of blocks $B$. Specifically, each value is obtained by multiplying: $m \cdot B \cdot b_{msg}$. However, for the sake clarity of figures, we report results only for a set of bundles in the table, and, precisely, those highlighted in gray color. This does not affect the general discussion as all obtained results are aligned. Reported results correspond to bundles with the same number of blocks, while the number of messages for each block is different since we consider $m = 4$ and $m = 8$.

Figs. 4 and 5 show the decoding probability as a function of the stretch factor for $m = 4$ and $m = 8$, respectively. Specifically, we consider $s_f$ in the range [1–10].[7] From the above figures we can see that there is a general agreement in the trend of analytical and simulation results. However, there exists a small difference between them. This is due to a slight discrepancy between the discovery phases used by analytical and simulation approaches. By considering the average discovery time, we note that they are different in the two cases: 3.6 s for the analytical model vs 5.2 s for simulation. The main reason is due to a different modeling of the initial state of the sensor in the two cases. Specifically, in the simulation the initial state of the discovery phase is modeled through a random variable, whereas in the analytical model it is assumed that the sensor always starts from the listening state (see Section 5). Consequently, the discovery time is shorter in the latter case. This difference in the discovery time obviously affects the communication phase. Specifically, the analytical model penalizes smaller bundles, i.e., less than 1780$B$ (see Fig. 4a). In such cases, the transmission starts earlier, i.e., when the packet loss is higher, and spans over an interval with higher packet losses than simulation. As a consequence, the analytical model underestimates the decoding probability of about 10%, on average. Instead, when the bundle size increases, the transmission continues over the minimum of packet loss curve. In such cases, message losses are similar between analysis and simulation, on average. Therefore, although the model again underestimates the probability of decoding, the approximation is reduced. The same line of reasoning can be applied to the $m = 8$ case. Specifically, by looking at Fig. 5, we can see that it exists a small difference between analytical and simulation curves. However, the gap is at most 10%. Apart from this, curves have the same trend.

As this comparison shows an agreement between the simulation and the analysis, it allows us to conclude that the analytical model proposed is accurate and can be used to investigate the performance of the system. For this reason, in the rest of our evaluation study we use the analytical model only.

### 6.2. Impact of the stretch factor

In this section we study the sensitiveness with respect of the stretch factor for the "Single Bundle Transmission" case. Fig. 4 shows that, for $m = 4$, the decoding probability increases with the stretch factor for all the considered bundle sizes, and approaches one for high $s_f$ values. This is intuitive as an increase in the stretch factor corresponds to an increase in the number of redundant messages generated by the sensor and, as a consequence, on a greater number of new information on which the MN can count on in order to decode the original bundle. Also, note that the largest increase in the decoding probability occurs at low values of stretch factor. For example, there is a huge performance gain (i.e., about 40% on average) by increasing $s_f$ from 1 to 3. For $s_f$ value beyond 6 the decoding probability still improves but the performance increase is less apparent. Moreover, Fig. 4 shows that the maximum value is obtained with the highest value considered of redun-

[7] Note that $s_f = 1$ means that no codes are produced and the sensor only transmits the original bundle.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

**Table 4**
Bundle sizes used in the analysis for different values of $m$ and $B$ ($b_{msg}$ = 89 Bytes).

| #B | m = 4 (Bytes) | m = 8 (Bytes) | #B | m = 4 (Bytes) | m = 8 (Bytes) |
|---|---|---|---|---|---|
| 1 | 356 | 712 | 6 | 2136 | 4272 |
| 2 | 712 | 1424 | 7 | 2492 | 4984 |
| 3 | 1068 | 2136 | 8 | 2848 | 5696 |
| 4 | 1424 | 2848 | 9 | 3204 | 6408 |
| 5 | 1780 | 3560 | 10 | 3560 | 7120 |

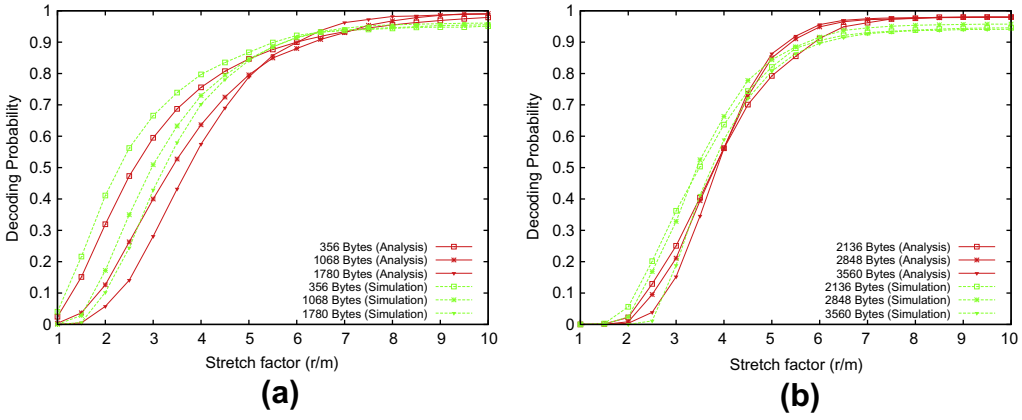


**Fig. 4.** Single bundle transmission: impact of the stretch factor ($m$ = 4).

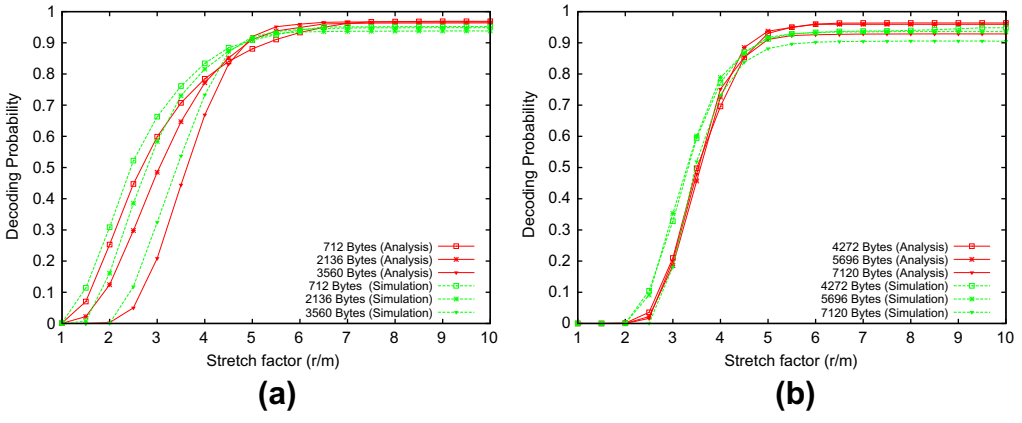


**Fig. 5.** Single bundle transmission: impact of the stretch factor ($m$ = 8).

dancy, i.e., $s_f$ = 10. However, the overall performance is very high even with a lower value of $s_f$: with $s_f$ = 5 the decoding probability is still around 0.8.

Fig. 5 shows the decoding probability for $m$ = 8. Results for this set of bundles are aligned with the previous ones. Specifically, the major decoding probability gain is achieved at low levels of redundancy, beyond which its effect on performance is less apparent. For example, it is possible to decode the original bundle with a 90% of probability with a stretch factor equal to 5. The latter result is very important as it shows that, through a smart use of encoding techniques, a good compromise between performance and consumption can be achieved. Indeed, from the above discussion it derives that by keeping the stretch factor low (i.e., $s_f$ = 5), the performance gain is about 85%, on average.

### 6.3. Impact of the duty cycle

In this section we investigate the impact of the duty-cycle $\delta$ used by the sensor on the performance of the protocol. Fig. 6 shows the decoding probability as a function of $\delta$, when $m$ = 4, for three different bundle sizes. Results for $m$ = 8 are similar and, are thus, omitted. We can observe that, in general, curves are similar to those shown in the previous section for the analysis of the stretch factor. However, the decoding probability changes significantly with the duty cycle. Specifically, very similar performance is

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

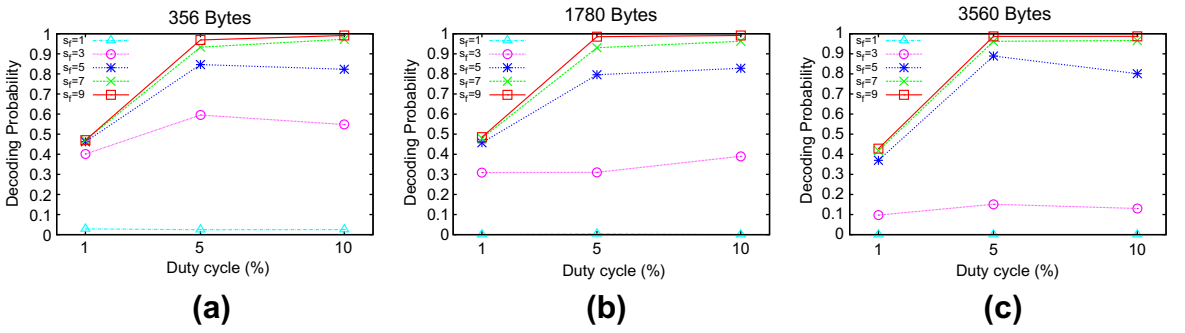


**Fig. 6.** Single bundle transmission: impact of the duty cycle ($m = 4$).

achieved with duty cycles of 10% and 5%, irrespective of the bundle size. The decoding probability exceeds 85% with $s_f \geqslant 5$. Instead, the decoding probability drops below 50% when the duty cycle is equal to 1%. This behavior is essentially due to different discovery times associated with various duty cycles. A high duty cycle means that the sensor stays in the listening state, waiting for the reception of a beacon, for a longer period, thus increasing the MN discovery probability and reducing the discovery time. In contrast, a lower duty cycle implies on average a higher discovery time as the sensor listens for a shorter period. Consequently, since the residual contact time is lower, the sensor is able to transmit – and the MN to receive and decode – for a shorted time interval.

From the above discussion, it follows that the duty cycle plays a key role on performance, and must be carefully chosen. This choice is influenced by two factors: performance and energy consumption. On one hand, the duty cycle must be higher than 1%, otherwise performance remains low, even when using high $s_f$ values. In addition, $\delta > 1\%$, also ensures a high MN discovery probability. For example, we measured a 99% discovery probability, on average, with $\delta \geqslant 5\%$, but the discovery probability is drastically reduced to 55% with $\delta = 1\%$. On the other hand, it is necessary to take into account also the energy consumed in the discovery phase. From this regard, a duty cycle too high could be expensive in terms of energy consumption. For instance, $\delta = 10\%$ could lead to an excessive energy consumption depending for example on the speed or the MN mobility pattern [44]. It follows that when choosing optimal duty cycle a trade-off between power consumption and performance must be reached.

### 6.4. Path-constrained mobility scenario

We conclude our analysis considering different mobility patterns for the MN to show how the proposed analytical model can cope with different MN mobility patterns. So far, we have assumed that the MN moves along a linear trajectory. To generalize our analysis and make the validation of the analytical model more complete and realistic, in this section we assume that the MN follows a curvilinear path. For example, this can be a realistic case of urban mobility where the MN is carried by a user that is constrained to follow a curvilinear path on urban roads. This scenario, more general and certainly more complex than the previously considered one, is depicted in Fig. 7. The MN moves from a point A to a point B, on a curvilinear path (i.e., the blue thick line in figure), within a band of fixed amplitude $X$ and centered on the line whose minimum distance from the sensor is $D_y$. Obviously, the MN moves on a longer trajectory, with respect to the linear distance AB. In addition, assuming that the MN moves at a constant speed equal to that used previously, it will spend more time within the sensing area, thus experiencing a longer contact time (which also affects its decoding probability).

For making the analysis tractable, and still considering a general case, among all the possible paths that the MN may follow, we focus on paths that can be modeled as a sinusoidal function. Under this assumption, the MN trajectory over time can be described through the following equation:

$$y(t) = Y_{max} \cdot sin(\omega \cdot t + \varphi) \tag{13}$$

where $Y_{max}$ represents the *amplitude*, $\omega = \frac{2 \cdot \pi}{T}$ is the *angular frequency*, $T$ is the *period* and $\varphi$ the *phase*. In our case, $Y_{max} = \frac{X}{2}$ since we assume that the MN moves within the band of amplitude $X$. Let us denote by $CT'$ the new contact time (obviously $CT' \geqslant CT$). The period $T$ of the sinusoidal path will be a fraction of $CT'$, i.e., $T = \alpha \cdot CT'$, where $0 < \alpha \leqslant 1$. In addition, the new contact time can be expressed as:

$$CT' = CT \cdot (1 + \varepsilon X) \quad \text{with} \quad \varepsilon \geqslant 0 \tag{14}$$

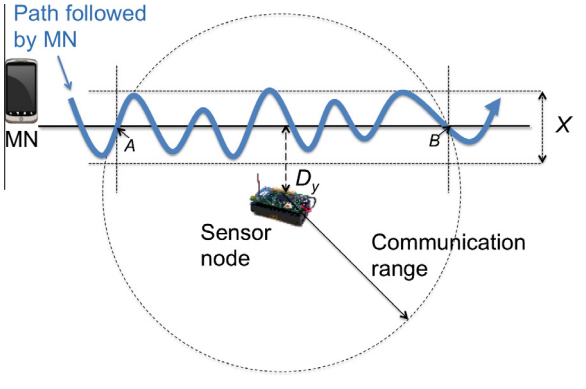


**Fig. 7.** Path-constrained mobility reference scenario.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

Hence, the period $T$ can be determined once $X$, $\alpha$, and $\varepsilon$ have been fixed, i.e.,

$$T = \alpha \cdot CT' = \alpha \cdot CT \cdot (1 + \varepsilon X) \quad (15)$$

Fig. 8 shows some examples of sinusoidal paths, each of which is characterized by a different set of ($X$, $\alpha$, and $\varepsilon$) values (in Fig. 8 we assume $\varphi = 0$ for the sake of clarity). Since time is slotted, we use a discrete-time function $y[n]$, derived by sampling Eq. (13) with frequency $f = \frac{1}{T_{msg}}$ and starting at time $t_0 = \frac{T_{msg}}{2}$, to determine the position of the MN during the transmission of a single message.

Since the communication between the MN and the sensor is affected by packet loss based on their distance, the packet loss model previously used (see Eq. (12)) cannot be directly applied. Therefore, what is needed is a plausible packet loss model that takes into account the new mobility pattern followed by MN. Hence, we derive, for each message transmission, the MN-sensor distance at that time (e.g., $d_P$ in Fig. 9 where the MN is assumed to be in position P along the sinusoidal path), and we calculate the packet loss value associated with that distance by using Eq. (12). This is reasonable because we assume that MN moves at the same speed, and it can be assumed, with a small approximation, that the loss values of points at the same distance are equal. To this end, it may be worthwhile emphasizing that the original packet loss model can be written as a function of time (as shown by Eq. (12)), but also in function of the distance traveled on the linear path, as $t$ and $s$ are linked by $s = v \cdot t$. In addition, it can also be expressed as a function of the actual distance from the sensor ($d$ in Fig. 9) since the following relation among $d$ and $s$ holds:

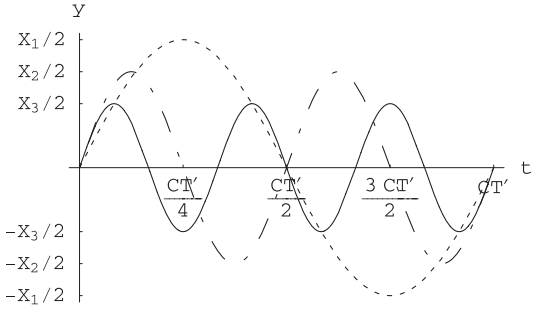


**Fig. 8.** Examples of the behavior of Eq. (13) when varying parameters $Y_{max}$ and $T$ for $\varphi = 0$.

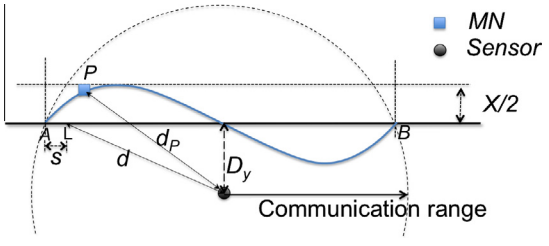


**Fig. 9.** Investigations of distances among points to derive the packet loss associated with the sinusoidal pattern.

$$d(s)\begin{cases} \sqrt{\left(\frac{\overline{AB}}{2} - s\right)^2 + D_y^2} & \text{if } s \leqslant \frac{\overline{AB}}{2} \\ \sqrt{\left(\frac{\overline{AB}}{2} + s\right)^2 + D_y^2} & \text{otherwise} \end{cases} \quad (16)$$

For convenience, let $p(d)$ denote the packet loss at a generic point located ad distance $d$ from the sensor, as derived from Eq. (12). To derive the packet loss of all points on the sinusoidal path ($p'(d_P)$) we repeat the following three steps:

1. For each point P on the sinusoidal path, we measure the corresponding distance $d_P$.
2. Among all the possible distances $d$ for which $p(d)$ is defined (i.e., $D_y \leqslant d \leqslant$ *communication range*), we look for the distance that minimizes the difference $|d - d_P|$. Let denote it by $\bar{d}$.
3. The packet loss value of point $P$ is $p'(d_P) = p(\bar{d})$.

For the performance evaluation we set the key parameters as shown in Table 3 and the duty cycle to 5%. Table 5 shows the settings for the sinusoidal path. Note that the $X$ values shown in Table 5 represent the increase with respect to the original communication range expressed in percentage. For example, $X = 10\%$ corresponds to a band that is approximately 9.5 m being the communication range about 95.16 m.

Fig. 10 shows the decoding probability, when varying the stretch factor, for different $X$ values and when $m = 4$. For each of the three bundles shown in figure, we also show the corresponding curve of Fig. 4 (i.e., labeled with "$X = 0$"), which corresponds to a linear path (i.e., band of zero amplitude). Curve behaviors in Fig. 10 are aligned with those of Fig. 4, that is, a $s_f$ increase corresponds to an increase in the decoding probability. In addition, these results highlight another feature: the decoding probability varies according to the band width. In particular, it is worth noting that all the curves with $X > 0$ are higher than the curve with $X = 0$, corresponding to the case where the MN moves on the linear path. As expected, this result shows that, by increasing the band width (and consequently the contact time), the decoding probability increases too. This aspect is noticeable also for low $X$ values as those shown in the figure, that is, a small increase in the band width and in the contact time guarantees higher performance. Results for $X \geqslant 20\%$ (omitted here for sake of clarity) do not show significant changes in the decoding probability, but are aligned with curves shown in Fig. 10.[8]

Fig. 11a shows the decoding probability as function of the parameter $\varepsilon$ when varying the stretch factor. As example, we show a specific case where the bundle size is set to 1780 bytes, $X$ to 15% and $\alpha$ to 0.5. We can see that the decoding probability increases with $\varepsilon$. Indeed, by increasing $\varepsilon$, the time the MN takes to cross the sensing area and to receive useful data also increases. The increase in the decoding probability is more apparent for intermediate values of stretch factor, while it remains less apparent for

[8] The small difference between curves is because the packet loss of some points on the sinusoidal path has been approximated with values at the interval endpoints on which $p(d)$ is defined, as explained in step 2.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

**Table 5**
Parameters used in the path-constrained mobility scenario.

| Parameter | Value |
|---|---|
| $X$ | 5%, 10%, 15% |
| $\varepsilon$ | 0.5, 1, 1.5 |
| $\alpha$ | 0.125, 0.25, 0.5, 1 |

high values (i.e., $s_f \geqslant 7$). This is because, in the latter case, the decoding probability is already very high (>0.9) and, therefore, a further increase in the contact time does not cause a further performance improvement. Similar results have been obtained by setting different bundle sizes and the other parameters (they are omitted for the sake of space).

The final aspect we analyze is the behavior when changing the value $\alpha$, which affects the period $T$. Fig. 11b shows the decoding probability for an increasing values of $\alpha$, for 1780 bytes, $X = 15\%$ and $\alpha = 0.5$. As apparent from the figure, the curves have a quite constant trend, with very small oscillations. This means that the value of $\alpha$, and consequently of the period, slightly affects decoding probability performance.

## 7. Experimental environment

The purpose of this section is to provide a careful analysis of our HI protocol by means of an experimental evaluation carried on a real testbed. The experimental evaluation becomes essential when we are interested in evaluating the real behavior of a communication protocol. Indeed, even if mathematical modeling and/or simulation analysis are valuable tools allowing to explore a variety of different scenarios, it is well known that they sometimes mask the real behavior of the system [48,49]. This is mainly due to the intrinsic difficulty of modeling some parameters and/or phenomena (e.g., propagation of the wireless signal). On the contrary, many of these approximations are overcome by using experimental measurements (even at a cost of a major effort).

The experimental evaluation is also required when we want to measure the effectiveness of the protocol, in terms of actual consumption of resources, otherwise difficult to be quantified. In our case this is much more important because devices in use have very limited hardware resources. Taking as an example the value of redundancy introduced, in a practical implementation of HI, $s_f$ cannot be excessively large because of the low available memory. As shown in the previous section, a large value of $s_f$ certainly improves the reliability of communication, but also increases encoding and decoding costs, especially in terms of energy consumptions and computational times. For all the above reasons we complemented the analytical evaluation with an experimental evaluation.

In the rest of the section we will describe the experimental testbed, the used methodology, and the performance metrics considered in the evaluation.

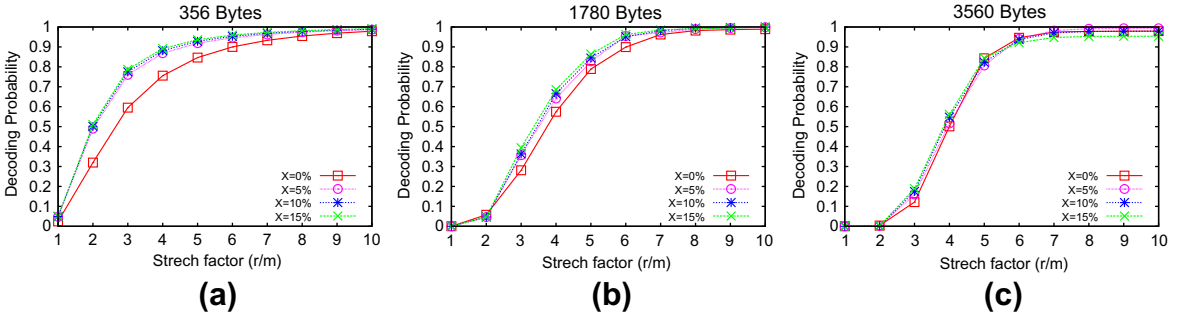


**Fig. 10.** Single bundle transmission: impact of the stretch factor for different band amplitude ($m = 4$, $\varepsilon = 1$ and $\alpha = 0.5$).

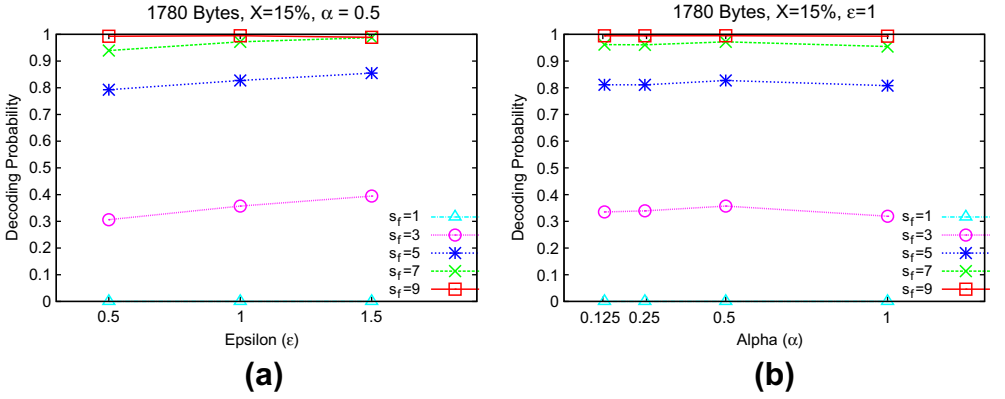


**Fig. 11.** Single bundle transmission: impact of (a) $\varepsilon$ and (b) $\alpha$ for different stretch factor ($m = 4$).

### 7.1. Experimental setup

Our testbed consists of Tmote Sky sensors [50], whose major components are:

- a Texas Instruments MSP430 micro-controller running at a 8 MHz clock, and equipped with 10 KB RAM and 48 KB program memory;
- an IEEE 802.15.4-compliant Chipcon CC2420 Wireless Transceiver, capable of a raw bitrate of 250 kbps over the unlicensed 2.4 GHz frequency band.

The Tmote Sky is also supported by the TinyOS operating system [51], which we used for implementing the reliable data delivery scheme.

The scenario under consideration is the original shown in Fig. 3, where the MN approaches the sensor moving on a linear path. However, here we consider a variable number of MNs (from 1 to 5) that approach the sensor node. The sequence with which MNs join the contact area is random, but it is such that there always exists a non-negligible time interval in which all the MNs are simultaneously within the contact area. Due to the high variability of channel conditions, obtaining comparable experiments (i.e., with the same statistics) is quite difficult. Thus, to guarantee the replicability of experiments we emulated the message loss due to mobility by using the packet loss model introduced in Section 6 and expressed by Eq. (12). As we focus on the high mobility scenario (i.e., a 40 km/h MN speed), the correspondent contact time between the sensor and the MNs is about 17 s.

In each experiment, a bundle of a given size is first encoded and then broadcasted by the static sensor to the MNs in the contact area. Each experiment is replicated 100 times, and the results are averaged over all the replicas (the standard deviations are also provided as error bars). Table 6 shows the parameter settings used in this experimental evaluation. Note that the power consumption values are derived in accordance with the data sheet values of the Chipcon CC2420 radio transceiver [52]. The remaining parameters are set as in the previous section (see Table 3).

### 7.2. Performance metrics

We evaluate our hybrid protocol both in terms of performance and resource utilization. To evaluate the performance we consider the *decoding probability* (introduced before) along with the *communication energy* ($E_c$). The communication energy is defined as the total average energy consumed by a sensor node per each byte correctly transferred to the MN. It can be derived from the following Equation:

$$E_c = \frac{(k \cdot T_{msg} \cdot P_{tx}) + \left(N \cdot \left\lceil \frac{k \cdot T_{msg}}{T_{ack}} \right\rceil \cdot T_{msg} \cdot P_{rx}\right)}{\hat{b}_{tot}} \quad (17)$$

where $k$ is the total number of messages transmitted by the sensor node; $T_{msg}$ is the message transmission time [9]; $P_{tx}$ and $P_{rx}$ are the power consumption of the radio in transmit and receive state, respectively; $T_{ack}$ is the time interval between two consecutive acknowledgments sent by the same MN; $N$ is the number of MNs considered in the experiment; finally, $\hat{b}_{tot}$ is the total number of bytes decoded by all the MNs. In Eq. (17), the term $\left\lceil \frac{k \cdot T_{msg}}{T_{ack}} \right\rceil$ represents the total number of acknowledgments received by the sensor node from each mobile node. In addition, note that the equation takes into account also the energy consumed for transmitting packets not successfully received. Specifically, the first term of the numerator captures all the packets that the sensor transmits, independently of their reception at the MN side.

**Table 6**
Parameters used for the experimental evaluation.

| Parameter | Value |
|---|---|
| Message (payload) size ($b_{msg}$) | 89 bytes |
| Frame size | 128 bytes |
| Acknowledgment period ($T_{ack}$) | $16 \cdot T_{msg}$ |
| Beacon period | 100 ms |
| End of contact timeout ($T_{eoc}$) | $8 \cdot T_{ack}$ |
| Duty-cycle ($\delta$) | 5% |
| Transmission power at 0 dBm ($P_{tx}$) | 52.2 mW |
| Receive power ($P_{rx}$) | 56.4 mW |
| CPU power when the radio is off ($P_e$) | 5.4 mW |

To evaluate the resource utilization, we consider the following metrics:

- *Memory usage*: the total amount of RAM, expressed as percentage with respect to the total available RAM (i.e., 10 KB), required by HI both for encoding and communication.
- *Encoding time*: the amount of time needed to encode a bundle of a given size at the sensor side.
- *Decoding time*: the amount of time needed to decode the bundle[10] at the MN.
- *Encoding energy*: the average energy consumed by a sensor node per each encoded byte.

Specifically, the encoding energy is obtained as:

$$E_e = s_f \cdot \frac{T_{code} \cdot P_e}{b_{msg}} \quad (18)$$

where $s_f \triangleq r/m; T_{code}$ is the average time required to produce an encoded data unit; $P_e$ is the power consumed by the sensor during the encoding phase (i.e., the energy consumed by the CPU); finally, $b_{msg}$ is the size (in bytes) of the message payload.

## 8. Experimental results

In this section we evaluate the behavior of our hybrid protocol. First, we focus on the analysis of resource

[9] We assume that the time to transmit data and acknowledgment messages is equal.

[10] The decoding time is computed only for the MNs which have correctly received the entire bundle. For comparison purposes, we assume that the MNs is using the same hardware platform and the software implementation of the reliable data delivery scheme as the static sensor.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

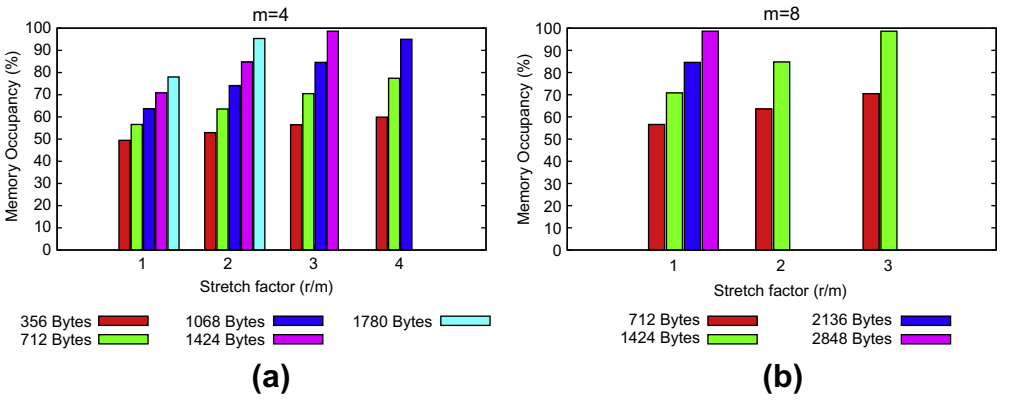


**Fig. 12.** Memory usage for (a) $m = 4$ and (b) $m = 8$.

utilization by showing the set of allowable values for various parameters, and their actual consumption for a real implementation. Then, we analyze the communication performance of the HI protocol for the range of allowable values.

### 8.1. Resource utilization

We start our analysis by considering the resource utilization in terms of used memory, since the amount of RAM represents the major limiting factor for both the bundle size and the encoding parameters. Then, we investigate the time needed to encode (decode) each bundle, as well as the energy consumed during the encoding process, since they strictly depend on the specific encoding parameters. All the above quantities are evaluated for different bundle sizes and redundancy levels. Specifically, we consider $m = 4$ and $m = 8$ as in the analytical evaluation, but we limit $s_f$ in the range [1–4]. Indeed, the small amount of available memory does not allow to encode very large bundles or to use redundancy levels (i.e., $s_f$ values) higher than those used in Section 6.

Fig. 12 illustrates the percentages of memory utilization at the sensor node for $m = 4$ and $m = 8$. It is apparent that a real implementation limits the maximum amount of data that can be stored and encoded. In fact, in a real implementation, the maximum size of the bundle is significantly reduced with respect to the size of bundles that we have considered in the previous section (see Table 4)[11]. Bundle sizes larger than those shown in Fig. 12 cannot be encoded because sensors have not enough memory to store all the needed information. As expected, the memory usage depends both on the size of the bundle and on the level of redundancy introduced. The higher the bundle size and the stretch factor, the higher is the memory usage.

When $m = 4$ (see Fig. 12a), encoding the smallest bundle (i.e., 356 bytes corresponding to about 3.5% of the RAM) with $s_f = 1$ consumes half of the available RAM in the system, so that the stretch factor can be increased, at most, up to 4. On the contrary, the largest bundle which can be encoded with $s_f = 1$ is of 1780 bytes. In addition, encoding a bundle of 1424 bytes (13.5% of the available RAM) with $s_f = 3$ almost uses all the memory. Therefore, the bundle size must be further reduced until approximately 1 KB when using $s_f = 4$.

This large increase in the memory usage can be explained by looking at how the available memory is used by the different components of the HI protocol. Fig. 13 breaks down the memory usage according to four different factors. Specifically, the used RAM mainly depends on the buffer size for storing the original bundle and the encoding structures (i.e., the encoding matrix, the lookup tables, and the encoded bundle). Additional data structures are used by the communication protocol. Finally, part of the RAM is used by the variables and the data structures required by the operating system (e.g., the queues used for data transmission and reception, timers). As highlighted in Fig. 13, most of the available RAM (i.e., 10 KB) is used by the operating system structures, which account for a 35% share of the RAM, irrespective of the encoding parameters and the bundle size. A different contribution, which is constant, is represented by data structures of the communication protocol. However, this factor has a very small impact on memory usage (i.e., less than 3%). In contrast with the two factors already discussed, the encoding contribution is variable and strictly dependent on the bundle size and stretch factor. It is up to 20% of the total RAM size for the smallest bundle, while it exceeds 40% for the following (*bundle size, stretch factor*) pairs: (1780,2), (1424,3), (1068,4). In all the other cases, the encoding contribution falls in the range (20%, 35%). More specifically, for a given bundle size and stretch factor, the amount of memory required by the encoding process can be derived as:

$$M_{tot} = \alpha + s_f \cdot b_{tot} \tag{19}$$

where $\alpha$ is a constant value representing the total amount of bytes required for the encoding matrix, the lookup tables, and a set of variables. This confirms the linear growth of the encoding contribution.

[11] Note that, for a fair comparison, in Section 6 we used 10 KB as reference value for the memory.

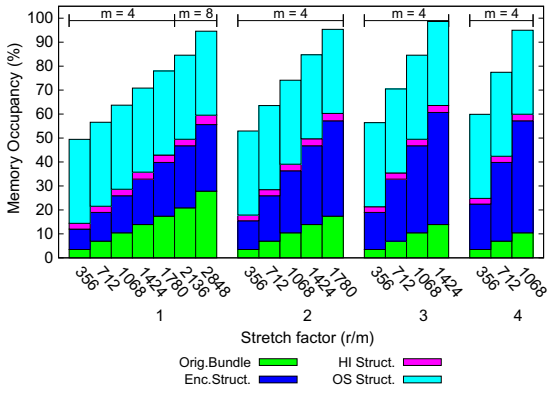


**Fig. 13.** Breakdown of the memory usage for different bundle size (expressed in bytes).

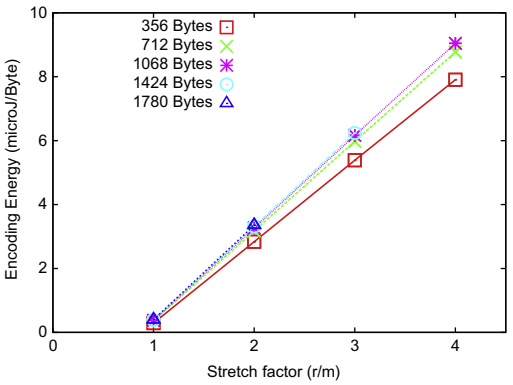


**Fig. 15.** Encoding energy as a function of $s_f$ for different bundle sizes where $m = 4$.

We also considered the case $m = 8$ (see Fig. 12b). The increase in the number of source data units results in a decrease for both the maximum bundle size and the stretch factor that can be used by the sensor. Fig. 12b highlights that it is possible to increase the stretch factor up to 3 for bundle sizes smaller than 1.5 KB; for higher bundle sizes, the maximum stretch factor is limited to 1, as more than 60% of the RAM is used for all the data structures required by the reliable data delivery scheme.

Fig. 14a shows the encoding time for $m = 4$. As expected, the encoding time increases linearly with the stretch factor. Note that the encoding process takes less time when the stretch factor is equal to 1, requiring less than 200 ms for the largest bundle. This is due to the use of systematic codes in the encoding process. This means that the first $m$ data units are just copied in memory, requiring about 6 ms each. On the contrary, the production of additional data units takes more time due to the encoding operations such as the matrix-by-vector multiplication. Specifically, we measured about 42 ms, on average, to generate each additional code (i.e., one order of magnitude higher than the simple memory copy). In addition it is worth pointing out that, since the time for generating encoded data units is higher than the average transmission time of a packet (17 ms), it is not possible to encode data units on the fly. This confirms the effectiveness of the approach used in HI, where the entire bundle is encoded in advance, without consuming the limited contact time.

Fig. 14b shows the decoding time for $m = 4$, which increases linearly for stretch factors up to 2 and, then, remains almost constant. Since systematic codes are used, the decoding time strongly depends on how many (copies of the) original messages have been received at the MN side.

We have found that in the considered scenario, a significant percentage (i.e., 40–50%) of the received encoded messages consists in a copy of the original data unit. Overall, the decoding phase remains in the order of hundreds of msec – about 100 ms for the smallest bundle, and about 700 ms for the largest one. Hence, the decoding delay is negligible if compared with the time needed to actually transfer the bundle (i.e., up to 3 s for the largest bundle). The results also confirm that, as expected, the decoding phase is faster than the encoding phase, requiring about half of the time needed for encoding the bundle.

To conclude our analysis about resource utilization, Fig. 15 shows the average energy spent to encode the bun-

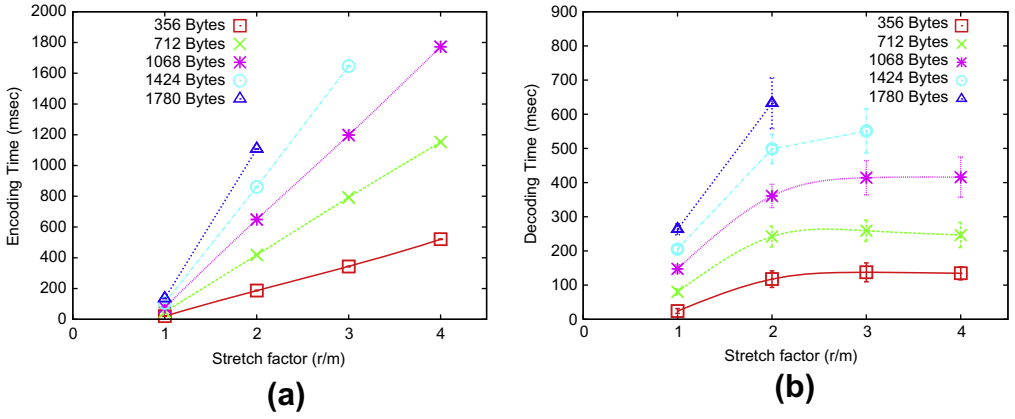


**Fig. 14.** (a) Encoding time and (b) decoding time as a function of $s_f$ for different bundle sizes where $m = 4$.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

**Table 7**
Encoding time, decoding time, and encoding energy as a function of $s_f$ for $m = 8$.

| Bundle size (bytes) | Encoding time (ms) | | | Decoding time (ms) | | | Encoding energy (μJ/bytes) | | |
|---|---|---|---|---|---|---|---|---|---|
| | $s_f = 1$ | $s_f = 2$ | $s_f = 3$ | $s_f = 1$ | $s_f = 2$ | $s_f = 3$ | $s_f = 1$ | $s_f = 2$ | $s_f = 3$ |
| 712 | 41.2805 | 718.652 | 1395.08 | 66.6473 | 399.361 | 411.859 | 0.302474 | 5.28353 | 10.2673 |
| 1424 | 99.9418 | 1608.92 | 2718.64 | 193.813 | 846.286 | 1068.12 | 0.375482 | 5.93352 | 8.66746 |
| 2136 | 163.219 | n.a. | n.a. | 332.427 | n.a. | n.a. | 0.399871 | n.a. | n.a. |
| 2848 | 223.873 | n.a. | n.a. | 432.233 | n.a. | n.a. | 0.411727 | n.a. | n.a. |

dle. The trend of the curve is linear, according to Eq. (18). Curves are almost overlapped, with a slight difference only for values of $s_f$ higher than 3. What is important to highlight here is that the energy consumption for encoding bytes is very limited and low, and requires only few micro Joules. This is fundamental in order to save energy at the sensor side.

Finally, Table 7 provides the encoding time, the decoding time, and the energy consumed for encoding the bundle when $m = 8$. For clarity, the label "n.a." (i.e., not allowed) contained in some cells means that, for that specific (bundle size, $s_f$) pair, there is not enough of RAM to store all the different components and, consequently, to encode the given bundle. As shown in both tables, the investigated metrics are aligned with the results obtained for the case $m = 4$. The encoding time is limited to a few milliseconds in the case of small bundles, while it is of 2.7 s for the 1424 bytes bundle and $s_f = 3$. The decoding time is approximately one half than the encoding time, and the energy spent to encode the bundle is still of a few micro Joules.

### 8.2. Communication performance

In this section we evaluate the reliability and the energy efficiency of the communication protocol. The analysis is split into two parts. In the first part, we assume that the bundle is transmitted just once by the sensor ("Single Bundle Transmission" case) to compare experimental results with analytical and simulation results presented in Section 6, while in the second part we focus on the original HI protocol (see Section 4.2), which provides a continuous transmission of the bundle, as long as all the MNs have completed decoding ("Continuous Bundle Transmission" case).

#### 8.2.1. Single bundle transmission

The "Single Bundle Transmission" case represents the most simple transmission scenario, where the sensor stops sending data once it has sent the encoded message in position $(B-1,\ r-1)$. This scenario is useful to show the achieved decoding probability gain when encoding techniques are applied to a single transmission of the bundle.

Fig. 16a and b shows the decoding probability and the energy spent for communications, respectively, when $m = 4$ and when only one MN crosses the sensing area. The results show two important features. First, they confirm that, by increasing the stretch factor, the MN increases its ability to receive and correctly decode the data bundle. Obviously, the energy spent increases as well, because the sensor sends a greater number of encoded messages. However, as shown by Fig. 16b, it is very low remaining below 50 μJ/Bytes. In addition, the decoding probability never reaches 1 (or at least 0.9), regardless of the redundancy (i.e., the stretch factor) and bundle size. For example, it does not exceed 0.7 for 356 bytes at the maximum allowed stretch factor ($s_f = 4$), or 0.4 for 1068 bytes, dropping drastically to less than 0.1 for 1424 bytes. This highlights that, in general, encoding techniques are advantageous for improving performance, but they are not always able to ensure satisfactory results, if used alone. This clearly appears in implementations on real sensor nodes with limited resources. A possible way to improve performance is thus to use encoding techniques in conjunction with transmission strategies that provide smart multiple transmissions of the bundle (as we will show in the next section).

Finally, it is interesting to compare these experimental results with the analytical and simulation ones presented in Section 6.2 (obtained under the "Single Bundle Transmission" assumption, as well). By comparing curves of Fig. 16a with the corresponding ones in Fig. 4a, we see that results are almost aligned. Obviously, there is a small discrepancy in the corresponding values. This is mainly due to the fact that experimental results are highly variable, being subject to a number of external factors that cannot be controlled (e.g., noise on the channel, position of the antennas), as proved by the high confidence intervals (see Fig. 16a).

#### 8.2.2. Continuous bundle transmission

In contrast with "Single Bundle Transmission", the "Continuous Bundle Transmission" case implements the original HI communication algorithm where the bundle is transmitted multiple times during the contact time, i.e., once the message in position $(B-1,\ r-1)$ has been transmitted, the sensor starts the transmission again from message (0,0). This has two major side effects. First, the stretch factor can be significantly reduced with respect to the $s_f$ values used in Section 6 (but also in Section 8.2.1), thus resulting in a lower resource utilization. In addition, a continuous bundle transmission allows to decouple the sensor transmission from the number of MNs, and from the instants they enter in the contact area, thus increasing the overall system performance.

Figs. 17 and 18 show the decoding probability and the energy spent for communications, respectively, when $m = 4$.[12]

[12] We omit the discussion of results for the case $m = 8$ mainly for the sake of space, and also because they are aligned with the results obtained for the case $m = 4$.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

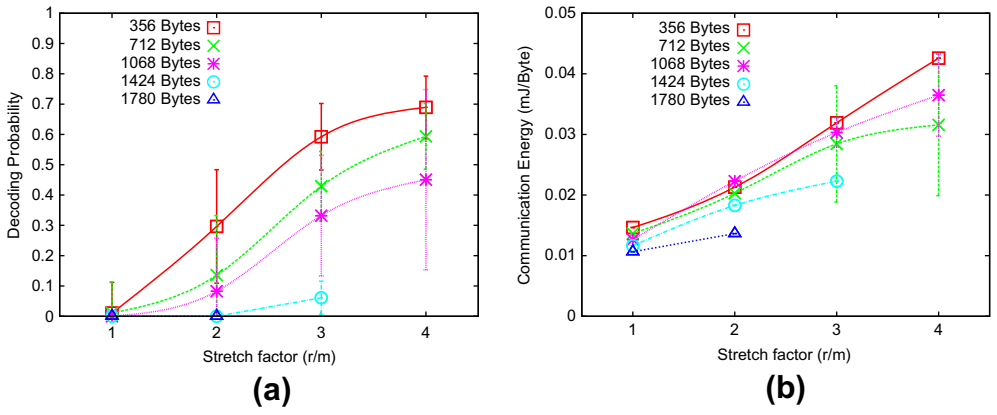


**Fig. 16.** Single bundle transmission: decoding probability and communication energy as a function of $s_f$ for one mobile node for $m = 4$.

By looking at results in Fig. 17, we can see that, as $s_f$ increases, the probability of receiving different but useful piece of information to reconstruct the original bundle also increases, resulting in an higher decoding probability, as expected. This is more apparent for larger values of the bundle size (i.e., $\geqslant$1424 bytes). On the contrary, the highest value of decoding probability (about 1) is reached with $s_f = 1$ for smaller bundles (i.e., <1424 bytes). Obviously these results are better than those shown in Fig. 16a since the bundle is now transmitted several times. We measured that, for small bundles, two complete bundle transmissions are needed, on average, to guarantee the correct decoding. Furthermore, by comparing results shown in Figs. 16a and 17a, we can measure the probability gain obtained by HI thanks to its continuous-bundle-transmission strategy: up to 90% for bundles smaller or equal to 1068 bytes and up to 70% for bundles larger than 1068 bytes. This result gives a precise technical indication for setting the stretch factor. $s_f$ can be kept to the minimum value with bundles size of maximum 1424 bytes, thus saving resources, since the transmission used by HI is sufficient to achieve the maximum performance. Instead, a low $s_f$ value is not sufficient for higher bundle size, for which the redundancy level must be increased.

The results in Fig. 17a can also be compared with those shown in Table 8. The latter have been obtained through using the analytical model presented in Section 5 in the case of a continuous bundle transmission. Note that we have captured this behavior by modeling it with equation Eq. (6). Table 8 shows the asymptotic behavior of the decoding probability. Experimental results are in line with the analytical ones. The small difference in some points is due to a greater variability of the experimental results. This comparison highlights that, although the proposed analytical model introduces some simplifying assumptions with respect to the HI protocol (i.e., a block transmission should be skipped if all the MNs have received enough data to decode that block), it is able to capture the real behavior of HI protocol with a good approximation.

Fig. 17 shows another important advantage of jointly using HI and $s_f > 1$: better performance is obtained even when more than one MN are within the sensor contact area. This is well highlighted by Fig. 17b and c. For example, the sensor node is able to transfer bundles of large size (i.e., 1424 bytes) with a decoding probability close to one with five MNs, while the decoding probability is almost 0.8 when a single MN is present. The above behavior can be explained by considering two main reasons: (i) the stretch factor used for encoding the bundle and (ii) the weight of the discovery phase on the residual contact time of each MN in the different experiments.

In the case of $s_f = 1$, since $r = m = 4$, each MN needs to receive exactly 4 out of the 4 sent messages for each block to decode the entire bundle. In other words, the MN must receive *all* the sent messages. Clearly, by increasing the bundle size and the number of MNs, the probability that *each* MN receives *all* the four messages decreases. This explains why curves in Fig. 17c are lower than those of Fig. 17a and b for $s_f = 1$. By increasing $s_f$ the number of encoded messages sent for each block increases too. As a consequence, the number of encoded messages that potentially can be used by *all* the MNs increases as well. For example, if $s_f = 2$,$r$ is equal to 8, i.e., 8 different messages are sent for each block, and each MN must receive at least 4 (out of 8) different messages for each block, thus increasing its chance to decode the bundle as also confirmed by figures.

Furthermore, the discovery phase and its weight on the residual contact time (i.e., the useful time for the bundle transmission) are additional factors for the decoding probability increase. The weight of the discovery phase decreases when the number of MNs increases. Indeed, the discovery phase, which is always executed to detect the presence of the first MN entering in the sensor area, is not performed by the remaining MNs with increasing probability. Indeed, the discovery of MNs arriving after the first one is almost immediate as the sensor node is already active. This means that these MNs will start receiving data immediately. Thus, by exploiting on average a higher residual contact time, they will have more chance to complete the decoding, with a consequent increase of the overall decoding probability.

As for the communication energy (see Figs. 18), increasing the bundle size decreases the energy consumption, which is less than 0.2 mJ/byte for bundle sizes larger than 1 KB. In addition, the communication energy tends to

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

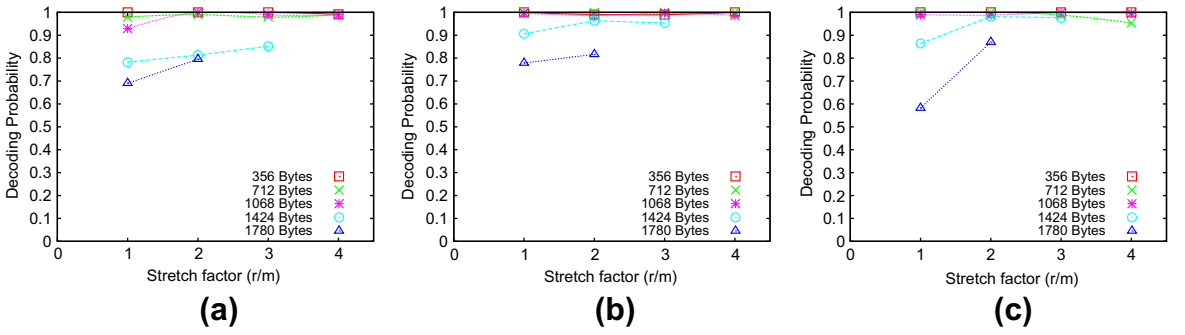


**Fig. 17.** Decoding probability as a function of $s_f$ for (a) one mobile node, (b) 3 mobile nodes, and (c) 5 mobile nodes for $m = 4$.

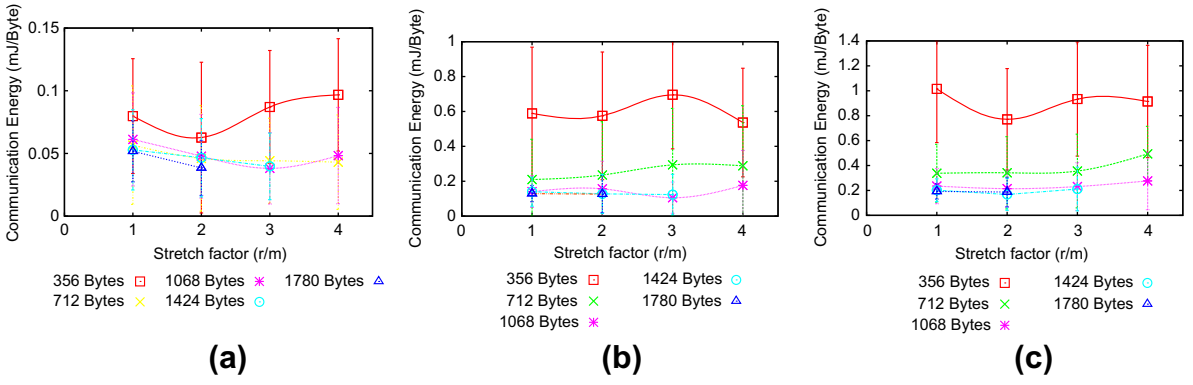


**Fig. 18.** Communication energy as a function of $s_f$ for (a) one mobile node, (b) 3 mobile nodes, and (c) 5 mobile nodes for $m = 4$.

**Table 8**
Decoding probability obtained from the theoretical evaluation for one mobile node for $m = 4$.

| Bundle size (bytes) | Decoding probability |
|---|---|
| 356 | 0.999 |
| 712 | 0.998 |
| 1068 | 0.955 |
| 1424 | 0.899 |
| 1780 | 0.776 |

decrease with the stretch factor. Apparently, in some cases the communication energy actually decreases when the stretch factor increases (see also Fig. 18c for $s_f = 4$). This is mainly because real experiments exhibit a very large variability due to the limited number of replications.

A complete evaluation of the total amount of energy consumed by the system should also include the encoding energy. As we already noted in Section 8.1, the energy consumption due to encoding is limited to few micro Joules (i.e., less than 10 μJ/Byte for $s_f = 4$). Hence, this contribution can be ignored, as it is at least one order of magnitude lower than the minimal energy consumed for the communications (i.e., 0.2 mJ/Byte). On the contrary, in case of single MN within the sensor contact area, the encoding contribution is negligible only when $s_f < 4$. However, using a lower stretch factor does not affect system performance as the decoding probability still remains a high value. As a consequence, it is convenient to spend some processing energy to improve the communication efficiency, thus justifying the EC approach we used.

## 9. Conclusions

In this paper we have investigated a reliable data delivery scheme for sparse Wireless Sensor Networks (WSNs) with Mobile Nodes (MNs). The proposed solution, named *Hybrid Adaptive Interleaved Communication Protocol* (HI), exploits a hybrid approach where both Erasure Coding (EC) and retransmissions are used. The main contributions of this paper include: (i) an analytical model to derive the performance of the overall data delivery process and (ii) an experimental evaluation carried on with real sensor platforms which complements the analytical evaluation. By exploiting the developed analytical model we carefully studied the general properties of the HI protocol with respect to a number of parameters (e.g., bundle size, stretch factor, sensor duty cycle, mobility pattern of the MN). In addition, through the real prototype, we focused on those parameters which cannot be evaluated with an analytical model, such as resource utilization and energy consumption at sensor nodes.

Our results show that the use of encoding techniques greatly improve the reliability of communication. For example, in the case of the single transmission of a burst of data, the use of stretch factor equal to 5 allows more than 80%

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

performance gain. Furthermore, results show that the sensor duty cycle plays a central role. It must be kept low to save energy, but higher than 1%, otherwise even a high stretch factor would not be enough to balance the correspondent performance reduction. In addition, we have shown that, by changing the mobility patterns followed by the MN (for example, from a linear one to a curvilinear one), MN is still able to decode the transmitted bundle with high probability. Furthermore, we have shown that, despite the very limited computational and memory resources on the sensor side, the HI protocol is feasible for a real implementation, results in a high probability of data delivery, and it is also particularly suitable to scenarios where more than one MN are in contact with static sensors at the same time. Finally, our findings provide also indications of the optimal stretch factor to be used with respect to the size of the bundle.

In this paper we have considered sensor nodes with very limited resources. Note that the proposed communication protocol can be successfully used with more powerful sensor platforms (i.e., Jennic, Sun Spot), hence guaranteeing the transmission of larger bundle size. Moreover, we have considered Reed Solomon codes as EC technique because they are optimal for the scenario considered. We leave the evaluation of different EC schemes, as well as model extensions to include more generic multiple MNs, for future work.

## References

[1] G. Anastasi, E. Borgia, M. Conti, E. Gregori, A hybrid adaptive protocol for reliable data delivery in WSNS with multiple mobile sinks, The Computer Journal 54 (2) (2011) 213–229. doi:http://dx.doi.org/10.1093/comjn/bxq038.
[2] M. Conti, S. Chong, S. Fdida, W. Jia, H. Karl, Y.-D. Lin, P. Mhnen, M. Maier, R. Molva, S. Uhlig, M. Zukerman, Research challenges towards the Future Internet, Computer Communications 34 (18) (2011) 2115–2134.
[3] M. Conti, S.K. Das, C. Bisdikian, M. Kumar, L.M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, F. Zambonelli, Looking ahead in pervasive computing: challenges and opportunities in the era of cyber-physical convergence, Pervasive and Mobile Computing 8 (1) (2012) 2–21.
[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.
[5] I. Akyildiz, M.C. Vuran, Wireless Sensor Networks, John Wiley & Sons, Inc., New York, NY, USA, 2010.
[6] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, Mobile Ad hoc Networking: The Cutting Edge Technologies., IEEE Press and John Wiley and Sons, Inc., New York, NY, USA, 2013.
[7] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transaction on Wireless Communications 11 (4) (2009) 660–670.
[8] O. Younis, S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Transactions on Mobile Computing 3 (4) (2004) 366–379.
[9] H. Frey, I. Stojmenovic, Handbook of Sensor Networks: Algorithms and Architectures, John Wiley & Sons Inc., 2005 (Ch. Geographic and Energy-Aware Routing in Sensor Networks, pp. 381–416).
[10] H. Kalosha, A. Nayak, S. Rhrup, I. Stojmenovic, Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks, in: Proceedings of IEEE INFOCOM, 2008, pp. 346–350.
[11] C.K. Rupani, T.C. Aseri, An improved transport layer protocol for wireless sensor networks, Computer Communications 34 (6) (2011) 758–764.
[12] R.A. Mini, A.A. Loureiro, Energy-efficient design of wireless sensor networks based on finite energy budget, Computer Communications 35 (14) (2012) 1736–1748.
[13] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer Networks 52 (12) (2008) 2292–2330.
[14] M. Di Francesco, S.K. Das, G. Anastasi, Data collection in wireless sensor networks with mobile elements: a survey, ACM Transactions on Sensor Networks 8 (1) (2011) 7:1–7:31.
[15] C. Konstantopoulos, P. Bellavista, C.-F. Huan, D. Turgut, Special issue: reactive sensor networks, Computer Communications 36 (9) (2013) 963–964.
[16] X. Li, H. Huang, X. Yu, W. Shu, M. Li, M.-Y. Wu, A new paradigm for urban surveillance with vehicular sensor networks, Computer Communications 34 (10) (2011) 1159–1168.
[17] J. Chen, H. Frey, X. Li, Special issue: wireless sensor and robot networks: algorithms and experiments, Computer Communications 35 (9) (2012). iii–iv.
[18] P. Serrano, A. de la Oliva, P. Patras, V. Mancuso, A. Banchs, Greening wireless communications: status and future directions, Computer Communications 35 (14) (2012) 1651–1661.
[19] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Networks 7 (3) (2009) 537–568. doi:http://dx.doi.org/10.1016/j.adhoc.2008.06.003.
[20] M.I. Khan, W.N. Gansterer, G. Haring, Static vs. mobile sink: the influence of basic parameters on energy efficiency in wireless sensor networks, Computer Communications 36 (9) (2013) 965–978.
[21] G. Anastasi, M. Conti, E. Gregori, C. Spagoni, G. Valente, Motes sensor networks in dynamic scenarios, International Journal of Ubiquitous Computing and Intelligence 1 (1) (2007) 9–16. 8.
[22] V. Gallart, S. Felici-Castell, M. Delamo, A. Foster, J.J. Perez, Evaluation of a real, low cost, urban wsn deployment for accurate environmental monitoring, in: Proceedings of IEEE MASS '11, 2011, pp. 634–639.
[23] S. Kim, R. Fonseca, D. Culler, Reliable transfer on wireless sensor networks, in: Proceedings of IEEE SECON '04, 2004, pp. 449–459.
[24] R.C. Shah, S. Roy, S. Jain, W. Brunette, Data mules: modeling a three-tier architecture for sparse sensor networks, in: Proceedings of IEEE SNPA '03, 2003, pp. 30– 41.
[25] S. Jain, R. Shah, W. Brunette, G. Borriello, S. Roy, Exploiting mobility for energy efficient data collection in wireless sensor networks, ACM/Springer Mobile Networks and Applications 11 (3) (2006) 327–339.
[26] A. Chakrabarti, A. Sabharwal, B. Aazhang, Using predictable observer mobility for power efficient design of sensor networks, in: Proceedings of IPSN '03, 2003, pp. 129–145.
[27] A. Somasundara, A. Kansal, D. Jea, D. Estrin, M. Srivastava, Controllably mobile infrastructure for low energy embedded networks, IEEE Transactions on Mobile Computing 5 (8) (2006) 958–973.
[28] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, D. Estrin, Intelligent fluid infrastructure for embedded networks, in: Proceedings of ACM Mobisys 2004, 2004, pp. 111–124.
[29] G. Anastasi, M. Conti, E. Monaldi, A. Passarella, An adaptive data-transfer protocol for sensor networks with data mules, in: Proceedings of IEEE WoWMoM 2007, 2007, pp. 1–8.
[30] K. Shah, M. Di Francesco, G. Anastasi, M. Kumar, A framework for resource-aware data accumulation in sparse wireless sensor networks, Computer Communications 34 (17) (2011) 2094–2103.
[31] F. Shaikh, A. Khelil, A. Ali, V. Suri, TRCCIT: tunable reliability with congestion control for information transport in wireless sensor networks, in: Proceedings of ICST WICON '10, 2010, pp. 1–9.
[32] T. Le, W. Hu, P. Corke, S. Jha, ERTP: Energy-efficient and reliable transport protocol for data streaming in wireless sensor networks, Computer Communications 32 (7–10) (2009) 1154–1171.
[33] R.G. Dimakis, V. Prabhakaran, K. Ramch, Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes, in: Proceedings of ACM/IEEE IPSN '05, 2005, pp. 111–117.
[34] P. Karlsson, L. Öberg, Y. Xu, An address coding scheme for wireless sensor networks, in: Proceedings of ADHOC '05, 2005, pp. 3–4.
[35] H. Wen, C. Lin, F. Ren, Y. Yue, X. Huang, Retransmission or redundancy: transmission reliability in wireless sensor networks, in: Proceedings of IEEE MASS '07, 2007, pp. 1–7.
[36] S. Ali, A. Fakoorian, H. Taheri, Optimum Reed–Solomon erasure coding in fault tolerant sensor networks, in: Proceedings of IEEE ISWCS '07, 2007, pp. 6–10.
[37] W. Xiao, D. Starobinski, Extreme value FEC for reliable broadcasting in wireless networks, IEEE Journal on Selected Areas in Communications 28 (2010) 1180–1189.
[38] M. Sammer Srouji, Z. Wang, J. Henkel, RDTS: A reliable erasure-coding based data transfer scheme for wireless sensor networks, in: Proceedings of IEEE ICPADS '11, 2011, pp. 481–488.
[39] B. Marchi, A. Grilo, M. Nunes, DTSN: Distributed transport for sensor networks, in: Proceedings of IEEE ISCC '07, 2007, pp. 165–172.
[40] M.C. Vuran, I.F. Akyildiz, Error control in wireless sensor networks: a cross layer analysis, IEEE/ACM Transaction on Networking 17 (2009) 1186–1199. doi:http://dx.doi.org/10.1109/TNET.2008.2009971.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025

[41] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, M. Zorzi, SYNAPSE++: code dissemination in wireless sensor networks using fountain codes, IEEE Transactions on Mobile Computing 9 (2010) 1749–1765.
[42] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, IEEE Communication Magazine 44 (11) (2006) 134–141.
[43] M. Conti, J. Crowcroft, S. Giordano, P. Hui, H.A. Nguyen, A. Passarella, Middleware for Network Eccentric and Mobile Applications, Springer, 2009 (Ch. 6: Routing Issues in Opportunistic Networks, pp. 121–147).
[44] G. Anastasi, M. Conti, M. Di Francesco, Reliable and energy-efficient data collection in sparse sensor networks with mobile elements, Performance Evaluation 66 (12) (2009) 791–810. doi:http://dx.doi.org/10.1016/j.peva.2009.08.005.
[45] I.S. Reed, G. Solomon, Polynomial codes over certain finite fields, Journal of the Society for Industrial and Applied Mathematics 8 (2) (1960) 300–304. doi:http://dx.doi.org/10.1137/0108018.
[46] L. Rizzo, L. Vicisano, RMDP: an fec-based reliable multicast protocol for wireless environments, ACM SIGMOBILE Mobile Computing and Communications Review 2 (1998) 23–31. doi:http://doi.acm.org/10.1145/584017.584020.
[47] F. Restuccia, G. Anastasi, M. Conti, S. Das, Performance Analysis of a hierarchical discovery protocol for WSNs with mobile elements, in: Proceedings of IEEE WoWMoM 2012, 2012.
[48] G. Anastasi, E. Borgia, M. Conti, E. Gregori, A. Passarella, Understanding the real behavior of mote and 802.11 ad hoc networks: an experimental approach, Pervasive Mobile Computing 1 (2) (2005) 237–256.
[49] A.K. Dwivedi, O.P. Vyas, An exploratory study of experimental tools for wireless sensor networks, Wireless Sensor Network 3 (7) (2011) 215–240.
[50] Sentilla Corporation, Tmote Sky Datasheet v.1.04, <http://sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf> (retrieved 23.11.10).
[51] TinyOS <http://www.tinyos.net/>, 2010.
[52] Chipcon, 2.4 GHz IEEE 802.15.4/ZigBee-Ready RF Transceiver, Chipcon Products from Texas Instruments, CC2420 Data Sheet, 2004.

**Eleonora Borgia** is a Researcher of the Italian National Research Council (CNR). She received the Laurea and the Ph.D degree in Computer Engineering both from the University of Pisa, Italy. Her research activities are focused on wireless and pervasive computing addressing issues related to data dissemination for WSNs and to social community detection and service provisioning for DTNs. In the past she also worked on routing algorithms and MAC protocols for MANETs. She was/is in the organizing and technical committee of international conferences/workshops serving as Publicity Chair, Publication Co-Chair and TPC member (e.g, IEEE MASS, IEE GLOBECOM, IEEE ICC, IFIP Wireless Days, IEEE CCNC, ACM PM2HW2N, IEEE PerMoby, IEEE AOC and IEEE PerSeNS).

**Giuseppe Anastasi** is a Professor and the Vice Chair of the Department of Information Engineering at the University of Pisa, Italy. His research interests include pervasive computing, sensor networks, sustainable computing, and computing for sustainability. He is the founding co-chair of the Pervasive Computing & Networking Laboratory (PerLab), and has contributed to many research programs funded by both national and international institutions. He is a co-editor of two books: Advanced Lectures in Networking (LNCS 2497, Springer, 2002) and Methodologies and Technologies for Networked Enterprises (LNCS 7200, Springer, 2012). He has published over 100 research papers in the area of computer networking and pervasive computing. Currently, he is an Associate Editor of Sustainable Computing (SUSCOM) and an Area Editor of Pervasive and Mobile Computing (PMC). Previously, he served as Area Editor of Computer Communications (ComCom), General Chair of IEEE WoWMoM 2005, Program Chair of IFIP/IEEE SustainIT 2012, IEEE PerCom 2010 and IEEE WoWMoM 2008. He is one of the co-founders of the IFIP/IEEE SustainIT conference and of the IEEE PerSeNS workshop. He has been a member of the IEEE Computer Society since 1994.

**Marco Conti** is a Research Director of the Italian National Research Council (CNR). He has published in journals and conference proceedings more than 300 research papers related to design, modelling, and performance evaluation of computer networks, pervasive systems and social networks. He co-authored the book "Metropolitan Area Networks (MANs): Architectures, Protocols and Performance Evaluation" (Springer 1997), and he is co-editor of the books: "Mobile Ad hoc networking: the cutting edge technologies," (IEEE-Wiley 2012), "Mobile Ad Hoc Networking" (IEEE-Wiley 2004), and Mobile Ad Hoc Networks: from Theory to Reality (Nova Science Publishers 2007). He is Editor-in-Chief of Elsevier *Computer Communications* journal and Associate Editor-in-Chief of Elsevier *Pervasive and Mobile Computing* journal. He received the *Best Paper Award* at IFIP TC6 Networking 2011, IEEE ISCC 2012 and IEEE WoWMoM 2013. He served as TPC chair for several major conferences – including IFIP Networking 2002, IEEE WoWMoM 2005, IEEE PerCom 2006, and ACM MobiHoc 2006 – and he was general chair (among many others) for IEEE WoWMoM 2006, IEEE MASS 2007 and IEEE PerCom 2010. He is the founder of successful conference and workshop series, such as *ACM RealMAN, IEEE AOC, ACM MobiOpp, and IFIP/IEEE SustainIT*.

Please cite this article in press as: E. Borgia et al., Energy efficient and reliable data delivery in urban sensing applications: A performance analysis, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.07.025