2020 IEEE International Conference on Smart Computing (SMARTCOMP)

QoS-aware Data Management Mechanisms for Optimal Resource Utilisation in Crowd-assisted Shared Sensor Networks

Simone Bolettieri, Raffaele Bruno
Institute of Informatics and Telematics (IIT)
Italian National Research Council (CNR)
Via G. Moruzzi 1, 56124, Pisa, ITALY
Email: {s.bolettieri,r.bruno}@iit.cnr.it

***Abstract*—In this study, we focus on the problem of managing a hybrid, shared IoT-based monitoring system, in which stationary sensor devices are complemented with user-carried personal devices embedded with sensing capabilities. The envisioned crowd-assisted monitoring system must support the sharing of the sensing infrastructure among multiple concurrent sensing tasks that can have highly varying QoS requirements. In such a scenario, a key issue is to maximise the utilisation efficiency of the physical sensing resources and the QoS satisfaction of sensing tasks while limiting the redundancy of collected data. As in previous research, we advocate the use of an IoT Broker, an intermediary entity that (i) interacts with the IoT applications to collect their QoS requirements (i.e., spatial coverage, data notification frequency); and (ii) coordinates with the redundant sensor deployments and mobile devices to selectively activate and configure the data streams that are needed to fulfil application requirements in a cost-efficient way. Then, we have developed an optimisation framework to jointly select the set of physical sensing resources to activate and the data update frequency for maximising the overall sensing performance while limiting redundant data. A key feature of our proposed framework is to be privacy-friendly as it only requires coarse-grained space-time knowledge of device location. Extensive simulations under realistic WSN deployments and real-life mobility patterns confirm the efficiency of the proposed solution in terms of data-coverage gain and reduction of data redundancy with respect to classical non-hybrid monitoring systems.**

I. Introduction

The number of connected Internet-of-Things (IoT) devices continues to grow at a steady pace, triggering a massive influx of sensory data. New market forecasts estimate that there will be between 40 to 70 billion IoT devices, or "smart things", by 2025 [1]. This opens the way for a wide variety of new IoT-enabled use cases and applications that will have a significant impact on our daily lives [2]. Smart cities are one of the most promising use cases for new IoT solutions, as various types of sensors and devices are now being deployed everywhere in the urban environment to manage the city services and its infrastructures [3] efficiently.

Smart city monitoring systems require new design approaches for wireless sensor networks (WSNs) to overcome the limitations of flexibility and utilisation efficiency of traditional task-oriented and application-specific solutions. In a smart-city scenario, different devices with heterogeneous sensing, processing and communication capabilities coexist in the same infrastructure and eventually support multiple applications and services. Furthermore, different IoT services might have different QoS requirements, e.g. in terms of data quality and accuracy, geographic granularity, and timeliness of data delivery. Two major design trends are emerging to satisfy such design goals. The first one promotes the utilisation of virtualisation technologies to decouple the underlying sensing infrastructure from the services provided to applications while facilitating the sharing among different applications of the sensed data collected by sensors [4]. An example of such a design approach is the concept of virtual sensors, namely virtualised representations and combinations of different sensor measurements, typically maintained within a cloud platform [5]. This approach inherits the typical limitations of cloud-based technologies, such as communication latency, network overloading, and lack of data locality. A second compelling paradigm for collecting sensed data in urban environments is mobile crowdsensing (MCS), in which mobile devices of a potentially large number of participants (a crowd) contributes data produced by their embedded sensors [6]. The focus of existing research in mobile crowdsensing is mainly on participant selection, incentive mechanisms to increase users' willingness to participate in sensing tasks, and task allocation.

Recently, a number of research works have proposed *hybrid* sensing platforms that combine mobile crowdsensing with static sensing to provide more ubiquitous coverage with lower deployment costs [7], [8]. Several challenges arise in enabling a crowd-assisted shared sensing system (CA-SSN). The key issues we focus in this study are: i) the coordination between heterogeneous stationary and mobile devices to pursue improved QoS satisfaction for the sensing tasks, and better utilisation efficiency of the physical sensing resources; and ii) the optimisation of the sensed data streams to limit data redundancy. To achieve these goals, we promote the use of an IoT Broker to manage the data collection process efficiently. Inspired by previous works that exploited a similar concept [9], [10], [11], an IoT Broker is a logical entity that acts as a proxy to satisfy service requests from IoT applications. Specifically, the IoT Broker interacts with the applications, which ask for a specific type of sensor data from a given geographical area, and with a minimum frequency of data updates. At the same time, the IoT Broker interacts with the sensing platforms that

978-1-7281-6997-2/20/$31.00 ©2020 IEEE
DOI 10.1109/SMARTCOMP50058.2020.00031

are available within the monitored area to collect the requested information. Following this approach, in this study we propose a QoS-aware data management scheme to enable the IoT Broker to optimally *activate* the available sensing resources (either stationary or mobile), and to properly *shape* the generated IoT data traffic to maximise the overall sensing performance while limiting the redundancy of the collected data. Note that the IoT Broker implements a local, short-term cache to store the collected sensory data to be able to respond to multiple service requests using the same data. A fundamental assumption of our system is that it can rely only on an aggregate and coarse-grained information on device location to protect users' privacy. We have modelled our resource allocation problem as a mixed-integer linear problem. Our mathematical framework explicitly accounts for: $i)$ data redundancy due to multiple sensing tasks asking for similar data at similar locations, and $ii)$ location uncertainty of mobile devices due to the coarse-grained location information. Finally, the proposed scheme is evaluated via simulation considering realistic heterogeneous WSNs, deployed IoT applications, and mobility patterns. Results illustrate that the proposed solution is capable of outperforming the evaluated benchmarks in terms of data coverage and reduction of data redundancy.

In the rest of this paper, we first discuss the related work in Section II. Afterwards, we present the system model, and we formulate the problems of optimally activating and shaping IoT data streams in Section III. Section IV describes the mathematical programming framework to solve the considered problem. Evaluations are performed in Section V. We finally draw the conclusion in Section VI.

II. Related Work

This work is related to various streams of existing literature on the design of novel architectures for urban sensing.

There is a considerable body of work focusing on MCS systems, and we refer the reader to [6] for a comprehensive survey. A fundamental research challenge in MCS is the optimisation of the sensing task allocation (separately or jointly) considering various aspects, such as spatial coverage, incentive costs, energy consumption, and task completion time [12]. For instance, several MCS frameworks have been proposed which aim at either maximising the sensing quality with budget constraints (e.g., [13]), or minimising the incentive cost while guaranteeing a minimum level of sensing quality (e.g [14]). Other studies focus on how to optimise the overall energy consumption of an MCS system rather than the energy consumption of individual mobile nodes (e.g. [15]). An important thread of research tackles the problem of multi-task allocation, where the interplay of multiple tasks is also considered (e.g. [16]). A hybrid two-phased MCS system is described in [17], which combines opportunistic and participatory sensing modes. As to our knowledge, the problem of task allocation in hybrid MCS system in which mobile nodes are complemented with in-situ sensor nodes is much less investigated. The authors in [8] formulate the scheduling problem of mobile and stationary sensor nodes to maximise the average coverage and data accuracy, and minimise the average number of active nodes given the knowledge of the users' movements. A prototype for an hybrid MCS framework, named HySense, is described in [7], which proposes to migrate redundant users from densely populated areas to sparsely populated areas to balance sensing opportunities among different regions. Differently from these previous studies, we do not assume a fixed sensing rate for each sensor type, but we exploit a QoS-aware, adaptive sampling scheme to control data redundancy under bandwidth constraints.

It is also important to point out that task allocation and task scheduling have been well studied also in the context of shared sensor networks or virtual sensor networks (see [18] for a comprehensive survey on the topic). However, most of the proposals assume that sensing tasks are directly allocated to sensor nodes. Thus, the focus of these studies is to find an optimal mapping of application tasks on a set of heterogeneous sensor nodes subjects to applications requirements, e.g. to maximise the system lifetime and the number of admitted applications (e.g. [19], [4]). The authors in [20] formulate the allocation problem of multiple sensing tasks by assuming that multiple sensors can collect data for the same target location cooperatively. There is also a vast literature on adaptive sampling in traditional WSNs. However, the focus of this studies is how to leverage the spatial and temporal correlations between measurements to adapt sampling frequency to improve the power efficiency while ensuring the accuracy of sampled data (e.g. [21], [22]. Differently from our work, these studies do not apply application information-sharing mechanisms to reduce data redundancy and improve the utilisation efficiency of physical resources.

Finally, some recent research works have proposed to provide more advanced QoS support for IoT applications by introducing a broker entity that selects and properly shapes the IoT data streams to satisfy application requirements. One of the first examples of resource allocation policies for IoT brokers is proposed in [23], to dispatch application requests to multiple sensor nodes to maximise the lifetime of the IoT platform while meeting the QoS requirements specified by applications. An IoT broker is also proposed in [9] to efficiently manage and adjust network slices assigned to an IoT platform based on aggregate QoS requirements. In our recent work [11], we proposed an algorithm to allow an IoT broker to adjust the polling rate of physical sensing resources in a shared sensor network to reduce network congestion. To the best of our knowledge, there are no previous studies that use an IoT broker in the context of CA-SSN.

III. Models and Problem Definition

In this section we detail the models that are used in our proposal, including the overall system architecture and the application model. Afterwards, we introduce the core problem addressed in this paper.

A. Application model

In this study, we assume that the monitored area is divided into a set $\mathcal{C}$ of square grid cells. A cell is described by c_k, $k = 1, \ldots, M$, where M is the total number of cells . Similarly to [8], a data type represents a physical parameter that is measured in the reference area. Generally, complex data types may require the composition of multiple parameters (e.g. a fire detection application may require CO_2, temperature and smoke). Without loss of generality, we assume atomic monitoring applications, and each data type is associated with a single sensor type. Formally, let $\mathcal{D} = \{d_1, \ldots, d_L\}$ be a set of L different sensor types that are accessible by external monitoring applications. Each sensor type has its own characteristics in terms of temporal and spatial resolution, maximum sampling frequency, and resource consumption. For the sake of model simplicity, we assume that a sensor measurement refers only to the cell where the sensor is located. However, our framework can be easily generalised to more complex settings in which a sensor measurement has spatial coverage beyond a single cell. Furthermore, we assume that a constant energy e_l is dissipated by the sensor type d_l when it makes a measurement (processing load or memory requirements are not considered). A data payload b_l (expressed in terms of bytes) is needed to convey the measured data. Then, we assume that at regular time intervals equal to T minutes, external monitoring applications generate a set of U heterogeneous sensing tasks $\mathcal{S} = \{s_1, \ldots, s_U\}$ to be served by a CA-SSN platform. The QoS requirements of a task s_j are specified through a 3-tuple $\langle \mathcal{C}_j, r_j, d(j) \rangle$, where $\mathcal{C}_j \subseteq \mathcal{C}$ is the set of target regions to monitor (expressed in terms of cells), r_j is the desired notification frequency of sensed data, and $d(j)$ is the data type to be collected. In a hard-QoS allocation scheme (e.g. [4]), a sensing task would be admitted in the system only if all its requirements are fully satisfied (namely, all target locations are covered with the requested sensing rate). In this study, we consider a *soft-QoS model* where there is not an absolute reservation of resources, but the system goal is to maximise the spatio-temporal coverage of the sensing tasks (see Section IV for a detailed explanation).

B. System model

In our CA-SSN system, sensors are embedded into two classes of nodes. Specifically, let $\mathcal{N}^s = \{n_1, \ldots, n_{N^s}\}$ be the set of stationary sensor nodes that are randomly deployed by one or more infrastructure providers over the monitored area, and $\mathcal{N}^m = \{n_1, \ldots, n_{N^m}\}$ be the set of user-carried mobile devices (namely smartphones or vehicles' sensors), which are embedded with built-in sensors. Let $N = N^s + N^m$ and $\mathcal{N} = \mathcal{N}^s \cup \mathcal{N}^m$. Additionally, each node $n_i \in \mathcal{N}$ is characterised by a given resource vector $O_i = \langle E_i^B, \mathcal{D}_i \rangle$, which specifies its battery capacity E_i^B, and the set $\mathcal{D}_i$ of sensor types embedded into that node. Note that static sensor nodes are expected to have more limited resources that typical mobile devices. Each mobile node has his visiting pattern of cells based on the mobility behaviours of carrying users. Due to privacy concerns, we assume that *users are not willing to share the GPS trajectories of their movements but only aggregate information*. Hence, to characterise the location of nodes over time, we assume a discrete and coarse-grained space-time representation. Formally, the time domain is divided into equally spaced time slots with duration τ (with $T = \omega \cdot \tau$). Then, the placement of nodes during time slot t is represented by an $N^s \times M$ placement matrix $P[t] = \{p_{ik}[t]\}$, where $p_{ik}[t] = 1$ iff $n_i \in \mathcal{N}^S$ and it is positioned in c_k. Depending on the spatial and time granularity of the model representation (i.e., cell size and time slot duration), and the users' mobility pattern, a mobile device could visit more than one cell during a single time slot. To capture this variability, $p_{ik}[t] = \alpha$ if $n_i \in \mathcal{N}^m$ spends a percentage α of time slot t moving within cell c_k. By definition, we have that $\sum_{k=1}^{M} p_{ik}[t] = 1$. Note that we assume that the system has no control over the movement of mobile nodes, but their location is observable and predictable, i.e. $P[t]$ is known at any time slot.

Stationary nodes are organised into several WSNs, each one with a different number of nodes but, possibly, partially overlapping sensing areas. Furthermore, each WSN is independently managed and operated by an IoT Gateway (GW), which is also the data collection node (or sink) to which the data locally generated by the sensor nodes are delivered. We assume a classical tree-like routing structure (as in 6LoWPAN-based WSNs [24]), and each node of the tree forwards its traffic to the IoT GW through a single parent node using short-range wireless links. Formally, let $\mathcal{G}$ be the set of gateways, and $\mathcal{G}_g$ the set of static nodes $n_i \in \mathcal{N}^s$ that are managed by gateway g. Now, let denote with $p(i)$ the parent node for node $n_i \in \mathcal{N}^s$. The total traffic flow (in bytes/s) that is transmitted by node i towards its parent node $p(i)$ during time slot t is denoted by $f_i[t]$. Note that $f_i[t]$ includes both the sensed data locally collected by node n_i during time slot t, and the traffic forwarded on behalf of neighbouring nodes. We assume to know the application-level bandwidth bw_i between n_i and $p(i)$, i.e. the maximum application traffic that can be transmitted by node n_i to its parent node with negligible message losses. Thus, it holds that $f_i[t] \leq bw_i$. The definition of an application-level link capacity allows us to abstract all modelling complexities due to interference and medium access protocols. As far as the mobile sensor nodes, we assume that they are connected to the network using long-range wireless communication links (e.g., NB-IoT or LTE-M). In this case, we can assume that a limited and predefined *data quota* B^m is assigned by the communication infrastructure to the crowdsensing service (see also [9]) to ensure that the users' participation to the sensing platform has a negligible impact on the transfer delays of user-generated traffic.

To complete the description of the system architecture, we introduce the IoT Broker component, i.e., the orchestrator that handles the data requests from the sensing tasks by exploiting the data collected by the different IoT GWs. Similarly to [9], we envision that the IoT Broker exposes a northbound interface to allow applications to request a monitoring service with specific QoS requirements. Then, the IoT Broker interacts with the IoT GWs to retrieve the descriptions of the data types provided by the sensor nodes (a separate IoT GW is assumed to manage the repository of the service descriptors for the mobile sensor nodes in the network), as well as the current network and resource status. Based on the collected information, the IoT Broker can activate the data streams from the sensing devices, and configure their notification periods to satisfy the application requests best. It is important to point out that the data retrieved by the IoT Broker is locally *cached* to facilitate information application sharing. In other words, the IoT Broker can reuse the cached data to respond to multiple application requests demanding the same data type for the same target cell (but with different notification periods).

C. Problem definition

In our model, we assume that each sensor can be activated individually during a time slot. Qualitatively, the goal of the data management scheme implemented by the IoT Broker is to dynamically negotiate with the IoT GWs an *activation plan* for their managed resources, namely (i) which sensor on which node should be activated during each time slot, and (ii) which notification period should be assigned to activated sensors, to efficiently satisfy the QoS requirements of the sensing tasks. Formally, we introduce a $N \times L$ plan matrix $R[t] = \{r_{il}[t]\}$, where $r_{il}[t] \geq 0$ is the notification frequency assigned to the sensor of type d_l on node n_i during time slot t. Note that we consider a node n_i active during time slot t, if at least one sensor on n_i is active during time slot t. We can now compute the *total sensing data rate* $B_i[t]$ generated by node i as $\sum_{l \in \mathcal{D}_i} r_{il}[t] \times b_l$.

As explained in the following, different sensing tasks could have overlapping QoS requirements as they require the same data type for the same cell (with a similar notification frequency), which allows optimising the data flows from activated sensors. The goals of the data-management problem could be to reduce data redundancy or energy consumption while maximising the spatio-temporal coverage of the application. In principle, the IoT Broker should prioritise the use of static nodes over mobile nodes, as the latter typically need and incentive to be enrolled in a sensing task. The allocation problem is further complicated by the uncertainty of information about mobile nodes, as the system can only rely on an aggregate estimate of the time a mobile node spends within a cell. Thus, optimally scheduling service requests can be difficult or unfeasible.

IV. QoS-Aware Data Management

With the assumptions and models we have introduced above, we formulate the activation and configuration problem for the sensor data streams as the following (linear) integer programming problem. To simplify the notation, and when no ambiguity occurs, in the following we use the subscript index i to refer to a node n_i, the subscript index j to refer to a sensing task s_j, the subscript index k to refer to a cell c_k, and the subscript index l to refer to a data type d_l.

First of all, we introduce the decision variable $x_{ij}[t]$, which is a binary variable equal to one if a sensor embedded into node n_i is activated during time slot t to satisfy the notification requests of sensing task s_j. Thus, an *activation plan* $X[t] = \{x_{ij}[t]\}$ is a $N \times U$ binary matrix. Note that if a stationary node and a mobile node can both satisfy the service requests of the same sensing task, our framework gives priority to the stationary node. To this end, we introduce the spatial impact function $h_{ik}[t]$, which assigns a weight to node n_i located in cell c_k, when contributing sensed data during time slot t. We write that

$$h_{ik}[t] = \begin{cases} 0, & \text{if } p_{ik}[t] = 0 \\ I, & \text{if } n_i \in \mathcal{N}^s \\ p_{ik}[t], & \text{otherwise} \end{cases}, \tag{1}$$

where $I > 1$. Basically, the impact of a mobile node on the *coverage quality* of a sensing task is equal to the duration of its presence in the cell; while the impact of a stationary node is an arbitrary number greater than one (the reason of this choice will be clear later in the model development). Finally, for notation compactness we introduce the following auxiliary sets: $i)$ $\mathcal{S}_i^a[t]$ is the set of tasks j that can be served by a node i during time slot t, i.e., i provides data type $d(j)$ ($d(j) \in \mathcal{D}_i$) and $\exists k \in \mathcal{C}_j : p_{ik}[t] > 0$); $ii)$ $\mathcal{S}_l^b$ is the set of tasks j that requested data type l ($d(j) = l$); and $iii)$ $\mathcal{S}_k^c$ is the set of tasks j interested in monitoring cell k ($k \in \mathcal{C}_j$). We also introduce the auxiliary binary variable $x_{ilk}[t]$, which is equal to one iff the sensor type l on node i is active during time slot t. We can now define the overall average coverage obtained by sensing task j given an activation plan $X[t]$ as follows:

$$Z_j[t] = \sum_{k \in \mathcal{C}_j} \sum_{i \in \mathcal{N}} x_{ij}[t] \times h_{ik}[t] \ . \tag{2}$$

It is worth point out that the coverage formulation in equation (2) implicitly assumes that each active sensor generates data at the rate requested by the associated sensing tasks. Our formulation can be generalised to include sensing tasks that can accept a lower notification frequency. In this case, a data utility function needs to be specified to quantify the amount of sensing data that is provided, and its accuracy. This extension is left as future work.

The overall optimisation problem can be written as follows:

$$\max \sum_{t=1}^{\omega} \sum_{j \in \mathcal{S}} Z_j[t] \tag{3}$$

$$\text{s.t. } x_{ij}[t] = 0 \qquad d(j) \notin \mathcal{D}_i \tag{4}$$

$$(p_{ik}[t] - \beta) \times x_{ij}[t] \geq 0 \qquad \forall i \in \mathcal{N}^m, \forall j \in S_k^c \tag{5}$$

$$x_{ilk}[t] \leq \sum_{j \in \mathcal{S}_k^c \cap \mathcal{S}_l^b} x_{ij}[t] \qquad \forall l \in \mathcal{D}_i \tag{6}$$

$$x_{ilk}[t] \geq x_{ij}[t] \qquad j \in \mathcal{S}_k^c \cap \mathcal{S}_l^b, \forall l \in \mathcal{D}_i \tag{7}$$

$$I \geq \sum_{i \in \mathcal{N}} x_{ilk}[t] \times h_{ik}[t] \qquad \forall k \in \mathcal{C}, \forall l \in \mathcal{D} \tag{8}$$

$$r_{il}[t] \geq x_{ij}[t] \times r_j \qquad \forall j \in \mathcal{S}_l^b, l \in \mathcal{D}_i \tag{9}$$

$$r_{il}[t] \leq \sum_{j \in \mathcal{S}_l^b} x_{ij}[t] \times r_j \qquad \forall l \in \mathcal{D}_i \tag{10}$$

$$f_i[t] = B_i[t] \qquad \forall i \in \mathcal{N}^m \tag{11}$$

$$\sum_{i \in \mathcal{N}^m} f_i[t] \leq B^m \tag{12}$$

$$f_i[t] = B_i[t] + \sum_{\substack{h \in \mathcal{N}^s \\ i = p(h)}} f_h[t] \qquad \forall i \in \mathcal{N}^s \tag{13}$$

$$f_i[t] \leq bw_i \qquad \forall i \in \mathcal{N}^s \tag{14}$$

$$\sum_{i \in \mathcal{G}_g} B_i[t] = \sum_{\substack{i \in \mathcal{G}_g \\ p(i) = g}} f_i[t] \tag{15}$$

$$\sum_{t=1}^{\omega} \frac{E_i[t]}{E_i^B} \leq LT_i \qquad \forall i \in \mathcal{N} \tag{16}$$

Objective (3) maximises the overall sensing coverage over the time frame T. Constraint 4 ensures that a node i cannot be activated if data type $d(j)$ is not present on board. Constraint 5 requires that an active mobile node resides in a cell of interest for task j for at least a time $\beta\tau$. This design choice prevents using nodes that can only contribute a small amount of relevant data. Equations (6)-(7) check if node i is used to collect data type l in the target cell k. Constraint (8) controls data redundancy. Specifically, depending on the mobility pattern, a mobile node may provide only partial coverage of cell k during a time slot t. Thus, it may be useful to have data collected from more mobile nodes in the cell. The parameter I is used to set a bound on the data redundancy. As introduced earlier, a stationary node guarantees a coverage of the entire slot duration and it should be always preferred over a mobile node. Setting up the contribution of a static node to the coverage quality to the maximum value I ensures that the solution of the optimisation problem always selects a stationary node if available. Constraints (9)-(10) ensure that the sampling frequency assigned to a sensor type l on an active node is able to satisfy the QoS requirements of all the sensing tasks that are served by than node. Equations (11) and (13) specify the total data rate in time slot t that is transmitted by a mobile node and a static node, respectively. Note that Equation (13) expresses the classical flow-conservation property. Constraint (12) ensures that the total data rate of all active mobile nodes does not exceed the data quota B^m. Constraint (14) ensures that the total data rate of an active stationary node i does not exceed the communication capacity bw_i. Equation (15) specifies that all sensed data generated by the stationary nodes that are managed by the same IoT GW is received by the gateway. Finally, constraint (16) ensures that the normalised energy dissipation during the time frame T is bounded by parameter LT_i. This is equivalent to guarantee a minimum lifetime for each sensor node. To compute the energy $E_i[t]$ consumed during time slot t we assume a simplified energy model that accounts only for the energy dissipation due to sensing and data transmission, while data processing and idle times are ignored. Let e_i^b denote the energy consumed during the reception or transmission of 1 bit of data by node i. It holds that

$$E_i[t] = \sum_{l \in \mathcal{D}_i} r_{il}[t] \times e_l + \left(f_i[t] + \sum_{\substack{h \in \mathcal{N}^s \\ i = p(h)}} f_h[t] \right) \times e_i^b \qquad i \in \mathcal{N}^s \tag{17}$$

$$E_i[t] = \sum_{l \in \mathcal{D}_i} r_{il}[t] \times e_l + B_i[t] \times e_i^b \qquad i \in \mathcal{N}^m \tag{18}$$

Since we consider a multi-hop WSN, the energy consumed by the energy transceiver of a stationary node takes into account also forwarded traffic. On the contrary, mobile nodes do not forward traffic on behalf of other nodes.

V. Performance Evaluation

In this section, we evaluate the benefit of deploying a CA-SSN system that exploits the proposed QoS-aware data management scheme. To this end, we compare performance with two benchmark cases: i) a traditional WSN without mobile nodes, and ii) a mobile crowdsensing system in which all mobile nodes that visit a cell of interest for a sensing task are activated during that time slot. For the sake of brevity, afterwards, we refer to our solution as `CA-SSN`, while we refer to the former and latter benchmark cases as `WSN` and `MCS`, respectively. Simulations are performed using a custom simulation framework written in Matlab. Main simulation parameters are summarised in Table I. For the baseline case, we set the time slot duration τ to one minute, while the planning period for the resource allocation is set to $T = 1$ hour. To ensure statistical soundness, the following results have been obtained by considering 100 scenarios in which mobility patterns and application mixture are randomly varied.

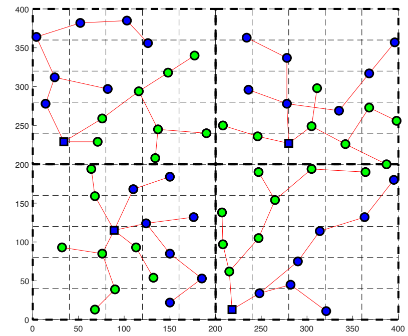


Fig. 1. The deployment of stationary sensor nodes used in the experiments. Green and blue circles depict LL and HL sensor nodes, respectively. Squares are the sinks. Red lines represents the wireless links between nodes and their parent in the routing tree. Dashed lines identify the cells and regions.

TABLE I
BASELINE SIMULATION SETUP

Simulation Parameters		Values
Cells	Number M	100
	Size (length) (m)	40
Nodes	N^s	64
	N^m	[0,128]
	Speed of Mobile Nodes (m/s)	[0.5, 1]
	B^m (Mbps)	2
	bw_i (kbps)	125
Simulation	duration of time slot t (s)	60
	duration of time frame T (h)	1
	redundancy level I	1.5
	β	0.15

A. Sensor nodes

Sensor nodes are deployed in a $400x400$ m scenario. The stationary sensing infrastructure consists of $N^s = 64$ randomly deployed nodes. Furthermore, we assume that the reference area is divided into four equal regions as shown in Figure 1. Each region as its one data collection point or sink (i.e. $|G| = 4$), which is the root of the routing tree used to deliver the sensed data to the wired network. Then, each stationary node is connected to the sink which can be reached with the minimum number of hops (we consider a default transmission range of 60 m). Similarly to [8] we divide the monitored area into cell with side length equal to 40 m, which corresponds to a grid of $M = 100$ cells. In the network shown in Figure 1, 57% of the cells are covered by at least a stationary sensor node.

The stationary sensing infrastructure accounts for the typical hardware heterogeneity of sensor nodes. As in previous work (see [25]), we consider two categories of sensor node hardware, namely high-level (HL), and low-level (LL) sensor nodes. The former category accounts for devices with a more powerful processor and an energy budget of 180 KJ (roughly a 5V with 10Ah of battery supply), which can be equipped with power-hungry sensors (e.g. a camera for multimedia applications). The latter category are devices with a more constrained energy budget of 32.4 KJ (roughly a 3.6V with 2.4Ah of butter supply), which can only support low-power analog sensors. Each stationary node is randomly assigned to one of the two node classes. We assume that two separate routing trees are maintained, one for each node category. This is equivalent to envision that two separated WSNs coexist in the same geographic area and share the same data collection points[1]. Thanks to the proposed brokering framework, the two WSNs are used as a single, shared sensing infrastructure. Note that we use two separate WSNs to avoid that an LL node running out of energy may affect the sink connectivity for HL nodes, and vice versa. Finally, we set the maximum application data rate between two stationary nodes equal to 125 kbps (i.e. $bw_i = 125$ kbps).

Regarding the user-carried mobile nodes, we envision a set of 4G-enabled smartphones integrating all the sensor types that are described in Section V-B, and a battery with capacity $E_i = 4000$ mAh. To accurately model the users' movement patterns, we use the well-known HCMM model [26]. Basically, HCMM is a community-based mobility model that organises the users into communities, and mimic typical human mobility patterns, such as the tendency of spending time in a limited set of favourite locations (e.g. the home and working place), and the inclination to prefer short trips. Thus, each community is associated to a preferred region (called home cell), and the strength of the social relationships between the users of the same community or different communities is the key driver to select the destination region of a movement. In the following experiments, we have partitioned the mobile nodes into four equally-sized groups, each one assigned to one of the four regions in Figure 1. Then, a rewiring probability equal to 0.5 is used to create relationships between nodes belonging to different communities. When moving, each node selects a speed uniformly at random over $[v_{min}, v_{max}]$. Finally, for the crowdsensing service, the data quota B^m is set to 2 Mbps.

B. Applications and sensor types

In this study, we assume that four different data types are monitored in the reference area, which corresponds to sensed data collected by simple analog sensors or more complex digital sensors, and whose characteristics are summarised in Table II. Specifically, we consider a temperature sensor (d_1) and a CO sensor (d_2), which are analog sensors characterised by a small data payload (127 bytes), and a limited power consumption ($e_1 = 15$ mJ and $e_2 = 85$ mJ, respectively). For the category of digital sensors, we consider an embedded camera (d_3), which produces a medium data payload (2.5 Kbytes) and has a medium energy dissipation (100 mJ) per collected frame; and an embedded microphone (d_4), which has both a high bandwidth (192 Kbytes) and energy (2500 mJ) consumption for each second of recorded audio.

[1]Note that this is easily implementable using RPL as an RPL-based border router supports the formation of multiple DODAGs as a means to dynamically and autonomously partition the network.

TABLE II
PARAMETER VECTOR OF SENSOR DATA TYPES.

Data Type	d_1	d_2	d_3	d_4
Sensor type	Temperature	CO	Camera	Audio
E_l^s (mJ)	15	85	100	2500
b_l (B)	127	127	2500	192K
Technology	analog	analog	digital	digital

In the performance evaluation we assume that four classes of sensing tasks are to be deployed: (i) a temperature monitoring application (`appT`) that require a reference sampling rate equal to 0.5 Hz; (ii) a CO monitoring application (`appG`), which requests to measure the CO gas concentration at a rate of 1 Hz; (iii) a multimedia application (`appC`) that acquires an image per second; and, (iii) an audio recording application (`appA`) that acquires an audio sample of 4 seconds once every minute. Table III summarises the characterisation of sensing applications. We set the number of target cells to be monitored by an application equal to five for `appT` and `appG` sensing tasks, and to three for the `appC` and `appA` sensing tasks (a similar design is also adopted in previous works, as as [25]). Note that the applications randomly select their target cells in the reference area. Clearly, the greater the number of active sensing tasks, and the higher the probability that one cell is of interest for more than one application of the same category.

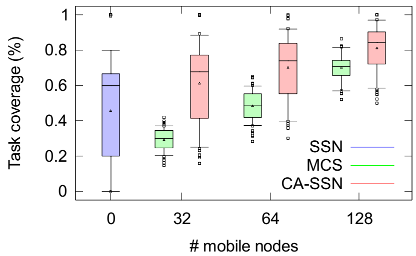
TABLE III
QOS REQUIREMENTS FOR REFERENCE APPLICATIONS.

Parameter	Application class			
	`app`	`appG`	`appC`	`appA`
r_j (sample/min)	30	60	60	1
$\lvert C_j \rvert$	5	5	3	3
$d(j)$	d_1 (Temp)	d_2 (CO)	d_3 (Camera)	d_4 (Audio)

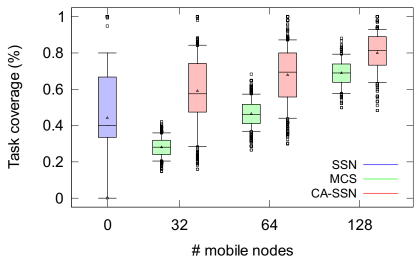
C. Results

The performance evaluation metrics that we use to asses the efficiency of our QoS-aware data management scheme for CA-SSNs are the following:

- *Task coverage*: This metric measures the average fraction of sensor samples requested by a sensing task that is actually received during a time frame T. We remind that a sensing task demand a total amount of samples equal to $r_j \times T$. The task coverage will depend on the number of target cells that are covered by at least a (stationary or mobile) sensor node, and the temporal duration of the coverage.
- *Data redundancy*: This metric measures the fraction of sensor samples that are received by a sensing task, which are redundant. Depending on the I parameter, multiple mobile sensor nodes can be activated in the same time slot for a target cell. Furthermore, inaccurate location information leads



(a) $U = 4$



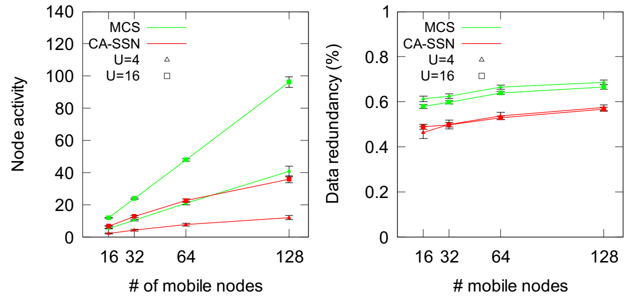
(b) $U = 16$

Fig. 2. Boxplots of task coverage vs node mobility. Triangle points are the mean value. The lower and upper whiskers are the 5% and 95% percentile, respectively. Circles are the outliers.

to sub-optimal activation plans. A proper balance between coverage and redundancy is crucial.

- *Node activity*: This metric measures the number of mobile nodes that are activated during each time slot, on average.

1) Impact of Network Scale: First of all, we assess the ability of our QoS-aware data management scheme to exploit mobile sensor nodes to improve system performance efficiently. Figure 2a and Figure2b show a boxplot of the average coverage per sensing task obtained with the tested mechanisms when increasing the number of mobile nodes. Results in Figure 2a and Figure2b are obtained in a scenario with one task per application category (i.e. $U = 4$), and four tasks per application category (i.e. $U = 16$), respectively. We decided to show a boxplot because it well describes not only how data are distributed but also the variability of the data set. In the plotted boxplots, the whiskers represent the 9th and 91st percentile. As shown in the results, in `WSN` scenario, 25% of the sensing tasks obtains an average coverage lower than 20% when $U = 4$. In the `MCS` system, as the number of mobile nodes increases, we notice that the task coverage rapidly increases. In a scenario with 128 mobile nodes (i.e. twice the number of stationary nodes of the `WSN` scenario), the median coverage is roughly the same as in the `WSN` case, but there is a threefold improvement of the lower quartile. Furthermore, the `MCS` system is characterised by a much lower dispersion of the task coverage values than the `WSN` case. Finally, while a `MCS` system is able to avoid the occurrence of sensing tasks with very poor coverage, it is less efficient than the `WSN` scheme to ensure high task coverage when node density is comparable. On the contrary, our `CA-SSN` scheme significantly outperforms all tested algorithms. It is sufficient to exploit 32 mobile nodes to obtain a 25% improvement over `WSN` of median task coverage. With 128 mobile nodes, the median task coverage is roughly 90%. Similar trends have been observed when increasing the number of sensing tasks (Figure 2b).



(a) (b)

Fig. 3. Node activity (3a) and data redundancy (3b) vs. node mobility. Plots show average values with 95% CI.

Figure 3a shows that the coverage gain of `opt-M` is obtained by maintaining the number of active mobile nodes almost three times lower than `MCS`. Furthermore, Figure 3b shows that the use of an IoT Broker to allow information sharing among similar applications, coupled with a smart strategy to select mobile nodes is also beneficial to reduce data redundancy. More precisely, data redundancy slowly increases as the number of mobile nodes increase, and `CA-SSN` obtains a 25% reduction of data redundancy when compared to `MCS`. All previous results have confirmed the benefits of using mobile sensing nodes as an addition to existing in-situ sensor deployments.

2) Impact of Node Mobility: Figure 4 shows the average task coverage when increasing the speed of mobile nodes for a scenario with 64 mobile nodes and $U = 4$. The results show that there is a negligible impact of speed on coverage performance. However, if we analyse the trends of node activity (see Figure 5a), we observe that node activity increases when increasing the speed of individual nodes increase. Such behaviour can be explained by noting that the higher the speed, and the lower the average sojourn time of a node in a cell, and the more the number of cells that can be visited by a node in a time slot. Thus, more mobile nodes can be potentially activated. It is also important to point out that the node activity is at least three times lower in `CA-SSN` than `MCS` because our solution aims at maintaining the node redundancy level below the threshold I (see constraint (8) in the optimisation framework). Finally, Figure 5b shows that data redundancy also increases with node mobility. Again, the explanation of such behaviour is a higher probability to visit more that one cell during a time slot, with a potential increase of the amount of sensed traffic that is generated when the mobile node is outside the assigned cell.

We conclude this section by observing that the uncertainties about the location of mobile nodes can be reduced by adopting smaller cells and shorter time slots. However, there is a trade-off between location accuracy, users' privacy and computational tractability of the optimisation problem.

VI. CONCLUSIONS

In this paper, we have proposed a crowd-assisted shared sensing infrastructure that leverage an IoT Broker to optimally activate

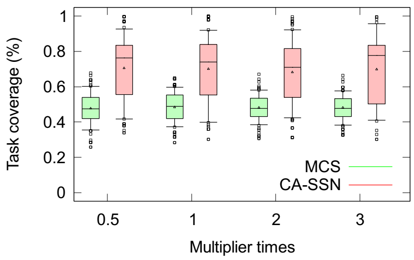
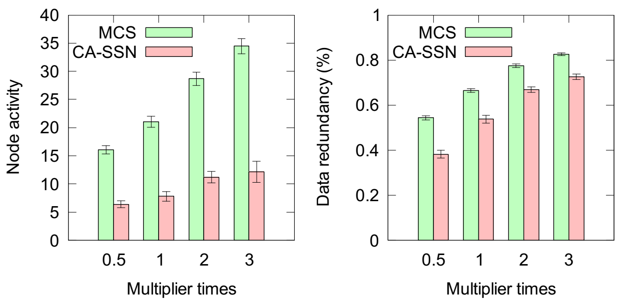


Fig. 4. Box-plots of task coverage vs. speed of mobile nodes (multiplier times $v \in [0.5, 1.0]$ (m/s), $N^m = 64, U = 4$). Triangle points are the mean value. The lower and upper whiskers are the 5% and 95% percentile, respectively. Circles are the outliers.



(a) (b)

Fig. 5. Node activity (5a) and data redundancy (5b) vs. speed of mobile nodes (multiplier times $v \in [0.5, 1.0]$ (m/s), $N^m = 64$). Plots show average values with 95% CI.

and shape IoT data streams from stationary and mobile nodes to satisfy the service requests from heterogeneous monitoring applications. Furthermore, we have formulated a mathematical programming framework to solve the considered problem, which accounts for data redundancy, users' mobility patterns, and applications' requirements. Finally, using simulations, we have investigated the performance gains of the proposed solution when compared to conventional WSNs and mobile crowdsensing systems.

Future work aims at exploring the impact of uncertainties in user mobility on the system efficiency by using different mobility prediction models. We also plan to include cooperation between different sensor nodes to improve the quality of sensing. Finally, our framework can be extended to consider elastic applications that dynamically adapt their QoS requirements based on general utility functions.

References

[1] Gartner Research, "IoT Global Forecast and Analysis, 2015-2025," online, March 2017.

[2] Sharma, Neha and Shamkuwar, Madhavi and Singh, Inderjit, *The History, Present and Future with IoT*. Springer, 2019, pp. 27–51.

[3] R. Du, P. Santi, M. Xiao, A. V. Vasilakos, and C. Fischione, "The Sensable City: A Survey on the Deployment and Management for Smart City Monitoring," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1533–1560, Secondquarter 2019.

[4] C. Delgado, M. Canales, J. Ortín, J. R. Gállego, A. Redondi, S. Bousnina, and M. Cesana, "Joint Application Admission Control and Network Slicing in Virtual Sensor Networks," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 28–43, February 2018.

[5] S. Madria, V. Kumar, and R. Dalvi, "Sensor Cloud: A Cloud of Virtual Sensors," *IEEE Software*, vol. 31, no. 2, pp. 70–77, March 2014.

[6] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A Survey on Mobile Crowdsensing Systems: Challenges, Solutions, and Opportunities," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2419–2465, thirdquarter 2019.

[7] G. Han, L. Liu, S. Chan, R. Yu, and Y. Yang, "HySense: A Hybrid Mobile CrowdSensing Framework for Sensing Opportunities Compensation under Dynamic Coverage Constraint," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 93–99, March 2017.

[8] Q. Zhu, M. Y. Sarwar Uddin, N. Venkatasubramanian, and C. Hsu, "Spatiotemporal Scheduling for Crowd Augmented Urban Sensing," in *Proc. of IEEE INFOCOM'18*, April 2018, pp. 1997–2005.

[9] V. Sciancalepore and F. Cirillo and X. Costa-Perez, "Slice as a Service (SlaaS) Optimal IoT Slice Resources Orchestration," in *Proc. of IEEE GLOBECOM'17*, December 2017, pp. 1–7.

[10] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on Multi-Access Edge Computing for Internet of Things Realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, Fourthquarter 2018.

[11] S. Bolettieri and R. Bruno, "Edge-centric Resource Allocation for Heterogenous IoT Applications using a CoAP-based Broke," *International Journal of Cloud Computing*, 2020.

[12] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task Allocation in Mobile Crowd Sensing: State-of-the-Art and Future Opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3747–3757, October 2018.

[13] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola, "Fostering participaction in smart cities: a geo-social crowdsensing platform," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 112–119, June 2013.

[14] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proc. of IEEE INFOCOM*, April 2015, pp. 2254–2262.

[15] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "iCrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, August 2016.

[16] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K. Leung, "Energy-Aware Participant Selection for Smartphone-Enabled Mobile Crowd Sensing," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1435–1446, September 2017.

[17] J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, D. Zhang, B. Guo, and Q. Lv, "HyTasker: Hybrid Task Allocation in Mobile Crowd Sensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 598–611, March 2020.

[18] C. M. De Farias, W. Li, F. C. Delicato, L. Pirmez, A. Y. Zomaya, P. F. Pires, and J. N. De Souza, "A Systematic Review of Shared Sensor Networks," *ACM Computing Surveys*, vol. 48, no. 4, pp. 51:1–51:45, February 2016.

[19] W. Li, F. C. Delicato, P. F. Pires, Y. C. Lee, A. Y. Zomaya, C. Miceli, and L. Pirmez, "Efficient allocation of resources in multiple heterogeneous Wireless Sensor Networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1775–1788, 2014.

[20] D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, and Y. Xiang, "Energy minimization in multi-task software-defined sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3128–3139, Nov 2015.

[21] P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings, "A Utility-based Adaptive Sensing and Multihop Communication Protocol for Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 6, no. 3, pp. 27:1–27:39, June 2010.

[22] R. Bruno and M. Nurchis, "Efficient data collection in multimedia vehicular sensing platforms," *Pervasive and Mobile Computing*, vol. 16, pp. 78–95, 2015.

[23] G. Tanganelli and E. Mingozzi, "Energy-Efficient IoT Service Brokering with Quality of Service Support," *Sensors*, vol. 19, p. 693, 02 2019.

[24] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, March 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6550.txt

[25] C. Delgado, J. R. Gallego, M. Canales, J. Ortin, S. Bousnina, and M. Cesana, "On Optimal Resource Allocation in Virtual Sensor Networks," *Ad Hoc Networks*, vol. 50, no. 1, pp. 23–40, November 2016.

[26] C. Boldrini and A. Passarella, "HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships," *Computer Communications*, vol. 33, no. 9, pp. 1056–1074, 2010.