Optimal Network Coding-based In-Network Data Storage and Data Retrieval for IoT/WSNs

Camila H. S. Oliveira, Yacine Ghamri-Doudane, Carlos E. F. de Brito,
Stéphane Lohier

Optimal Network Coding-based In-Network Data Storage and Data Retrieval for IoT/WSNs

Camila H. S. Oliveira*, Yacine Ghamri-Doudane†, Carlos E. F. Brito‡, and Stéphane Lohier*

* Université Paris-Est Marne-la-Vallée, LIGM Lab, 75420, Champs sur Marne, France
† University of La Rochelle, L3i Lab, Av. Michel Crépeau, 17042, La Rochelle CEDEX 1, France
‡ Federal University of Ceará, Computer Science Dept., Campus do Pici, 60.440-900, Fortaleza-CE, Brazil
Email: {souzaol; yacine.ghamri; lohier}@univ-mlv.fr and carlos@lia.ufc.br

Abstract—The evolution of the Internet of Things (IoT) allows the development of new services and applications but triggers as well a full set of new issues to be solved. Among them, there are the problems related to the integration of the WSNs in the IoT realm including those related to data access. In this paper, we address more precisely the in-network data storage and data retrieval performed in a WSN integrated in the IoT realm. In order to develop an adequate data storage scheme for this scenario, we first design a system that integrates the Virtual Broking Coding (VBC) data storage scheme in the IoT realm. Then, we propose an algorithm called Dynamic Adaptive Virtual Broking Coding (DA-VBC) that adapts dynamically the packet redundancy level adopted in VBC to the optimal redundancy level, regarding the actual condition of the network, in order to ensure a reliable data storage and data retrieval. To do so, we model the choice of the optimal redundancy level as a Markov Decision Process (MDP) problem. Using the optimal policy found by the MDP, DA-VBC always performs with the minimum cost-benefit for the network which means allowing more packet to be retrieved without overload the energy consumption. The simulation results confirm that the dynamic adaptation of the redundancy level improves the reliability of the data storage scheme while achieving an energy consumption comparable to the solution that does not use any redundancy. Besides, they show that the optimization of the cost-benefit metric is far more efficient than optimizing only one metric (for instance the cost or the packet delivery ratio), or using a fixed redundancy level.

I. INTRODUCTION

Internet of Things (IoT) is a huge virtual network made up of heterogeneous devices which are able to communicate through several different access technologies, building an intelligent abstraction of collected information about physical process around the world. The interoperability of different networks in the IoT allows smart environment applications to come to reality, as well as the creation of new services and applications in the Wireless Sensor Networks (WSNs) domain, [1], [2]. However, these new services and applications bring new issues to the WSNs in view of their integration within in the IoT realm, such as the data access management.

Among the research trends raised by the data management in the IoT, this paper addresses the in-network data storage and data retrieval performed in the WSNs integrated in the IoT realm. Although these problems have been approached in some works, there is still a lack of adequate solutions. Indeed, the several solutions found in the literature, such as in [3]–

[9], do not take into account the new challenges and needs imposed by the integration of the WSN in IoT. In our previous work [10], we proposed, as a first attempt, an in-network data storage scheme called VBC that assumes an IoT-based application scenario in order to overcome this shortage.

In [10], we were mainly interested in how to store the data produced in the network and how to ensure a reliable data retrieval by any consumer in the network. For that, we proposed to store coded packets generated through a network coding technique that allowed VBC to store packets with some redundancy level. We defined the packet redundancy level as the relation between the number of original packets used in the codification and the number of generated coded packet. Although VBC ensures an easier and reliable access to the information, which is of primary importance in IoT, VBC must also be able to adapt itself according to the changes in the network condition in order to keep both, its reliability as well as a transparent and seamless access to the resources.

In this paper, aiming to make VBC an adequate solution to the IoT, we propose its evolution through the design of a system that defines a communication model that integrates the WSNs in the IoT, as well as a dynamic self-adjusting in-network caching scheme called Dynamic Adaptive Virtual Broking Coding (DA-VBC) that ensures a reliable data storage and data retrieval through all changes during the network lifetime. DA-VBC performs the dynamic adaptation of the redundancy level, and also the choice of the optimal redundancy level considering the actual network state. To do so, we model the choice of the optimal redundancy level as a Markov Decision Process (MDP) problem. Our main goal with the optimization of the packets redundancy is to minimize the cost-benefit of VBC. For that, the MDP finds the optimal redundancy level taking into account the impact of its cost and gain in allowing the packets to be retrieved successfully. As a result in this work:

- We design a communication model that integrates the VBC-based WSNs in the IoT realm, and allows a local and global data access by building the abstraction of the information claimed in IoT.
- We provide a theoretical analysis of VBC from which we derivate general expressions that calculate the cost and the gain of VBC according to the redundancy level and the probability of packet loss in the network. In

this analysis, we show the need of adapting the data redundancy according to the network situation in order to achieve the best cost-benefit.

- We thus investigate, formulate and solve the optimization problem of choosing the redundancy level in the dynamic adaptation of VBC. More precisely, we formulate the problem as a MDP and propose the DA-VBC algorithm, which uses the value iteration algorithm in order to solve it.
- Simulation results show clearly the impact of the optimal choice of the redundancy level and its dynamic adaptation, proposed by DA-VBC, in the delivery of the packets requested by the consumers, the cost-benefit and the network lifetime.

The remainder of the paper is organized as follows. In Section II we discuss the related works. The VBC system model integrated in the IoT realm is shown in Section III. Then, in Section IV, we perform an analytical study of the performance of VBC in an IoT scenario. Section V outlines the problem formulation as a MDP and presents the Dynamic Adaptive VBC algorithm. We present our simulation results supporting our theoretical analysis in Section VI and conclude this work in Section VII.

II. RELATED WORKS

In the literature we can find several proposed solutions for the problems related to the WSNs integration to the IoT realm, [11]–[16]. However, we could not find a proposal that targets both the in-network data storage and the data retrieval. In [17], the authors propose a management platform for treating and processing all information collected in the IoT environment. Yet, they only approach the high level problem of gathering and analyzing the data. In [12]–[14], [18], [19] and [16], the authors propose different models for the WSNs integration in IoT, as well as for the access to data produced in these networks. But, also in these cases, the in-network caching, which is a major concern that may greatly influence the system performance, is neglected. In [20], the authors propose an in-network caching considering the IoT scenario. However, they consider that the storage nodes are all connected by cables, which does not suit all IoT practical scenarios. Nonetheless, they reinforce the result obtained by us in [10], which shows that using a structure that intermediate the access to the data is more efficient than caching the data at the source node and retrieving them through a directly communication between source and consumer nodes.

On the other hand, the works that perform in-network data storage do not assume the WSN integration in the IoT realm, [3]–[9]. In order to overcome this shortage we proposed in [10] an in-network data storage scheme called Virtual Broking Coding (VBC) that, although envisaged to an IoT scenario, was tested in the scenarios proposed in [21] for a fair comparison with its solution, the Balance Storage (BS). Both VBC and BS use a structure called Virtual Broker (VB) made up of Virtual Broker Nodes (VBNs) that intermediate the communication between the consumers and producers through the publish/subscribe model. In this model, the producers publish their data packets to the closest VBN, and the consumers subscribe their interests to their closest VBN as well. The solution presented in [21] performs a dynamic load balancing mechanism in the VB. It basically transfers packets from an overloaded VBN to another one with less stored data. In order to be able to keep a balanced virtual broker, each VBN periodically sends a control message to the neighboring VBNs. The control messages allow the VBNs to decide if they can transfer packets to another VBN, and to choose the less overload VBN to do so. However, BS does not perform any data redundancy which results in a high packet loss ratio. VBC, on the other hand, although improving the packet delivery ratio as a consequence of storing redundant coded packets, does not take into account that its redundancy level must be updated according to the change in the network condition. Besides, VBC has not defined its relation with the entities allowing the integration of the WSNs into the foreseen bigger picture, the IoT.

III. VBC SYSTEM MODEL

In this section we present the VBC scheme, as proposed in [10], and the communication model proposed here for its integration to the IoT realm.

A. System Definition

We define a system that makes local data storage while allowing local and global data retrieval. The system provides the access to both local and global data through the interaction among the four entities composing the system which are: the producers, the broker nodes, the IoT gateway and the consumers. The producers are the sensors that do the environment monitoring and generate the data to be stored. The consumers are the devices interested in receiving some data and they can be of two types: actuators, or mobile devices such as cell phones, tablets, notebooks and any other connected device. The broker nodes are the devices responsible for storing the data produced. And the gateway is a more robust device responsible for intermediating the communication between the Virtual Broking (VB) and the cloud. An application of the VBC system model is pictured in Figure 1.

In our system the access to the data can be performed in two levels: local and global. We define local data access when the consumer is interested in a data produced in the same partition that it comes from. Here, the consumer subscribes its request to the VB which gets the data locally from the virtual brokers nodes. The global data access, on the other hand, happens when a consumer subscribes for data published in another partition or network. In this case, the communication is performed directly to the gateway that get the data from the cloud and forwards it to the consumer. Besides, in the VBC system, we assume a new consumption model where the data are consumed in group of n packets that may come from either the same source or multiple sources. The new data consumption model allows a better adaptation of VBC in an IoT scenario, where the applications are interested in, for

Fig. 1: System model: an example

example, a multi-source data or measurements summarizing some value in a short period of monitoring. Figure 2 illustrates the relationships among the entities in the VBC system model.
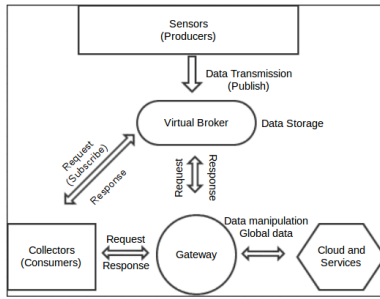

Fig. 2: System model: iteration among entities

B. Storage Scheme

The VBC storage scheme proposed in our previous work [10] is a Linear Network Coding (LNC) based scheme that stores the data in a virtual broking made up of broker nodes responsible for coding the data published by the producers. Using a virtual broking for storing the coded packets, VBC ensures an easy and reliable data retrieval for the consumers. In [10], we showed that VBC achieves a better distribution of the packet load among the broker nodes and a high level of data reliability. The most remarkable, though, is the fact that despite introducing redundancy in the network, with VBC the network delivers more packets for a lower cost. VBC uses LNC to store redundant data through generating M coded packets from N original packets published at the VB, where $M > N$, and M/N defines the data redundancy level of the stored data. Assuming the LNC definition, we have M coded packets that are generated as a linear combination of the N original packets $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

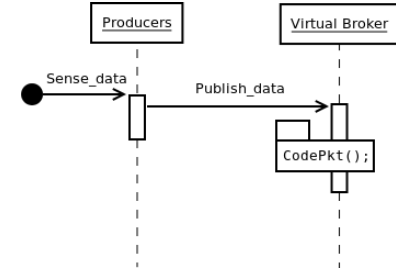$$\mathbf{y_j} = \sum_{i=1}^{N} \mathbf{C}_{ij} \cdot \mathbf{x_i} \qquad j = 1, \ldots, M$$

Fig. 3: Sequence diagram of the publish operation

and are represented as a coefficient matrix $N \times M$. As a consequence, in order to recover the requested packets, the consumers only need to receive any subset of N coded packets. As we assume a Pub/Sub architecture, VBC obviously works in two different phases: publish and subscribe. During the publish operation, the producers sense the data and send them to the closest VBN that after receiving N packets generates M coded packets. The M coded packets are then distributed among the VBNs and remain stored in the VB. As the publish operation does not concern the consumers, its communication model works independently of the consumers which are subscribing for getting information. The sequence diagram of the publish communication model is presented in Figure 3.

In the subscribe operation, the consumers subscribe to receive information at any VBN in the VB. The VBN that receives the subscription broadcasts a request in the VB and all VBNs that hold the requested packet answer the request directly to the consumer. In the subscribe operation, as we have different types of consumers, we have two communication models. In the first one, we assume that the consumers are more robust and sophisticated devices that can communicate through various technologies, such as wifi, 3G, 4G, etc. In this case, the communication is performed as follows:

- The device subscribes for one information through sending a request to the gateway. Here, we have two possibilities. If the device subscribe for a local information:
  - The gateway forwards the request to the VBNs
  - The VBNs answer the request to the gateway
  - The gateway sends directly the information required to the device

  Otherwise:
  - The gateway catches the information from the cloud
  - The gateway sends the information required to the device

In the second case, however, the consumers are mostly static devices interested in local information in order to take some decision at a local level. Thus, the communication model follows the steps below, and the sequence diagram of the subscribe operation is presented in Figure 4.

- The devices subscribe for information to the closest VBNs
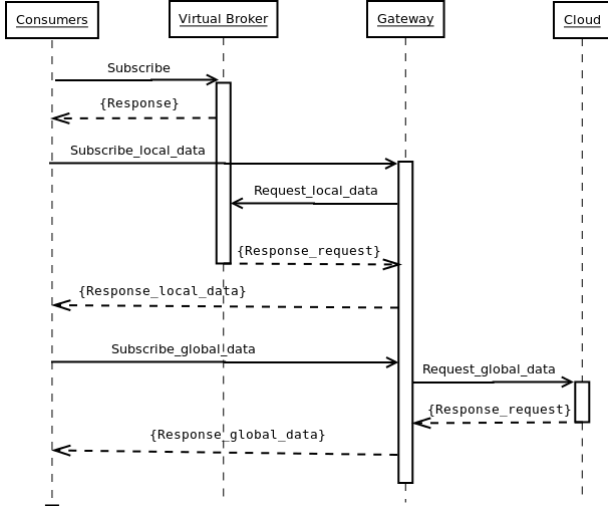- The VBN forwards the request to the VB

Fig. 4: Sequence diagram of the subscribe operation

- The VBNs that have the requested information send an answer directly to the device

After describing the systemic view of the data access management in WSN while integrated in the IoT realm, let us concentrate on the in-network data storage, more precisely, we focus on the problem of dynamically adapting the redundancy level of the data storage scheme performed in the VBC system.

IV. THEORETICAL ANALYSIS

In order to formulate our optimization problem, we need to derive the expressions used as the objective function of the targeted optimization. In [10], we presented a detailed analysis of the communication cost and the packet delivery ratio of VBC assuming the scenario proposed in [21] with a virtual broker of a fixed size of 4 VBNs. In this work, after making a few simplifications in the assumptions adopted in [10], we derive general expressions for the cost and the packet delivery probability in an arbitrary scenario based on a VB network.

Consider an observation interval T, and assume the network is divided into partitions of size s, each with a VB of size V. Denote the (average) loss probability in end-to-end communications by σ. We further assume that there is a single source p which generates packets at the rate of r_1 pkts/sec, and a single data collector c which consumes packets at the rate of r_2 pkts/sec.

We have the VBC scheme using a redundancy level of M/N. Our simplification consists of assuming that each coded packet has a probability σ of being lost during the distribution phase, even if the packet is to be stored in the same VBN where it was generated. We further assume that each request has probability a σ of being lost during the broadcast phase, even if the request is to be 'sent' to the same VBN which is performing the broadcast. These assumptions allow a uniform processing of the coded packets and requests, and lead to relatively simple expressions for the communication cost and packet delivery ratio. Now, consider a request from the collector node c that reaches the VB. The collector will

be able to recover the requested data only if it receives at least N coded packets. Focusing on a single coded packet, we observe that the packet will reach the node c if (a) the packet is not lost during the distribution phase, (b) the request is not lost during the broadcast phase, and (c) the packet is not lost during the transmission from the VBN, where it is stored, to the node c. This occurs with probability $(1-\sigma)^3$. Since there are M coded packets associated with each request, the number of coded packets that reach the node c is a binomial random variable with parameters M and $(1-\sigma)^3$. So, the probability for a request reaching the VB to lead to a data recovery is given by:

$$f_p = \sum_{j=N}^{M} \binom{M}{j} (1-\sigma)^{3j} (1-(1-\sigma)^3)^{M-j} \qquad (1)$$

The collector node consumes data at the rate of r_2 pkts/sec, and each (successful) request yields N data packets. Thus, during the observation interval T, the collector node generates Tr_2/N requests. Only a proportion $(1-\sigma)$ of them actually reach the VB. So, the (average) number of data packets recovered by the collector during the interval T is

$$\frac{Tr_2}{N} \cdot (1-\sigma) \cdot f_p \cdot N = Tr_2(1-\sigma)f_p \qquad (2)$$

Next, we estimate the expected communication cost associated with VBC. The cost associated with the publish operation consists of M transmissions of coded packets, in the distribution phase, for each N received data packets, in a total of MTr_1/N transmissions during the observation interval T. The cost associated with the subscribe operation consists of two components: V transmissions of requests in the broadcast phase, and $(1-\sigma)M$ transmissions of coded packets (on average) from a VBN to the consumer node c. Since the consumer node generates Tr_2/N requests during the observation interval T, the total expected cost of VBC is:

$$f_c = \frac{m(Tr_1)}{n} + (V + m(1-\sigma))(\frac{Tr_2}{n}) \qquad (3)$$

Equations (1)-(3) allow us to compute the cost-benefit metric (used here as the objective function of our optimization problem) which captures the communication overhead associated with the data packets actually recovered by the consumer node and the number of delivered packets. We would like to point out that in this work we have two different aspects. First, the packets are recovered together, which reduces the cost in delivering the requested packets. And, second, the number of VBNs are not anymore fixed up to 4. With this regard, Figure 5 shows the cost-benefit performance of VBC with a VB of size 6 and several redundancy levels, varying the parameter σ.

We clearly see in Figure 5 that the best redundancy setting changes according to the underlying conditions of the network, represented here by the parameter σ. For instance, while with $\sigma = 0.05$ we have the minimum cost-benefit using the redundancy level of 6/5, with $\sigma = 0.2$ the more adequate redundancy level would be 6/3. So, aiming to provide an in-network data storage scheme with optimal performance across
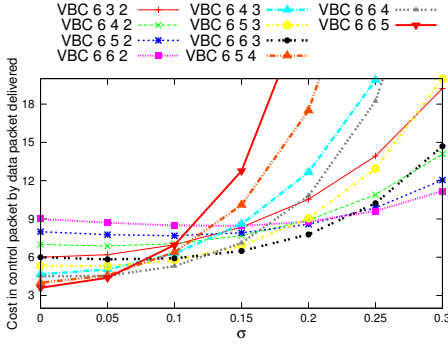
Fig. 5: Cost-benefit with virtual broker of size 6

all network conditions, we propose a dynamic optimization of the redundancy level used in VBC. To do so, in the next section we formulate an optimization problem and present the technique we used to solve it.

V. PROBLEM FORMULATION

In this section, we formulate the problem of dynamically choosing the redundancy level of VBC according to the current network conditions as a Markov Decision Process problem. The goal is to find a strategy for adjusting the redundancy level of the scheme that minimizes the cost-benefit metric over a long period of operation. What makes this problem non-trivial is the fact that the choice of a redundancy level for VBC may have an impact on the future conditions of the network, so that it may not be clear in a given situation what would be the best strategy in the long run. Moreover, since we consider that VBC will be integrated in an IoT scenario, we also take into account the possibility that application requirements might direct the performance of the scheme. More concretely, depending on the specific needs of the application, VBC may adopt a strategy that privileges the packet delivery over communication cost, or vice-versa.

A. Dynamic Optimization Methodology

MDPs are used to model and solve dynamic decision making problems, and consist of five elements: state space, decision epochs, actions, transition probabilities and rewards [22]. Our state will be defined as a triple $s = (m, n, \sigma)$, which describes the current redundancy level and estimated loss probability in the network. The variable σ will take values in the set $O = \{g, l, b\}$, which represent probability intervals associated with good, normal and bad conditions of the network, respectively. Formally, the *state space* is defined as

$$S = \{(m, n, \sigma) \mid 1 \le m, n \le V, m \ge n, \sigma \in O\} \quad (4)$$

The *decision epochs* are $T = \{1, 2, 3, \ldots\}$. At each decision epoch $t \in T$, an action determines the redundancy level to be used until the next epoch, based on the current state s_t of the system. So, the set of actions available to the system can be formally defined as

$$\mathcal{A} = \{(M, N) \mid M \le V, N \le M\} \quad (5)$$

Now, although an action deterministically sets the redundancy level that will be used next, the state of the system also includes the variable σ which has only a probabilistic dependence on this choice. The matrix below shows the estimated transition probabilities of σ for all redundancy levels. This matrix was found in [23] and defines the transition matrix of the Markov chain that represents the network behavior.

$$P_\sigma = \begin{bmatrix} P_{BB} & P_{BN} & P_{BG} \\ P_{NB} & P_{NN} & P_{NG} \\ P_{GB} & P_{GN} & P_{GG} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.7 & 0.3 \end{bmatrix} \quad (6)$$

Finally, the reward function $r(s, a)$ is basically the cost-benefit metric obtained from Equations (1) and (3). As we mentioned above, we would like to permit the applications to indicate the performance compromise that best suits them. For this purpose, we introduce the parameters p_1, p_2, which should be set by the application, and define the reward function as follows:

$$r(s, a) = \frac{[f_c(s)]^{p_1}}{[f_p(s)]^{p_2}} \quad (7)$$

Once the MDP is defined, the goal is to find a strategy for choosing the actions that should be taken in each state. Formally, a *policy* is a function $\pi : S \to A$ that assigns an action $a \in A$ to each state $s \in S$ of the system. Since we are interested in optimizing the performance of VBC in long periods of operation, we will consider the infinite horizon version of the problem, where each policy π is associated with an expected discounted reward defined by the formula

$$v^\pi(s) = \lim_{T \to \infty} E\Big[\sum_{t=1}^{T} \lambda^{t-1} \cdot r(X_t, A_t^\pi) \Big] \quad (8)$$

where λ is a discount factor in the interval $[0, 1)$, X_t is the random variable that indicates the state of the system at time t, and A_t^π is the random variable that indicates the action taken by the system at time t under policy π.

Now, we can formally define our goal as that of finding an optimal policy π^* with respect to Equation 8, in the sense that $v^{\pi^*}(s) \le v^\pi(s)$, for all π, s.

B. Dynamic Adaptation VBC Algorithm

As is well-known, the optimal expected discounted reward satisfies the following recurrence relation

$$v(s) = \min_{a \in \mathcal{A}} \Big\{ r(s, a) + \sum_{j \in S} \lambda p(j|s, a) v(j) \Big\} \quad (9)$$

called the Bellman's Equation. On the right-hand side of Equation (9), the first term represents the reward obtained at the current decision epoch when action a is taken, whereas the second term represents the expected discounted future reward when action a is taken. The solution of Equation (9) gives not only the minimum expected discounted reward $v(s)$, but also allows to recover an optimal policy π^* that achieves the optimal reward $v(s)$. There are several methods to solve the optimality equation (9) such as value iteration, policy iteration

and linear programming. In our proposed DA-VBC algorithm (see Algorithm 1), we will use the value iteration algorithm defined in [22].

Algorithm 1 Algorithm DA-VBC

1: **Input:** $s_0 = (n_0, m_0, \sigma_0)$, $\varepsilon > 0$
2: **Output:** π_t^*
3: **Pre-computed phase**
4: Select $v_0 \in V$, set $t = 0$
5: For each $s \in S$, compute $v_{t+1}(s)$ by
6: $v_{t+1}(s) = \min_a \left\{ r(s,a) + \sum_{j \in S} \lambda p(j|s,a) v_t(j) \right\}$
7: **if** $\|v_{t+1} - v_t\| < \varepsilon(1-\lambda)/2\lambda$ **then**
8:     Go to line 10. Otherwise, increment t by 1 and return to line 5
9: **end if**
10: For each $s \in S$, choose ε-optimal policy
11: $\pi^*(s) = \arg\min_a \left\{ r(s,a) + \sum_{j \in S} \lambda p(j|s,a) v_{t+1}(j) \right\}$
12: and stop.
13: **Operational phase**
14: Set t = 0
15: **while** $t \le T - 1$ **do**
16:     Track the channel condition
17:     Set $s = (m, n, \sigma)$
18:     Obtain action $\pi^*(s) = (m,n)$ based on the optimal policy
19:     Send updated (m,n) to VBNs
20:     Set $t = t + 1$
21: **end while**

Algorithm 1 is performed at the gateway that is responsible for detecting the network conditions. Then according to the observed behavior (σ), such as the increase or decrease in the packet loss, the gateway broadcasts to the VBNs the optimal redundancy level corresponding to the new state. To do so, Algorithm 1 consists of two phases: the Pre-computation phase and the Operational phase. In the Pre-computation phase the gateway finds the optimal deterministic police π^* based on the value iteration algorithm and records it as a look-up table. In the Operational phase, the gateway choose the action $\pi^*(s) = (m, n)$ (the redundancy level) based on the current state s and the optimal deterministic policy π^*.

VI. PERFORMANCE EVALUATION

A. Simulation Settings

In this section, we evaluate the performance of DA-VBC based on the the expected total reward defined in (7) to find the optimal police π^*(represented in the curves by π^{mdp}). As, for the best of our knowledge, we are the first work to dynamically optimize the redundancy level of packets stored in a WSN, we do not have another similar method to compare with DA-VBC. Therefore, we compare it with the performance of our benchmark BS [21], the previous original VBC scheme [10], and with two heuristic-based policies. The heuristics are: the policy π^{Cost} which always choose the state with the lower cost, and the policy π^{Benef} which always choose the state
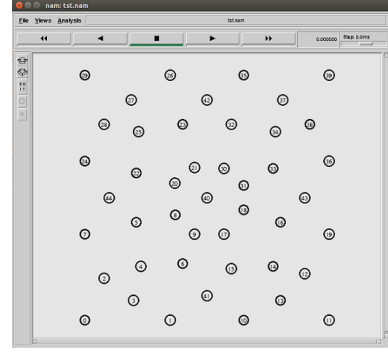

Fig. 6: Scenario of simulation.

with the greater packet delivery. We also show the impact of different application weight factors assigned to the two metrics ($\pi^{mdpCost_2}$ and $\pi^{mdpBenef_2}$) used to calculate the expected total reward of the DA-VBC MDP-based optimal policy.

We perform the experiment using the Network Simulator version 2(NS-2) where each solution is executed 100 times with a simulation time of 300s. We assume the scenario depicted in Figure 6, with four publishers and four subscribers. The node in the center represents the gateway defined in our system model (responsible for finding the optimal policy applied in the WSN) and the 8 nodes around it represent the broker nodes. We also introduce a background traffic in order to change the network condition represented by σ. At each time interval of 20s the traffic changes according to the matrix in (6).

In the experiments simulated, we define $\lambda = 0.99$ and we have a VB of size 8. We defined this VB size because we assume that the network is partitioned according to the algorithm proposed in [24] where they find the best VB size regarding the nodes density in the partition, that is in our case 8. As the maximum value allowed to M and N is defined by the size of VB, M and N can both go from 2 to 8. Regarding σ, it can assume three different values according to its situation (good, normal and bad). Thus, as a state is defined as the triple (m, n, σ), we have that our MDP has a total of 84 states respecting the condition that $M >= N$.

B. Results

Now, we discuss the impact of the π^{mdp} and the other heuristics regarding to three metrics: the network lifetime, defined here as the time in which the first node runs out of battery; the packet delivery ratio and the cost-benefit. Besides, we show the impact of applying weights to the cost and benefit metrics (used to define the cost-benefit) according to the application interest, and the improvement reached through the dynamic optimization comparing the result of π^{mdp} to those of VBC without optimization and BS.

Figures 7, 8 and 9 show the performance of π^{mdp} compared to BS and VBC with different redundancy level. In Figure 7, we clearly see that the π^{mdp} finds the optimal trade-off between the cost and the reliability through determining the states that result in the optimal expected total discounted

reward, which means the minimum cost-benefit. As a result of choosing the optimal cost-benefit, we have π^{mdp} performing better than VBC and BS regarding the percentage of packets delivered (8) and the cost-benefit (7). Figure 8 reveals that π^{mdp} delivers more packets than VBC, managing to deliver 70% more packets even when VBC performs with the maximum redundancy level possible (8/2) which represents the best option to VBC in terms of reliability. Besides, π^{mdp} delivers 170% more packets than BS. Regarding the network lifetime, Figure 9 confirms the result expected for this metric that comes from the fact that the bigger the redundancy level, the lower the network lifetime. What is more remarkable, though, is the fact that, the lifetime of π^{mdp} is comparable with that of BS even though it introduces redundancy in the network. Therefore, the value reached by π^{mdp} for the lifetime metric corroborates the results presented in [10] that, although introducing redundancy we achieves the delivery of more packets with a lower cost.
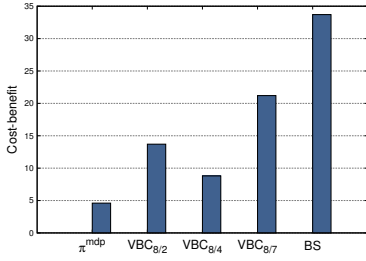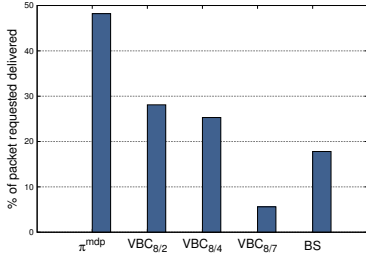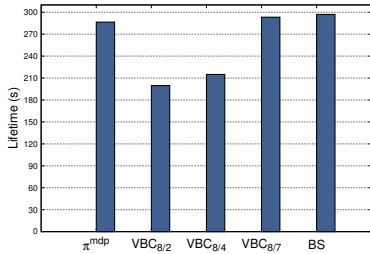

Fig. 7: Cost benefit

Fig. 8: Packet delivery

Fig. 9: lifetime

Now, we discuss the results of π^{mdp} when compared to the heuristics π^{Cost} and π^{Benef}. Observing Figures 10, 11 and 12 we have that the optimization through the cost-benefit metric is more efficient than when prioritizing only one parameter,

either the cost or the number of delivered requested packets. As we can see in 10, π^{mdp} presents the lowest cost-benefit, arriving to be three times less expensive than π^{Benef} and far less expensive than π^{Cost}. As a result of the redundancy level chosen by π^{mdp}, π^{Cost} and π^{Benef}, we have that the redundancy level found by π^{mdp} results in delivering more requested packets, and although π^{Cost} results in a greater lifetime, the value is comparable to that achieved with π^{mdp}. Actually, using π^{Cost}, π^{Benef} is the same that adopting VBC with a fixed redundancy level, in this case, $VBC_{8/7}$ and $VBC_{8/2}$, respectively. However, most importantly is that the π^{mdp} policy allows the network to perform in the best state, which means the best trade-off between the cost and the reliability incurred from the redundancy introduced by VBC.

In addition, another solution can be obtained assuming different weights to the the cost and the benefit in the calculation of the optimal expected total discounted reward of π^{mdp}. In this case, π^{mdp} can perform according to the need of the application. Using the weights, we can control the energy consumption without compromising the packet delivery so hard, as it happens when we consider only this metric as expected total discounted reward. For instance, we have $\pi^{mdpCost_2}$ that is defined as the expected discounted reward calculated with $P_1 = 2$ and $P_2 = 1$. In this case, the cost is prioritized in order to save energy. Indeed, we see in Figure 12 the improvement in the network lifetime. Besides, we observe the decrease of the cost-benefit compared to π^{mdp} in Figure 10 and the increase of the packet delivery compared to π^{Cost} in 11. This result proves that, even though one wishes to save energy, it is more efficient to do it optimizing our expected discounted reward by changing the weights than assuming only either the cost or the packet delivery metric. We observe the same when we prioritize the packet delivery by setting $P_1 = 1$ and $P_2 = 2$. In this case, $\pi^{mdpBenef_2}$ delivers more packets than π^{mdp} without compromising so much the cost-benefit when compared with π^{Benef}.
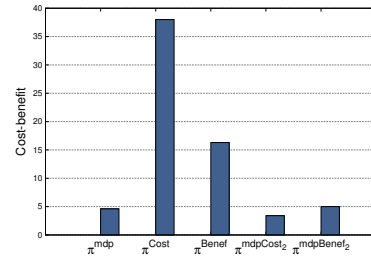

Fig. 10: Cost benefit

VII. CONCLUSION

In this paper, we proposed a communication model that integrates the VBC data storage scheme introduced by us in [10] in an IoT applications scenario, as well as a solution to the optimization problem of adapting dynamically the redundancy level (defined and used by VBC to ensure reliability in the data retrieval) according to the network condition represented by
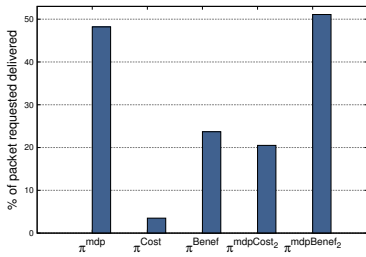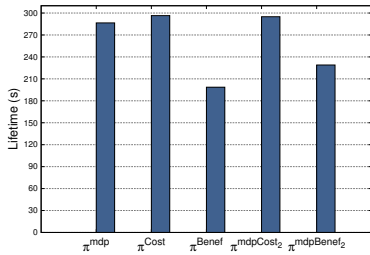
Fig. 11: Packet delivery

Fig. 12: lifetime

σ, the average loss probability of end-to-end communications. In order to solve this, we derived the expression for the cost-benefit metric and analyzed the impact of the redundancy level according to the σ. We saw that, indeed, the redundancy level that results in a better cost-benefit changes for different values of σ. So, we modeled the optimization problem as a MDP which uses the expression derived to calculate the cost-benefit as the expected total discounted reward, and proposed the DA-VBC to find the optimal redundancy level and dynamically adapt it according to the new state of the network.

The simulation results corroborate those presented by our theoretical analyses confirming that doing dynamic adaptation of the redundancy level improves the reliability of the data storage scheme while achieving an energy consumption comparable to the solution that does not use any redundancy, i.e. the Balanced Storage. Even more, simulation results also show that the optimization of the redundancy level regarding the cost-benefit metric is far more efficient than considering only either the cost or the packet delivery metric. Besides, DA-VBC allows an easy attribution of weights in the calculation of the expected total discounted reward, showing its flexibility as a great strength in the data storage process making it adaptable to the various IoT application scenarios.

REFERENCES

[1] Y. Liu, J. Niu, L. Yang, and L. Shu, "ebplatform: An iot-based system for ncd patients homecare in china," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec 2014, pp. 2448–2453.

[2] S. Moulik, S. Misra, C. Chakraborty, and M. Obaidat, "Prioritized payload tuning mechanism for wireless body area network-based healthcare systems," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec 2014, pp. 2393–2398.

[3] M. Albano and S. Chessa, "Data centric storage in zigbee wireless sensor networks," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*, June 2010, pp. 330–335.

[4] J. Crowcroft, M. Segal, and L. Levin, "Improved structures for data collection in wireless sensor networks," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 1375–1383.

[5] N. Cao, Q. Wang, K. Ren, and W. Lou, "Distributed storage coding for flexible and efficient data dissemination and retrieval in wireless sensor networks," in *IEEE Intl Conf on Comm*, May 2010, pp. 1–5.

[6] R. Zeng, Y. Jiang, C. Lin, Y. Fan, and X. Shen, "A distributed fault/intrusion-tolerant sensor data storage scheme based on network coding and homomorphic fingerprinting," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 10, pp. 1819–1830, Oct 2012.

[7] Z. Kong, S. Aly, and E. Soljanin, "Decentralized coding algorithms for distributed storage in wireless sensor networks," *IEEE Journal on Selected Areas in Comm*, vol. 28, no. 2, pp. 261–267, Feb 2010.

[8] A. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized erasure codes for distributed networked storage," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2809–2816, June 2006.

[9] S. Jafarizadeh and A. Jamalipour, "Distributed data storage in sensor networks based on raptor codes," in *IEEE 23rd Intl Symp on Personal Indoor and Mobile Radio Comm (PIMRC)*, Sept 2012, pp. 1509–1514.

[10] C. H. S. Oliveira, Y. Ghamri-Doudane, C. E. F. Brito, and S. Lohier, "Virtual broking coding for reliable in-network storage on wsans," in *Computers and Communications (ISCC), 2015 IEEE Symposium on*, July 2015.

[11] L. Jiang, L. D. Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An iot-oriented data storage framework in cloud computing platform," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 1443–1451, May 2014.

[12] S. Distefano, G. Merlino, and A. Puliafito, "Application deployment for iot: An infrastructure approach," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, Dec 2013, pp. 2798–2803.

[13] I. Zarko, K. Pripuzic, M. Serrano, and M. Hauswirth, "Iot data management methods and optimisation algorithms for mobile publish/subscribe services in cloud environments," in *Networks and Communications (EuCNC), 2014 European Conference on*, June 2014, pp. 1–5.

[14] W. Wang, K. Lee, and D. Murray, "Integrating sensors with the cloud using dynamic proxies," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, Sept 2012, pp. 1466–1471.

[15] W. Li-hong, T. Hai-kun, and Y. G. hua, "Sensors access scheme design based on internet of things gateways," in *Intelligent Systems Design and Engineering Applications (ISDEA), 2014 Fifth International Conference on*, June 2014, pp. 901–904.

[16] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, "An architecture to support the collection of big data in the internet of things," in *Proceedings of the 2014 IEEE World Congress on Services*, ser. SERVICES '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 442–449. [Online]. Available: http://dx.doi.org/10.1109/SERVICES.2014.83

[17] M. Moreno, J. Santa, M. Zamora, and A. Skarmeta, "A holistic iot-based management platform for smart environments," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 3823–3828.

[18] A. Kouche, "Towards a wireless sensor network platform for the internet of things: Sprouts wsn platform," in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 632–636.

[19] A. Al-Fagih, F. Al-Turjman, and H. Hassanein, "Ubiquitous robust data delivery for integrated rsns in iot," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 280–285.

[20] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of internet-of-things data," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 3185–3190.

[21] Y. Liu, B.-C. Seet, and A. Al-Anbuky, "In-network storage for virtual broker-based publish/subscribe in wsns," in *IEEE Intl Symp on Personal Indoor and Mobile Radio Comm*, Sept 2012, pp. 2553–2558.

[22] D. P. Bertsekas, *Dynamic Programming & Optimal Control, Vol. I*. Athena Scientific, Belmont, MA, 2005.

[23] S. Mao, M. H. Cheung, and V. Wong, "Joint energy allocation for sensing and transmission in rechargeable wireless sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 6, pp. 2862–2875, July 2014.

[24] Y. Liu, B.-C. Seet, and A.-A. A., "Virtual brokers for large-scale publish/subscribe in wireless sensor networks," in *IEEE/IFIP Intl Conf on Embedded and Ubiquitous Comp*, Dec 2010, pp. 240–246.