Virtual Broking Coding for Reliable In-Network Storage on WSANs

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

Stéphane Lohier

HAL Id: hal-01233513

https://hal.science/hal-01233513

Submitted on 25 Nov 2015

Virtual Broking Coding for Reliable In-Network Storage on WSANs

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

* Université Paris-Est, 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 emerging Internet of Things (IoT) paradigm makes Wireless Sensor and Actuator Networks (WSANs) seem as a central element for data production and consumption. In this realm, where data are produced and consumed within the network, WSANs have as a challenge to perform in-network data storage considering their resource shortage. In this paper, we propose the Virtual Broking Coding (VBC) as a data storage scheme compliant with WSANs constraints. As such, VBC ensures a reliable data storage and an efficient mechanism for data retrievability. To evaluate our proposed solution, we present a theoretical analysis as well as a simulation study. Using both, we show that VBC reduces the cost incurred by the coding techniques; and increases the delivery ratio of the requested data. The results presented by VBC suggest this solution as a new direction on how to use network coding based schemes to address the WSAN in-network storage problem.**

I. INTRODUCTION

A wireless sensor network (WSN) typically consists of a set of small sensor devices capable of collecting data from the environment and transmitting this information to a single collection point, the sink node. But, in the context of the Internet of Things (IoT), any informational device is a potential element of our global data communication network. This may change the way we understand and design WSNs. Important examples of the employment of WSN in this new realm are given by the concepts of smart cities and smart homes. In such application domains, the data consumers may be located in the network itself. They work as smart devices which take decisions and/or trigger actions (i.e., behave as actuators). The problem of reliable in-network data storage then arises from the necessity of making the collected data available at any time to these localized controllers and actuators. Therefore, the challenge is to design a robust and reliable data storage scheme for such constrained and automated Wireless Sensor & Actuator networks (WSANs).

Two different approaches to tackle the data storage problem exist in the literature: those based on the Publish/Subscribe paradigm [1]–[7], and those based on coding techniques. The Pub/Sub paradigm defines a content-oriented communication model. In this model, the data consumers must subscribe their interests to a mediation entity, the Virtual Broker (VB), which is responsible for retrieving the requested information from the data sources [8]. These schemes perform better than the solution of sending all the information to a base

station, usually placed on the border of the network. However, since they do not consider the possibility of failures in the virtual broker nodes and the imperfections in the wireless links leading to packet losses, the storage scheme does not ensure the reliability of the content delivery. The solutions based on coding techniques, on the other hand, propose to achieve reliability through the efficient implementation of data redundancy [9]–[13]. These solutions do improve the performance of the data storage system, but they still present a considerable communication overhead for implementation on WSNs.

The aim of this paper is to propose an in-network data storage system for WSANs which ensures that the data remain available and easily retrievable at all times, at an acceptable communication cost. In order to reach this goal, we have designed the Virtual Broking Coding (VBC) scheme taking advantage of the two approaches mentioned above: data storage reliability from coding technique, and low communication cost from a VB-based publish/subscribe architecture. The main obtained results are twofold:

- First, we indeed achieve a substantial increase in the reliability of the stored data when compared with the Balanced Storage system [14], which is adopted as a benchmark in this work. The increased reliability is demonstrated by a theoretical analysis of a simplified scenario, and corroborated by more extensive simulation studies based on the scenarios proposed in [14].
- As one may expect, the improvement in reliability based on redundant coding is associated with a penalty on the communication cost of the system. However, our analysis shows that the overall communication cost of VBC is considerably smaller than the cost of the Decentralized Erasure Coding (DEC) [12], which is one of the most efficient coding-based solutions for the data storage problem. The explanation here is that VBC operates on a structured network. The most impressive result, however, is that VBC improves the cost performance of BS in a well defined sense: it has a smaller communication cost per delivered packet to the consumers. In other words, our theoretical analysis and simulation results show that the improvement in reliability obtained with the introduction of redundancy not only compensates the increase in the

communication cost, but also proportionates some gains in performance.

The rest of the paper is organized as follows. Section II describes the related works and in Section III we present a background related to the technique discussed in this paper. In Section IV we introduce the Virtual Broking Coding (VBC) system. Section V shows a theoretical performance analysis of VBC while Section VI theoretically compares VBC with the related works. Section VII discusses our simulation results and, finally, Section VIII concludes this paper.

II. RELATED WORK

A. Balance Storage System

The Pub/Sub paradigm has been pointed out by [8] as an encouraging solution to face the problem of content delivery in the Internet as envisaged in our days. This paradigm has been explored in the context of WSNs through the introduction of the concept of Virtual Broking (VB). The VB is implemented by a number of virtual broker nodes (VBNs) that store all the data produced in the network, and mediate the communication between publishers and subscribers. As a basic operation made for such an approach, [15] proposes a VB based architecture where the network is divided into a number of small partitions similar to the one illustrated in Figure 1. Virtual broker nodes (depicted in black) are located at the center of partition; and the remaining nodes are the publisher and subscriber nodes.
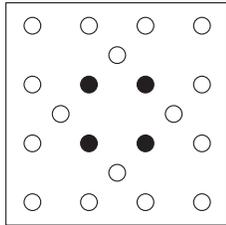

Fig. 1: Network partition.

In the basic operation of the Pub/Sub model, the information producers publish their sensed data on the VBN that is the closest to them. The data consumers also send their subscriptions to the closest VBN. If, upon receiving a subscription, the VBN has the requested packet stored in its memory, then it immediately forwards the packet to the consumer node. Otherwise, the request is broadcasted to the other VBNs in the partition.

Despite all researches on the Pub/Sub paradigm on WSNs [16]–[18], the reliability of the data storage has not received much attention so far. The work we found that is closer to this topic is done by Liu et al. [14] and propose an improvement of the popular Distributed Hash Table (DHT) approach to distribute the data among the VBNs. Their contribution is a dynamic load balancing mechanism, called Balanced Storage (BS). BS transfers packets from an overloaded VBN to another one with more available storage space. In the basic operation of BS, each VBN periodically sends a control message to the neighboring VBNs, containing information about its current storage level. Then, it applies a set of rules to decide whether to forward packets for storage in other VBNs or not. Although the main concern of this work is not reliability, one may argue that, by distributing the packets evenly among the VBNs, BS avoids the risk of losing a large fraction of the packets in case of the failure of an overloaded VBN. However, this is achieved through the exchange of packets among the VBNs, which may lead to packet losses in the links. Moreover, since this system does not implement any mechanism of redundancy, the packets which are eventually lost cannot be recovered and the data become unavailable.

B. Reliable Coding-Based Storage Systems

In order to address the reliability problem, several works propose the implementation of efficient data redundancy schemes using coding technique, such as erasure codes [9], [11], [13]. These solutions, however, adopt a strategy of distribution of coded packets based on random walks, which leads to a high communication cost. In the attempt to reduce this cost, Dimakis et al. [12] propose a mechanism of distributed erasure coding (DEC), inspired on the network coding technique presented in [19]. In the WSN scenario considered by Dimakis et al., there are k source nodes producing one piece of data each. These information are stored in n storage nodes where $n > k$. In the basic operation of DEC, each source node sends its packet through pre-computed routes to a subset of storage nodes, which are randomly chosen but fixed during the operation of the protocol. Then, each storage node produces a single coded packet from the collection of packets that arrive to it, and stores this coded packet in its memory. Finally, a data collector can recover the information produced by the sources by querying any subset of k storage nodes (presumably the closest ones), and decoding the packets obtained from them. Although DEC represents a step forward with respect to previous coding-based solutions, it still presents a considerable communication overhead for implementation on WSNs. The main source of inefficiency comes from the fact that the protocol operates on unstructured networks, where the data sources have to send their packets to storage nodes located all around the network.

III. LINEAR NETWORK CODING BACKGROUND

We use a linear coding methodology on the finite field GF(q), where q is the size of the field. For this purpose, packets are regarded as vectors, $\mathbf{x} = (x_1, \ldots, x_l)$, of elements of GF(q), which can be manipulated and combined through the arithmetic operations defined in the field. Now, suppose that we have N packets $\mathbf{x_1}, \ldots, \mathbf{x_N}$ with the same size l, and let $c_1, \ldots, c_N$ be arbitrary elements of GF(q). Considering that, we can compute a coded packet $\mathbf{y}$ as the following linear combination of the packets $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

$$\mathbf{y} = \sum_{i=1}^{N} c_i \cdot \mathbf{x_i}$$

Intuitively, the packet $\mathbf{y}$ has information about every packet $\mathbf{x_i}$ which is associated with a non-null coefficient c_i, but the

packet $\mathbf{y}$ alone is not sufficient to recover all the original packets $\mathbf{x_1}, \dots, \mathbf{x_N}$. Next, consider an $N \times N$ matrix $\mathbf{C}$ of elements of GF(q), which allows us to compute a collection of N coded packets as linear combinations of $\mathbf{x_1}, \dots, \mathbf{x_N}$:

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

Now, assuming that the coefficient matrix $\mathbf{C}$ is known and has rank N, the system of linear equations

$$\mathbf{Y} = \mathbf{C} \cdot \mathbf{X}$$

where $\mathbf{Y} = [\mathbf{y_1}, \dots, \mathbf{y_N}]$ and $\mathbf{X} = [\mathbf{x_1}, \dots, \mathbf{x_N}]$, can be solved uniquely in order to recover the packets $\mathbf{x_1}, \dots, \mathbf{x_N}$.

IV. VIRTUAL BROKING CODING (VBC)

Our proposal, VBC, is a system based on Linear Network Coding. In VBC, coded packets are stored in a redundant way. In order to introduce redundancy in our scheme, VBC needs to generate M coded packets from N data packets, where $M > N$. Formally, we assume an $N \times M$ coefficient matrix $\mathbf{C}$. These allow us to compute a collection of M coded packets as linear combinations of the N original packets $\mathbf{x_1}, \dots, \mathbf{x_N}$:

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

In this context, the desirable property of the redundant coding scheme is that any subset of N coded packets can be used to recover the original packets. In mathematical terms, this requires that every $N \times N$ sub matrix of the coefficient matrix $\mathbf{C}$ must have rank N. Assuming this condition and a VB based Pub/Sub architecture, we implement our scheme of redundant codification in the VB, instead of implementing it in all nodes as this is the case of traditional coding based approaches. This requires intrinsic modifications in the publish and subscribe operations. In the publish operation, each VBN now waits until it has received a window of N packets, which may come from any data source. When the window is complete, the VBN produces M coded packets, stores one or more of them in its own memory and distributes the rest among the other VBNs in the same partition. We define the relation M/N as the redundancy level of VBC. It means that the greater the difference between M and N, the bigger the number of redundant packets stored in the VB. Typical values for the redundancy level M/N in the scenario shown in Figure 1 are $4/3$, $4/2$ and $3/2$. When the number of coded packets is smaller than the number of VBNs, the destination of the coded packets is chosen randomly. Figure 2 illustrates the steps involved in the publish operation using redundancy level $4/3$.

In the subscribe operation, when a VBN receives the request for a given packet p, it broadcasts the request to the other VBNs. Then, any VBN that holds a coded packet that was produced from p in its memory, sends this coded packet directly to the subscriber. The subscriber is responsible for performing the decodification of the coded packets, as described in Section II-B, which is always pos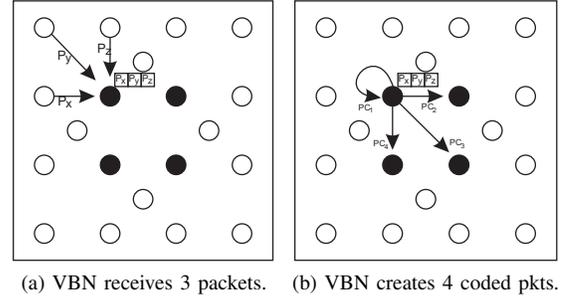sible if at least N coded packets are received. Figure 3 illustrates the steps involved in the subscribe operation using a redundancy level of $4/3$.

(a) VBN receives 3 packets. (b) VBN creates 4 coded pkts.

Fig. 2: Publish operation

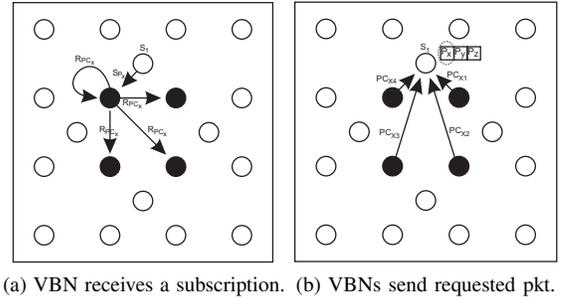(a) VBN receives a subscription. (b) VBNs send requested pkt.

Fig. 3: Subscribe operation

These modifications aggregate a number of good properties to the protocol. First of all, the way how coded packets are distributed naturally achieves an uniform balance of the overall storage load among the VBNs. For this reason, we do not need to execute a load balancing procedure as performed in [14]. Second, the implementation of the redundant codification of data packets makes the system tolerant to the failure of one or more VBNs (depending on the redundancy level which is being adopted). That is, in case of failure of a VBN, all the data stored in the VB remain available to the information consumers. Another advantage of the proposed system is the improvement on the reliability of the content-delivery to the data consumers. Consider, for example, the situation where the system is using a redundancy level of $3/2$. This means that for each request for a packet p stored with redundancy, the VBNs send three coded packets back to the consumer. The point, though, is that only two of these packets need to arrive at the destination in order to allow the consumer to recover the requested packet p. In addition to the robustness against some packet losses, we can also restore the redundancy that was lost due to the failure of a VBN. This can be achieved by producing copies of the packets that were stored in the faulty VBN, using the coded packets stored in the other VBNs. As we will see in the following section, this design choices has a considerable impact on the ratio of requests completed by the proposed solution compared to other existing schemes.

V. Performance Analysis of VBC

We aim with this analysis to show the impact of the redundancy scheme proposed by VBC on the packet delivery probability. In order to do so, we consider two different situations: an error-prone communication channel (EPCC) situation - where the packets might be lost due to the unreliable wireless links, with a probability σ; and a failure-prone VB (FPVB) - where besides the packet loss probability σ, one VBN fails.

A. Model assumptions

As we present a comparison with a benchmark that uses the same Pub/Sub architecture, we assume the scenario adopted for them in [14]. So, we focus on a single partition (see Figure 1), with four VBNs, one data source s which sends packets to let us say VBN_1; and one data collector t which sends requests to VBN_3. We assume a redundancy level of $3/2$, i.e. for each two data packets received by VBN_1, three coded packets are generated. One of these coded packets is stored in VBN_1, and the other two are distributed to randomly chosen VBNs. In the FPVB situation, we further assume that VBN_2 fails (it could be VBN_4 as well).

B. Analysis

We start calculating the probability of losing packets in the publish operation. In EPCC, we have that a packet may be lost due to the unreliability of the wireless channel. In FPVB, on the other hand, besides the probability σ of losing a packet, a packet is definitely lost if VBN_1 sends a coded packet to VBN_2. Taking this into account, standard calculations give us that

- Two packets are lost with probability:

  Error-prone communication channel: $a_1 = \sigma^2$
  Failure-prone VB: $a'_1 = \frac{2\sigma}{3} + \frac{\sigma^2}{3}$

- None of the packets is lost with probability:

  Error-prone communication channel: $a_2 = (1-\sigma)^2$
  Failure-prone VB: $a'_2 = \frac{1}{3} - \frac{2\sigma}{3} + \frac{\sigma^2}{3}$

- Exactly one packet is lost with probability:

  Error-prone communication channel: $a_3 = 2(1-\sigma)\sigma$
  Failure-prone VB: $a'_3 = \frac{2}{3} - \frac{2\sigma^2}{3}$

Now, in the subscribe operation, we compute the probability that a request for a packet p that arrives at the VB ends up with the delivery of the packet to the subscriber. Again, there are three cases and standard calculations give us that

- If the packet is unavailable, this probability is 0 in both situations. ($b_1 = 0$ and $b'_1 = 0$).

- If the packet is stored with redundancy, the probability is:

  Error-prone communication channel:

$$b_2 = \frac{2((1-\sigma)^5 + 3\sigma(1-\sigma)^4) + 2\sigma(1-\sigma)^3}{3}$$
$$+ \frac{(1-\sigma)^6 + 3\sigma(1-\sigma)^5 + 3\sigma(1-\sigma)^4}{3}$$

Failure-prone VB:

$$b'_2 = \frac{(1-\sigma)^3 + (1-\sigma)^4}{3}$$
$$+ \frac{(1-\sigma)^5 + 3(1-\sigma)^4\sigma + 2(1-\sigma)^3 * \sigma)}{3}$$

- And if the packet is stored without redundancy, it is:

  Error-prone communication channel:

$$b_3 = \frac{(1-\sigma)^3 + 2(1-\sigma)^4}{3}$$

Failure-prone VB: $b'_3 = \frac{(1-\sigma)^3 + (1-\sigma)^4}{3}$

So, the probability that a request that arrives at the VB for a packet which has also reached the VB is delivered to the subscriber is, in the EPCC situation,

$$\sum_i a_i b_i = \frac{-2\sigma^8 - 20\sigma^7 + 83\sigma^6 - 184\sigma^5 + 233\sigma^4 - 164\sigma^3 + 53\sigma^2 - 3}{3} \quad (1)$$

and in the FPVB situation,

$$\sum_i a'_i b'_i = \frac{2\sigma^7 - 14\sigma^6 + 37\sigma^5 - 41\sigma^4 + 4\sigma^3 + 32\sigma^2 - 27\sigma + 7}{9} \quad (2)$$

C. Comparison of Results

Figure 4 displays the packet delivery probabilities calculated above, varying the probability σ of a packet loss in a transmission from 0 to 0.6. Similar calculations (omitted due to space constraints) give the delivery probabilities for VBC under redundancy level 4/2, which are also shown in the graph. For the purpose of a comparison, we also present the delivery probability associated with the BS system in the EPCC situation. We did not evaluate the performance of BS in the FPVB situation because in this case the system simply collapses: since there is no mechanism of failure detection in BS, the other VBNs keep an outdated description of the storage level of VBN_2, and the load balancing procedure eventually transfers all the newly arrived packets to the failed VBN, losing them all. We can see in the graph of Figure 4 that the performance of VBC, with both redundancy levels, is clearly superior than that of BS in the EPCC situation in the most relevant range of the probability σ. For instance, with $\sigma = 0.15$, VBC with redundancy 3/2 delivers 25% more packets than BS, and VBC with redundancy 4/2 delivers 50% more packets than BS. What is more remarkable, though, is the fact that, for moderate values of the loss probability σ, the performance of VBC with redundancy 4/2 in the FPVB situation is comparable with the performance of BS in the EPCC situation. We do not see the same level of performance with redundancy 3/2 because in this case VBC is effectively operating without redundancy most of the time: when a coded packet is sent to the failed VBN the redundancy is lost.

VI. Communication Cost Analysis of VBC

As depicted Section V, the redundancy introduced by VBC results in a high packet delivery ratio. However, it has a cost: the increase in the number of transmission in the network. In this section, we theoretically analyze this cost; and show in a comparison to DEC and BS that VBC largely pays off its cost.
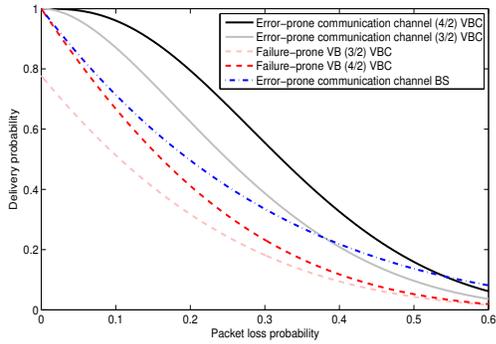
Fig. 4: Packet delivery probability.

A. Comparison of Coding Schemes

Both VBC and DEC achieve reliable in-network data storage through the implementation of redundancy schemes via coding techniques. However, the specific design features of the two systems make them appropriate for different WSN scenarios. The first thing to note is that DEC was designed to operate on large unstructured networks. Indeed, the probabilistic guarantees offered by the authors in [12] are supposed to hold when the number of nodes is very large (see Theorem 3, pg 7 in [12]). VBC, on the other hand, was designed to operate on networks organized into small partitions, as described in [14], and it takes advantage of this structure to: (1) offer a more flexible data storage service, and (2) at a lower communication cost. The important issue here is that, although DEC performs a decentralized coding procedure, the code that is produced is a global code. As a consequence, in order to recover any piece of information produced in the network, a collector node must retrieve as many coded packets as the number of source nodes, otherwise it cannot perform the decoding. VBC, on the other hand, produces local codes that correspond to small generations of packets that are produced in the same partition. As a consequence, the number of coded packets that must be retrieved by a collector node is proportional to the amount of information it is interested in. So, our target here is to prove that when we are not interested in the recovery of all information produced by the network at the same time, which is often the goal in practice, the all-or-nothing model of access to the data implemented by DEC is far more expensive in communication cost than VBC.

1) Assumptions: In the comparison of the communication costs of VBC and DEC, we consider the simplified scenario discussed in [12]: a grid sensor network with N nodes, and $k = O(\sqrt{N})$ data sources producing one piece of information each.

2) Analysis: According to the definition of DEC, the data produced by the sources are stored in $n = O(\sqrt{N})$ storage nodes, with $n = ck$ for some constant $c > 1$. The location of the storage nodes is arbitrary. During the publish operation, each data source sends its packet to a group of $5\frac{n}{k}\ln k$ randomly selected storage nodes. Under the assumption that the average distance between a data source and a storage node

is about $\frac{\sqrt{N}}{2}$, this corresponds to a total of $O(\sqrt{N}) * 5\frac{n}{k}\ln k * \sqrt{N}/2 = O(N \ln N)$ hop-to-hop packet transmissions. For the subscribe operation, a data collector must contact at least k storage nodes and obtain one coded packet from each of them. At this time, the data collector may choose to contact the storage nodes that are closest to it (that is, it does not have to choose them randomly, as in the publish operation). For this reason, the total communication cost of the subscribe operation may range from $O(\sqrt{N})$ hop-to-hop packet transmissions, in the best case where the distance between the data collector and each storage node is $O(1)$, to $O(N)$ transmissions, under the assumption of an average distance of $\frac{\sqrt{N}}{2}$ between the data collector and the contacted storage nodes.

Let us consider now the communication cost of VBC in the same scenario. In the publish operation, each of the $O(\sqrt{N})$ data sources sends one packet to a local VBN, in a total of $O(1) * O(\sqrt{N}) = O(\sqrt{N})$ hop-to-hop transmissions. Then, the VBNs that received packets produce local codes, each corresponding to a small generation of packets, and distribute coded packets to the other VBNs in the same partition. This step also involves $O(\sqrt{N})$ hop-to-hop transmissions, and so the total communication cost of the publish operation in VBC is $O(\sqrt{N})$. For the subscribe operation, we consider first the case where the data collector is interested in all the information that was produced in the network. In this case, the collector has to contact $O(\sqrt{N})$ VBNs which, according to the partition scheme adopted by VBC, are scattered all around the network. Under the assumption of an average distance of $\frac{\sqrt{N}}{2}$ between the data collector and the contacted VBNs, we have a total communication cost of $O(\sqrt{N}) * \frac{\sqrt{N}}{2} = O(N)$ hop-to-hop transmissions. However, the important advantage of VBC is that it allows the data collector to retrieve any portion of the data that was produced in the network. So, assuming that the collector is interested only in the information produced, say, by m data sources, then it needs to contact just $O(m)$ VBNs to obtain coded packets from them, and the total communication cost of the subscribe operation is reduced to $O(m * \sqrt{N})$.

In summary, the total communication cost (i.e. for both the publish and subscribe phases) of DEC and VBC are $O(N ln N)$ and $O(N)$, respectively. Indeed, this result proves that the flexibility offered by VBC, which allows data collectors located in any place of the network to have direct access to their desired data, incurs a cost that is proportional to the amount of information retrieved. As a consequence, this makes VBC more suitable for the paradigm of autonomous WSANs, where collectors are also spread out around the network and typically interested in local information only.

B. Comparison of Storage Systems on Structured Networks

Now, let us analyze VBC and BS and compare them in terms of a metric called cost-benefit. The cost-benefit is defined as the ratio of the communication overhead to the number of packets delivered to the subscribers. We have that as a consequence of the redundancy mechanism introduced in VBC, its cost in packet transmission increases. However, with the cost-benefit metric we aim to show that its gain

in delivering packets offsets its cost. It is worth to mention that, even it is not addressed explicitly, this also has a benefit in terms of energy waste as we assume that the energy consumption is directly related to the communication cost. This assumption is indeed justified as according to [20], the computational overhead incurred by the coding technique lies in the decoding process. As it happens, the subscribers are the responsible by the decoding process. Besides, we envisage application situations where the subscribers are more powerful devices, such as cellphones and actuator devices. This way, the computational overhead does not have much impact in the VBNs that perform relatively simple operations with low costs. Thus, it is the transmission cost that makes the difference in the energy consumption.

In order to estimate the cost-benefit, we need to analyze both the packet delivery ratio and the communication cost of VBC and BS. As we already computed the packet delivery probability of both solutions in Section V-B, now we calculate only their cost.

1) Assumptions: We assume the scenario showed in Figure 1 with one publisher and one subscriber as well as an observation time τ. We further assume that the source s is close to VBN_1 and generates packets at the rate of r_1 pkts/sec; and that the collector t is close to VBN_3 and generates requests at the rate of r_2 rqts/sec. Besides, we define the cost of transmitting a packet inside the VB as x, and from the VBNs to the subscribers as y. In the analysis of VBC, we consider a redundancy level of $3/2$. Regarding BS, we further assume the cost of the load balancing procedure as a cost of the publish operation. In the following we make the analysis for the EPCC situation. We do not make it for FPVB because, as explained in Section V-B, it is not possible to compute the packet delivery probability of BS for this case.

2) Analysis: In this analysis, we assume a probability σ of a packet loss during a transmission. We calculate the communication cost considering only the packets that arrived in the VB that are $\tau r_1 \cdot (1 - \sigma)$. Starting with the communication cost of VBC in the publish operation, we have that 2 coded packets are distributed with cost x. Thus, the cost is defined as $\tau r_1 (1 - \sigma)x$.

In the subscribe operation of VBC the cost is given by the addition of two different costs. The first one, it is the cost associated to the broadcast sent by the VBN_3, which costs z. The other one, it is the cost associated to the answer sent by the VBNs that received the broadcast with the packet request, which costs y. However, in this case, we do not know how many VBNs send a response to the collector. In order to find this out, we calculate the mean value associated to all VBNs that send the requested packet. Assuming the probability of losses σ, we have the probability $(1 - \sigma)^2 \cdot 3$. It gives the cost incurred for one request that arrives in the VB. So, the cost of the subscribe operation is obtained by multiplying this cost for the total of requests that arrives in the VB, which is $\tau r_2 \cdot (1 - \sigma)$. Thus, the total cost for VBC is:

$$\tau r_1 (1-\sigma)x + \tau r_2 (1-\sigma)(z + 3y(1-\sigma)^2) \quad (3)$$

Let us now analyze the communication cost of BS. In the publish operation, as only $3/4$ of the packets that arrive to the VB will be distributed, we have a cost of $3\tau r_1 (1-\sigma)x/4$. Besides, we have to add the cost of the broadcast message sent for all VBNs at each 3s, which gives $\tau 4/3$. In the subscribe operation, we have that $\tau r_2(1 - \sigma)/4$ requested packets are delivered with a cost y as these are directly transmitted from the VBN that received the request. To the other $3\tau r_2(1-\sigma)/4$ requested packets, we have to add the cost z of broadcasting the request to the VB. We also add the cost y of answering the request when a VBN has the requested packet, which means $3(\tau r_2(1 - \sigma)^3)$ requests. Thus the total communication cost is given by:

$$\frac{3\tau r_1(1-\sigma)x + \tau r_2(1-\sigma)y + 3\tau r_2 z + 3\tau r_2(1-\sigma)^3 y}{4} + \frac{4\tau}{3} \quad (4)$$

Finally, by computing the ratio between the cost and the delivery probability computed in Section V-B, we obtain the cost-benefit of both schemes. Table I presents the values of the parameters used to produce the curves in Figure 5. We do not show the results with $r_1 = 1/5$ due to space constraints and the fact that it presents similar results. The conclusion is that in a realistic situation, with packet loss, VBC largely pays off its communication cost. Indeed, in the EPCC situation with $\sigma = 0.5$, BS doubles the cost of VBC to deliver one requested packet. This means that BS makes more transmissions in order to deliver a packet than VBC, resulting in more energy waste as a consequence. Besides, we see that even in an ideal situation (EPCC when $\sigma = 0$), there is a small difference between the cost-benefit of both schemes. Indeed, VBC decreases its cost-benefit from $\sigma = 0$ to $\sigma = 0.2$, remaining stable up to $\sigma = 0.5$, where both schemes start to lose an important quantity of packets. BS, on the other hand, presents a rising curve from $\sigma = 0$, overcoming the cost of VBC from $\sigma = 0.1$ on.

Parameters	Values
τ	$150s$
r_1	$1/3$
r_2	$1/3$
x	1
y	1
z	3

TABLE I: Parameters of the cost-benefit estimation.

VII. SIMULATION EVALUATION

In the following, we replicate the situation proposed to the Balanced Storage (BS) scheme in [14] in order to assess the performance of VBC and compare it to the one of BS. As Liu et al. [14], we assume that all nodes have a transmission range of 16m and are placed in a partition (see Figure 1) that has a dimension of 50m x 50m. We implement the two solutions in the Network Simulator version 2 (NS-2) with the AODV and the IEEE 802.15.4 as the routing and the MAC protocols, respectively.
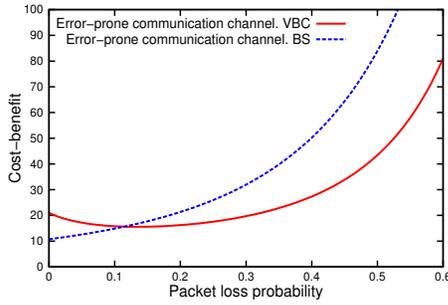
Fig. 5: Cost per packet delivered.

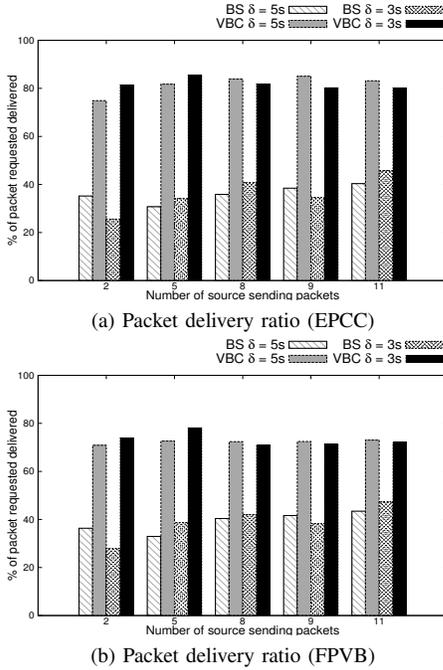(a) Packet delivery ratio (EPCC)

(b) Packet delivery ratio (FPVB)

Fig. 6: Packet delivery ratio.

We consider in this evaluation study, the EPCC situation that is the only situation considered in [14]; and the FPVB, where one VBN chosen randomly stays out of work in the last 50s of the simulation lifetime. The evaluation of VBC and BS in the EPCC situation takes into account three metrics, the packet delivery ratio, the VB communication cost and the relation between the cost and the benefits achieved with the schemes. In addition, their performance in the FPVB situation is also assessed regarding the packet delivery ratio and the cost-benefit metrics.

A. Packet Delivery Ratio

This metric assesses the performance of both schemes in the task of delivering the packets requested by the subscribers. More specifically, in Figure 6a we see the percentage of requested packets that were actually delivered by the two data storage systems, in scenarios where the number of data sources in the partition ranges from 2 to 11 and where all sources generate packets at constant intervals of 3 and 5 seconds. The

graph shows that VBC performs better than BS in all scenarios, delivering up to 200% more packets than BS in the best case. This result is explained by the data redundancy introduced by VBC. With the redundancy level of $3/2$ that we have adopted in these experiments, VBC can handle up to 1 packet loss and still deliver the requested packet (if the loss occurs after the codification). The graph also shows that the performance of both protocols does not degrade when the number of sources is increased. In Figure 6b, we see the results for the experiments where one VBN temporarily fails. Again, we have the same pattern of results with VBC performing better than BS in all scenarios. The remarkable fact here is that VBC can sustain a delivery rate of more than 60% in all scenarios.
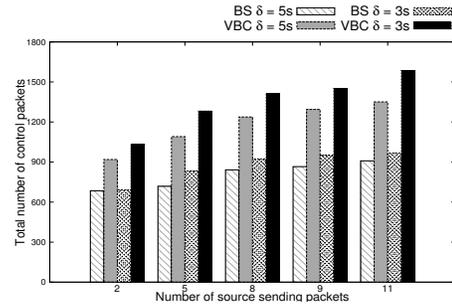
B. VB Communication Cost

Fig. 7: Communication Cost (EPCC)

This metric attempts to capture the communication overhead introduced by VBC and BS, by counting the total number of messages transmitted by the VBNs in both systems. In BS, this number corresponds not only to the transmission of requests and data packets, but also includes the control messages and data transfers associated with the load balancing procedure. In VBC, the main impact on the communication cost is due to the introduction of redundancy and the fact that packets are coded together and distributed to other VBNs. In particular, the later implies that several VBNs are involved in the handling of every request from a subscriber: the request must be forwarded to other VBNs which then transmit their coded packets to the subscriber. Figure 7 shows the simulation results that we obtained using the same scenarios described above. As expected, the operation of VBC is associated with a greater communication overhead than the operation of BS. However, it is important to observe how the difference between the two protocols increases and only becomes relevant when the number of data sources gets large. Indeed, the communication cost of VBC is either only 34% or 49% higher in the scenario with 2 sources and $\delta = 5s$ or $\delta = 3s$, respectively. But, it increases to respectively 48% and 63% in the scenario with 11 sources for the same values of δ. However, as we see next, it turns out that this cost is a small penalty in view of the reward in VBC performance.

C. Cost/benefit

The idea here is to integrate the results of the two previous measures in order to compute the cost-benefit metric. Through

(a) Cost/benefit (EPCC)
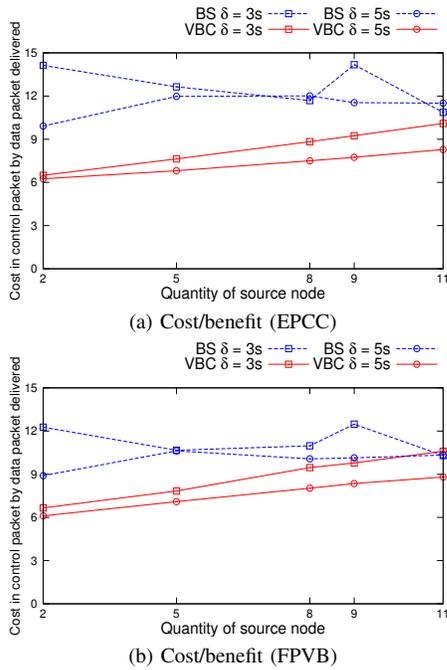

(b) Cost/benefit (FPVB)

Fig. 8: Cost/benefit

this metric we estimate the communication cost per delivered packet associated with each protocol. The results displayed in Figure 8a show a clear advantage for VBC in the situations with no failures. This indicates that the additional effort associated with the redundant codification implemented by VBC is more than compensated by the improvement in the performance of the protocol. When the number of sources in the partition is relatively small (up to 5 or 6, say) the cost per delivered packet in VBC is at least 42% smaller than in BS. When the number of sources increases the difference between the two protocols is reduced, reflecting the fact that those are favorable scenarios for the load balancing procedure implemented by BS. Moreover, Figure 8b shows that even in the case of failure in a VBN (FPVB situation), VBC also has a better performance than BS in all scenarios.

VIII. CONCLUSION

In this paper, we proposed the Virtual Broker Coding system that performs in-network data storage ensuring the availability of the data in the storage nodes and the reliability of the content delivery. To do so, VBC uses Linear Network Coding over structured networks. Both theoretical analysis and the simulation results show that the redundancy introduced by VBC implies in a high packet delivery ratio although it also incurs in an increase in the communication cost. However, we also demonstrated that using a virtual broker structure, VBC accomplishes to overcome this cost. Besides, the cost/benefit metric defined in the simulation study shows that the cost is compensated by the improvement in the performance of the protocol. Therefore, VBC reaches its objective of increasing the data storage reliability, and proves that it is worth using a coding technique when it is assumed a structured network.

The redundancy level

The redundancy level used in VBC involves a performance trade-off, as the improvement in the reliability is obtained at the expense of an increase in the communication cost. We believe that through finding the optimal trade-off according to the application and network conditions, VBC can do even better. The optimization of the redundancy level used by VBC according to this is the subject of our future work.

REFERENCES

[1] M. Matos, P. Felber, R. Oliveira, J. Pereira, and E. Riviere, "Scaling up publish/subscribe overlays using interest correlation for link sharing," *IEEE Trans on Paral and Distrib Sys*, vol. 24, Dec 2013.

[2] C. Tsilopoulos, I. Gasparis, G. Xylomenos, and G. Polyzos, "Efficient real-time information delivery in future internet publish-subscribe networks," in *Intl Conf on Comp, Net and Comm*, Jan 2013, pp. 856–860.

[3] M. Tariq, B. Koldehofe, and K. Rothermel, "Securing broker-less publish/subscribe systems using identity-based encryption," *IEEE Trans on Paral and Distributed Sys*, vol. 25, no. 2, pp. 518–528, Feb 2014.

[4] C. Stais, A. Voulimeneas, and G. Xylomenos, "Towards an error control scheme for a publish/subscribe network," in *IEEE Intl Conf on Communications (ICC)*, June 2013, pp. 3743–3747.

[5] Y. Sun, X. Qiao, B. Cheng, and J. Chen, "A low-delay, lightweight publish/subscribe architecture for delay-sensitive iot services," in *20th IEEE Intl Conf on Web Services (ICWS)*, June 2013, pp. 179–186.

[6] D. Xiao, Y. Zhang, and J. Chen, "A real-time and reliable forwarding scheme of publish/subscribe system," in *Intl Conf on Measuring Technology and Mechatronics Automation*, Jan 2014, pp. 51–54.

[7] C. Seeger, K. Van Laerhoven, J. Sauer, and A. Buchmann, "A publish/subscribe middleware for body and ambient sensor networks that mediates between sensors and applications," in *IEEE Intl Conf on Healthcare Informatics (ICHI)*, Sept 2013, pp. 199–208.

[8] M. Diallo, V. Sourlas, P. Flegkas, S. Fdida, and L. Tassiulas, "A content-based publish/subscribe framework for large-scale content delivery," *Computer Networks*, vol. 57, no. 4, pp. 924 – 943, 2013.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] H. Huang, E. Ngai, and J. Liu, "A location-based publish/subscribe framework for wireless sensors and mobile phones," in *Wireless Comm. and Networking Conf. (WCNC), 2012 IEEE*, April 2012, pp. 2173–2178.

[17] C. Chaudet, I. Demeure, and S. Ktari, "A model to evaluate brokers overlays for publish/subscribe in wireless sensor networks," in *Conf. on Wireless On-demand Net. Sys. and Serv. (WONS)*, Jan 2012, pp. 71–74.

[18] M. Abdullahi and G. Wang, "Secure publish-subscribe-based in-network data storage service in wireless sensor networks," in *IEEE 8th Intl. Conf. on Distributed Comp. in Sensor Sys (DCOSS)*, May 2012, pp. 297–299.

[19] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. on Infor. Theory*, vol. 46, pp. 1204–1216, Jul 2000.

[20] D. Wang, Q. Zhang, and J. Liu, "Partial network coding: Theory and application for continuous sensor data collection," in *IEEE Inter. Workshop on Quality of Service (IWQoS)*, June 2006, pp. 93–101.