

Reliability and cost efficiency in coding-based in-network data storage and data retrieval for IoT/WSNs

Camila Helena Souza Oliveira

▶ **To cite this version:**

Camila Helena Souza Oliveira. Reliability and cost efficiency in coding-based in-network data storage and data retrieval for IoT/WSNs. Computation and Language [cs.CL]. Université Paris-Est, 2015. English. NNT : 2015PESC1134 . tel-01368200

HAL Id: tel-01368200

https://pastel.hal.science/tel-01368200

Submitted on 18 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Paris-EST

École Doctorale MSTIC
Mathématiques et Sciences et Technologies de l'Information et de la Communication

PHD THESIS

Mention: Informatique

defended by

Camila Helena Souza Oliveira

Reliability and Cost Efficiency in Coding-based In-Network Data Storage and Data Retrieval for IoT/WSANs

---

Advisor:

Yacine Ghamri-Doudane

Jury:

M. Marcelo Dias de Amorim	Université Pierre et Marie Curie	Rapporteur
M. Steven Martin	Université Paris-Sud	Rapporteur
M^{me} Véronique Vèque	Université Paris-Sud	Examinateur
M. Karim Djouani	Université Paris Est-Créteil	Examinateur
M. Carlos Fisch de Brito	Federal University of Ceará	Examinateur
M. Yacine Ghamri-Doudane	Université de la Rochelle	Directeur de thèse
M. Stéphane Lohier	Université Paris-Est MLV	Examinateur

9 décembre 2015

Institut d'Informatique Gaspard-Monge (LIGM)
5, bd Descartes
77454 Marne La Vallée Cedex 02

Université Paris-Est
École Doctorale MSTIC
Mathématiques, Sciences et Technologies de l'information et de la communication
5, bd Descartes
77454 Marne La Vallée Cedex 02

Abstract

Wireless Sensor Networks (WSN) are made up of small devices limited in terms of memory, processing and energy capacity. They work interconnected and autonomously in order to monitoring a region or an object of interest. The development of more powerful devices (with new capability such as energy harvesting and acting) at a lower cost allowed the evolution of the WSNs to the Wireless Sensor Actuator Networks (WSANs), as well as made them a crucial element in the emergence of the Internet of Things (IoT). Nonetheless, assuming the new applications and services offered in some of IoT scenarios, new issues arise in the data management that needs to be performed in WSANs. Indeed, in this new context, WSANs have to deal with energy efficiency and the large amount of data, which are now consumed on-demand, while ensuring an efficient data reliability and retrievability.

In the scope of this thesis, we are interested in WSANs data management in the context of IoT realm. Specifically, we approach the problem of in-network data storage by posing the following question: *How to temporarily store data in WSANs so that the data could be easily retrieved by the consumers while ensuring the best trade-off between data reliability and conservation of energy resources?* Foremost, we propose to achieve reliable data storage by using network coding, and assuming a communication model defined by the Publish/Subscribe paradigm (Pub/Sub). We validate the efficiency of our proposal by a theoretical analysis and corroborate it by a simulation evaluation. The obtained results show that our scheme achieves a reliability of 80% in data packet delivery with the best cost-benefit trade-off compared to other data storage schemes.

Aiming to further improve the performance of the data storage scheme proposed in our first contribution, we propose its optimization in order to allow an autonomous and adaptive data storage whereas ensuring an optimal trade-off between reliability and communication overhead. The latter is interpreted as the impact of the data storage scheme on the nodes energy consumption. For the best of our knowledge, our optimized data storage scheme is the only to ensure data reliability while adapting itself according to the service requirements and network conditions. The performance evaluation of our adaptive and autonomous scheme, realized by simulation, shows that the optimization of the data storage mechanism (formulated as a Markov Decision Process (MDP)) according to the network condition allows the consumers to retrieve up to 70% more data packets than our scheme without optimization whereas increasing the network lifetime by 43%.

Finally, after being interested in finding the best trade-off between reliability and cost, in our third contribution, we focus on the performance evaluation of our data storage scheme in not-always-on WSANs, which means sensor networks performing a

node activity scheduling (duty cycle). In fact, in the two first contributions we assumed WSANs wherein the sensor nodes are always-on. However, WSANs based on the IEEE 802.15.4 standard may use its duty cycle mechanism in order to save nodes' energy. In the first part of this contribution, we propose an improvement in the duty cycle mechanism defined in the 802.15.4 standard to make it dynamic and adaptive. The simulation evaluation shows that our solution led in considerable saving in energy costs without compromising the performance of the network in delivering the data packets. In the second part, we evaluate the performance of our data storage scheme assuming that the broker nodes may work under a duty cycle mechanism. We aim with this approach to assess if both mechanisms could coexist positively. The performance evaluation realized by simulation shows that the duty cycle mechanism does not improve the energy saving in the network. However, it also shows that the optimal trade-off between reliability and communication overhead (energy consumption) is not affected. The most important conclusion we can take from this evaluation, though, is that our adaptive and autonomous data storage scheme, besides ensuring the optimal trade-off between reliability and cost, is further compatible with both always-on and not-always-on WSANs.

Keywords

Wireless Sensors Networks; IoT; data storage; network coding; MDP; Pub/Sub.

Résumé

Les réseaux de capteurs sans fil sont constitués de petits dispositifs dont les ressources (mémoire, traitement et énergie) sont limitées. Ils fonctionnent de manière autonome et sont interconnectés pour la surveillance d'une région ou d'un centre d'intérêt. Malgré leurs limitations, leur faible coût et leur facilité de déploiement ont favorisé l'extension de ces réseaux de capteurs sans fil qui sont ainsi devenus des éléments fondamentaux dans l'émergence de l'Internet des objets (IoT). Cependant, l'ouverture dans le contexte de l'Internet des objets à une plus large palette d'applications et de services entraîne de nouvelles problématiques dans la gestion des données issues des réseaux de capteurs. En effet, au sein de l'IoT, les réseaux de capteurs doivent contribuer à la gestion de la grande quantité de données récoltées continuellement et consommées à la demande, tout en assurant la fiabilité des informations et sans pour autant augmenter de façon importante la consommation énergétique du réseau.

Dans cette thèse, nous nous intéressons ainsi à cette gestion des données dans les réseaux de capteurs sans fil intégrés dans un contexte IoT. Plus précisément, nous aborderons la problématique du stockage des données au sein même du réseau de capteurs en se posant la question suivante : *Comment stocker provisoirement les données dans le réseau de capteurs de sorte que ces données soient facilement accessible par les consommateurs tout en assurant le meilleur compromis entre la fiabilité de livraison des données et la préservation des ressources énergétiques des capteurs ?* Il s'agit dans un premier temps de proposer un système fiable de stockage de données basé sur la théorie du codage réseau et sur le modèle de communication "Publish/Subscribe". Le système proposé est adapté à l'architecture des réseaux de capteurs ainsi qu'aux besoins des applications et services IoT localisés. Pour démontrer la validité du système de stockage proposé, des évaluations de performances au travers d'une analyse mathématique et de simulations sont conduites. Celles-ci montrent clairement une augmentation de la fiabilité de la livraison des données aux consommateurs avec un taux de livraison des paquets de 80% en moyenne.

Afin d'améliorer encore plus les performances du système de stockage de données, nous proposons, dans un second temps, l'optimisation du système afin que celui-ci puisse réaliser le stockage des données de manière adaptative et autonome, tout en assurant le meilleur compromis entre fiabilité et coût. Ce dernier se traduit par l'impact du système de stockage sur la consommation d'énergie du réseau de capteurs sans fil. À notre connaissance, notre système est le premier à proposer d'assurer la fiabilité du stockage des données en fonction des demandes des services et des conditions du réseau. L'évaluation des performances, par simulation, de notre système de stockage adaptatif et autonome montre que l'optimisation du stockage des données (formulée sous forme d'un Processus de Décision Markovien (MDP)) selon les conditions de fonctionnement du réseau permet l'accès à 70% de données en plus comparativement au système non-adaptatif proposé précédemment. Ce résultat est obtenu tout en augmentant la durée de vie du réseau de 43%.

Après avoir travaillé sur l'aspect quantitatif des performances du réseau à travers une étude sur le compromis coût - consommation énergétique, nous nous intéressons dans la troisième contribution de cette thèse à l'utilisation de notre système de stockage dans des réseaux de capteurs sans fil disposant de cycles de services (cycle

d'endormissement-réveil) variables. En effet, dans les deux premières propositions nous avons considéré des réseaux de capteurs dans lesquels tous les nœuds restent en éveil tout au long de la vie du réseau. Or aujourd'hui, les réseaux de capteurs reposant sur le standard 802.15.4 peuvent utiliser des cycles de services variables et avoir recours à l'endormissement des nœuds dans le but d'économiser leur énergie. Dans une première partie de cette contribution, nous avons ainsi proposé une amélioration du mécanisme de gestion du cycle de service (duty cycle) du standard 802.15.4 afin de le rendre dynamique et adaptable au trafic réseau. L'évaluation des performances par simulations de l'amélioration proposée montre que celle-ci aboutit à une économie d'énergie très significative tout en permettant au réseau de capteurs sans fil de remplir sa mission de prise en charge du trafic généré. Dans une seconde partie de cette contribution, nous évaluons les performances de notre système de stockage de données dans le but d'évaluer si un tel mécanisme pourrait cohabiter positivement avec un mécanisme de cycle de service variable (condition d'exploitation réaliste du réseau). L'évaluation des performances montre que l'activation d'un cycle de service variable dans le réseau de capteurs n'apporte aucune amélioration au niveau de la consommation énergétique mais que le compromis optimal entre la fiabilité et la consommation énergétique obtenu par notre système de stockage adaptatif et autonome n'est pas non plus affecté, celui-ci est maintenu. Ce resultat nous permet d'affirmer que notre mecanisme de stockage est compatible aussi bien pour des réseaux de capteurs sans cycle d'endormissements que pour des réseaux de capteurs avec cycle d'endormissements.

Mots-clefs

Réseaux de capteurs sans fil ; IoT ; stockage de données ; codage réseau ; MDP ; Pub/Sub.

Contents

Abstract **5**

1 Introduction **13**
1.1 Contributions . . . 14
1.2 Organization . . . 15

2 Background **17**
2.1 Wireless Sensor Networks (WSNs) . . . 17
2.2 IEEE 802.15.4 standard . . . 19
2.2.1 Duty cycle mechanism . . . 19
2.3 WSNs evolution (IoT) . . . 21
2.4 Data management . . . 22
2.4.1 In-Network data storage . . . 23
2.5 Publish/Subscribe paradigm . . . 24
2.5.1 Publish/Subscribe architecture categories . . . 25
2.5.2 Data dissemination . . . 25
2.6 Coding techniques . . . 26
2.6.1 Linear Network Coding (LNC) . . . 27
2.7 Conclusion . . . 28

3 VBC: Virtual Broking Coding data storage scheme **31**
3.1 Motivation and context . . . 32
3.2 Related works . . . 32
3.2.1 Pub/Sub-based storage schemes . . . 33
3.2.2 Reliable coding-based storage schemes . . . 33
3.2.3 Summary . . . 34
3.3 VBC: Virtual Broking Coding data storage scheme . . . 35
3.3.1 Publish operation . . . 35
3.3.2 Subscribe operation . . . 35
3.3.3 Discussion . . . 36
3.4 Theoretical packet delivery ratio analysis . . . 37
3.4.1 Model assumptions . . . 37
3.4.2 Analysis . . . 37
3.4.3 Comparison of results: VBC vs BS . . . 38
3.5 Theoretical communication cost analysis . . . 39
3.5.1 Comparison of Coding Schemes: VBC vs DEC . . . 39

3.5.1.1 Assumptions . . . 40
3.5.1.2 Communication costs analysis . . . 40
3.5.2 Comparison of VB-based data storage schemes: VBC vs BS . . . 41
3.5.2.1 Assumptions . . . 41
3.5.2.2 Error-prone communication channel (EPCC) . . . 42
3.5.2.3 Failure-prone VB (FPVB) . . . 43
3.6 Performance evaluation . . . 44
3.6.1 Simulation settings . . . 44
3.6.2 Simulation results . . . 44
3.6.2.1 Packet delivery ratio . . . 44
3.6.2.2 VB communication cost . . . 45
3.6.2.3 Cost-benefit trade-off . . . 46
3.6.2.4 Storage load balancing . . . 46
3.6.2.5 Delay . . . 48
3.7 Conclusion . . . 50

4 DA-VBC: Dynamic Adaptive VBC Data Storage Scheme 53
4.1 Motivation . . . 54
4.2 State of the art . . . 55
4.2.1 WSAN integration in IoT realm . . . 55
4.2.2 Data storage in IoT/WSANs . . . 55
4.3 System model . . . 56
4.3.1 System definition . . . 56
4.3.2 Storage scheme . . . 57
4.3.2.1 Publish operation . . . 58
4.3.2.2 Subscribe operation . . . 59
4.3.3 Application Example . . . 59
4.4 Theoretical analysis: a generalization . . . 60
4.4.1 Packet delivery probability . . . 61
4.4.2 VB communication Cost . . . 61
4.4.3 Cost-benefit trade-off . . . 62
4.5 Optimization problem formulation . . . 63
4.5.1 Dynamic Optimization Methodology . . . 63
4.5.2 Dynamic Adaptive VBC algorithm (DA-VBC) . . . 64
4.6 Performance evaluation . . . 64
4.6.1 Simulation settings . . . 64
4.6.2 Simulation results . . . 66
4.6.2.1 Comparison of π^{mdp} and fixed redundancy levels . . . 66
4.6.2.2 Comparison of π^{mdp} and heuristic-based redundancy levels . . . 68
4.7 Conclusion . . . 69

5 Node Activity Scheduling and In-Network Data Storage: Impact of Introducing duty-cycling on DA-VBC 71
5.1 Context and issues . . . 72
5.2 Related works . . . 73

5.3 Duty cycle adaptation in traditional WSN . . . 74
5.3.1 Problem description . . . 74
5.3.2 DBSAA: a self-adaptive duty cycle approach . . . 75
5.3.3 DBSAA algorithm . . . 77
5.3.4 Performance evaluation . . . 78
5.3.4.1 Simulation setting . . . 78
5.3.4.2 Simulation results . . . 80
5.4 Duty cycle in DA-VBC-based WSANs . . . 84
5.4.1 Theoretical analysis of the impact of duty cycling in a VB structure 84
5.4.2 The DCDA-VBC approach . . . 87
5.4.3 Performance evaluation . . . 88
5.4.3.1 Simulation setting . . . 88
5.4.3.2 Simulation results . . . 88
5.5 Conclusion . . . 90

6 Conclusions **93**
6.1 Perspectives . . . 94

List of Figures **115**

List of Tables **117**

Bibliography **119**

Chapter 1

Introduction

NOWADAYS, we witness a revolution in the way we interact with the world through technology. The development of devices integrated in the objects used in our everyday life changes this interaction through the creation of applications and services before inconceivable.

Wireless Sensor and Actuators Networks (WSANs) are in the center of this revolution benefiting from the evolution in the technologies behind the miniaturization of devices which are based on micro electromechanical system (MEMS). These devices can be used to measure a variety of physical, chemical and biomass values, such as pressure, stress, light, sound, magnetisms, pH values, among others; and to take actions that are triggered depending on the measured values. All these capabilities allow the use of WSANs in different domains such as industrial automation, healthcare, agriculture, military applications, vehicular automation and intelligent transport action system, environmental monitoring, etc. In fact, the variety of application scenarios where WSANs may be employed, as well as the evolution of more powerful sensor devices, made the WSANs a crucial factor in the emergence of the Internet of Things (IoT). However, with the ability of the sensor nodes in performing an autonomous and direct interaction with the environment, the applications and services idealized for the IoT realm got much more complex than those idealized for traditional WSNs.

In order to deal with the requirements of such new set of services and applications, the WSANs have to change their way of doing data management. Now, they have to enable a smooth, reliable and cost efficient communication between the data producers and the data consumers in order to ensure the proper operation of the new services. Besides, these services also need to have access to the *fresh* data generated in the network. Thus, in order to make the data available and quickly retrieved by the consumers, the in-network data storage approach presents itself as the most suitable solution. Nonetheless, despite all the advances underwent by the sensors, controllers and actuators, the WSANs remain resource lacking networks (with respect to energy, memory and computation power) that now have to face the challenge of dealing with a huge volume of data of different types.

1.1 Contributions

In this thesis we are interested in the control of energy consumption in the sensor nodes and in the in-network data storage in the context of the IoT realm. The latter requires high reliability guarantees in data management, a fault tolerant data retrievability, an operation mode able to adapt itself according to the application needs and network condition, as well as a worthwhile communication model in terms of energy consumption. To meet these requirements, through the three contributions of this thesis, we provide answers to the questions: *How to do reliable data storage? How to do in-network data storage ensuring the optimal trade-off between reliability and energy consumption? And how duty cycle mechanisms do impact on the optimal trade-off reached by such a data storage scheme?*

In our first contribution, we propose an in-network data storage scheme for WSANs named Virtual Broking Coding (VBC). Aiming at answering the question of *How to do reliable data storage?* we propose to store redundant coded packets generated through a linear network coding technique. We assume a sensor network divided in small partitions which have a Virtual Broker (VB) structure made up of broker nodes responsible for storing the data packets and intermediate the communication between data producers and data consumers. The data are disseminated and retrieved according to the Publish/Subscribe communication model. The use of a coding technique introduces redundancy in the network without taking more memory space in an individual node, as well as providing data reliability ensuring a low packet loss ratio. We show through a theoretical analysis that our coding-based data storage scheme achieves data reliability whereas decreasing the communication cost usually associated with this type of solutions, resulting in a worthwhile cost-benefit solution. In addition, our theoretical results are corroborate by simulation experiments.

Our second contribution constitutes the main contribution of this thesis. After having demonstrated that it is worth using linear network coding to ensure data reliability, as long as it is assumed a VB-based Pub/Sub model in the network, we aim to go further and make our data storage scheme a suitable solution for IoT applications and services. In our first contribution, we assume a fixed redundancy level since we are interested only in proving the advantages of our method. In the IoT realm, however, we are in a dynamic and decentralized environment that serves as a basis to the implementation of a variety of services, that may be localized or not and work according to different requirements. To face that challenge, we start by designing a system that defines a communication model that integrates our scheme in the WSANs/IoT scenario, as well as a dynamic and autonomous data storage scheme called Dynamic Adaptive Virtual Broking Coding (DA-VBC). 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 generalize the expression found in our first contribution, and then model the problem of choosing the optimal redundancy level as a Markov Decision Process (MDP) problem. Our performance evaluation shows that the self-adaptation aspect as well as an optimization technique that take into account the probability of future changes in the network conditions result in substantial improvement in the data packet delivery ratio and network lifetime.

Although the energy consumption has been taken into consideration indirectly,

given that we assume communication overhead as a metric to asses the impact of our solution on the energy consumption, we did not specifically focus on energy saving mechanisms (we assume always-on WSANs in our experiments). However, as most of WSANs use the IEEE 802.15.4 standard, and it offers the possibility of performing node activity scheduling in order to save node's energy, it is important to evaluate our data storage scheme also in this case. Thus, in our third contribution, we first study the impact of a duty cycle mechanism in the general case of WSANs (i.e. WSANs without a data storage scheme). To do so, we propose a dynamic and self-Adaptive solution called Dynamic Beacon interval and Superframe Adaptation algorithm (DBSAA) that makes the duty cycle mechanism of the 802.15.4 standard performing according to the network traffic. The results obtained by simulation show that DBSAA improves the energy saving in the network as well as increases the network performance in terms of packet delivery ratio. Then, in the second part, we introduce a duty cycle mechanism in our VB, by changing the MDP defined in our second contribution, in order to evaluate the impact of such a mechanism in the trade-off optimized by the DA-VBC. In this case, the performance evaluation does not show any improvement in the energy saving, but let clear that, independent of the network performs under a duty cycle mechanism or not, DA-VBC ensures the optimal trade-off between reliability and energy efficiency (i.e. communication cost).

1.2 Organization

This thesis consists of six chapters that follow the logical organization of the progress of our contributions in the improvement of the in-network caching of the IoT data in WSANs.

We present in the Chapter 2 the Wireless Sensor Networks, their characteristics and operation modes. We further detail the standard that defines their physical and MAC layers, and its duty cycle mechanism, given that these definitions are necessary to the understanding of our work. Then, we describe the evolution of the WSNs toward being a cornerstone in the IoT realm, the challenges emerged thereof, as well as the proposed solutions to overcome these challenges and their drawbacks.

In Chapter 3, we present our first contribution, a reliable coding-based data storage scheme. We present the related works used as our benchmarks and their drawbacks. Then, we propose our scheme named VBC as well as its operation mode, and discuss how it overcomes the deficiency of previous solutions. We analyze the impact of using linear network coding in our scheme regarding the packet delivery ratio and communication overhead, and compare it, theoretically and by simulation, to related works.

Chapter 4 presents our second contribution, a dynamic and adaptive data storage scheme suitable to the IoT realm. We first detail the communication system that allows the introduction of VBC in a WSANs/IoT scenario. Then, we generalize our theoretical analysis and define a Markov Decision Process that models the problem of determining the optimal redundancy level of VBC. The optimization is based on the cost-benefit trade-off incurred by VBC, according to the application requirements and the changes in the network conditions. Next, we present the algorithm DA-VBC that allows our data storage scheme to operate in an optimal state, self-adjusting its

behavior when necessary. We also provide a performance evaluation where DA-VBC is compared with different heuristics, as well as with different metrics chosen as a basis to the optimization.

In Chapter 5, we present a study of duty cycle mechanisms aiming to assess the network performance when such a mechanism coexists with our dynamic and adaptive storage scheme. We start our study with an evaluation of the Dynamic Beacon interval and Superframe Adaptation Algorithm (DBSAA) that we propose in order to make the duty cycle mechanism of the IEEE 802.15.4 standard adaptable and dynamic. Then, in the second part, we present the performance evaluation of DA-VBC when it assumes that the broker nodes may operate under a duty cycle mechanism.

We conclude this thesis by recalling our contributions and the results thereof, as well as some perspectives arisen from our work.

Chapter 2

Background

Contents

3.1 Motivation and context 32
3.2 Related works 32
3.3 VBC: Virtual Broking Coding data storage scheme 35
3.4 Theoretical packet delivery ratio analysis 37
3.5 Theoretical communication cost analysis 39
3.6 Performance evaluation 44
3.7 Conclusion 50

Today, Wireless Sensor Networks (WSNs) are not considered as isolated networks responsible only for collecting the data monitored and transmitting them to a base station. On the contrary, they may now be embedded in a larger realm, called Internet of Things (IoT), that integrates different technologies and allows the implementation of innovative applications and services. Aiming to provide an efficient performance of these new services and applications, the IoT boosted the evolution of the functionality of the WSNs, requiring them to do more than collecting and transmitting data.

In this chapter, we aim at providing an overview of the WSNs as well as the background needed to better comprehend the contributions presented throughout this thesis. We start by briefly describing the characteristics and the main standards related to WSNs, and then we approach the integration of the Wireless Sensor Actuator Networks (WSANs) in the IoT realm along with the issues incurred from it. We focus, mainly, on the in-network data storage and the energy consumption issues (the latter, still a sensible point in the WSN). Thereafter, we present the techniques used to ensure a good performance of the WSANs taking into account the problems approached in the context of this thesis.

2.1 Wireless Sensor Networks (WSNs)

The WSNs consist of autonomous and independent devices that are able to self-organize. They use different sensors to monitor the environment, collecting and transmitting information such as temperature, pressure, movement, etc. Although initially

motivated by military applications, with the evolution of the physical devices that are now much smaller and cheaper, WSNs are employed in several civilian applications, such as industrial automation, health monitoring, traffic control, and smart cities. The operation of the WSNs is based on the cooperation between sensor nodes that are basically composed of four parts: the power source, the micro controller, the sensors and the wireless transceiver. Traditionally these devices are responsible for collecting the monitored information and transmitting it hop by hop up to a sink/gateway that sends the information through the Internet to the end user or consuming service, as showed in Figure 2.1.

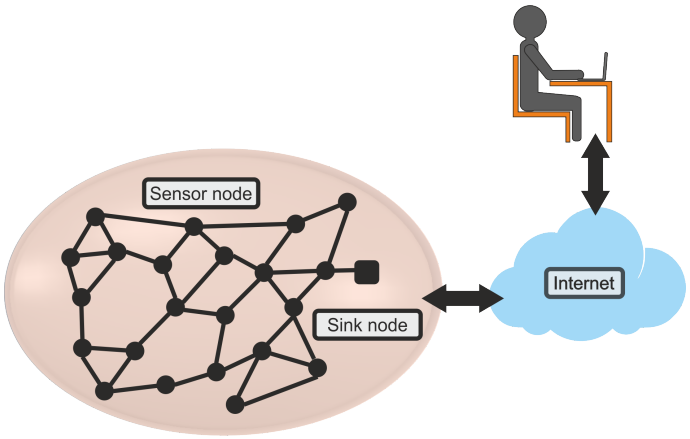


Figure 2.1: Traditional Wireless Sensor Network.

The sensor nodes are by definition devices with strict limitations in terms of processing, memory and energy. Among these three limitations, the energy shortage presents the most crucial problem in keeping the network operational. For this reason, there was a great evolution in the development of energy harvesting technologies, such as piezoelectric materials, micro oscillators, thermoelectric power generation elements and electromagnetic energy [76, 70, 21, 13], that gave to the sensor nodes the capacity of performing ambient energy harvesting from different sources of energy, for instance, those showed in Figure 2.2. Also taking into consideration the energy limitation, in the last 30 years, the research in WSNs focused on low energy consumption network technology, and data processing and routing algorithms. As a result, today we have a widespread and adapted standard that defines the protocols of the physical and MAC layers in WSNs: the IEEE 802.15.4 [1].



Figure 2.2: Ambient energy harvesting devices.

2.2 IEEE 802.15.4 standard

The IEEE 802.15.4 standard defines the physical and MAC layers of WSNs in order to deal with sensor node's limitations (energy, memory and processing). To do so, it specifies the topology, the operation mode, the network capacity and the channel access protocol (CSMA/CA). In the IEEE 802.15.4-based WSNs, the nodes can be classified as Full Function Devices (FFD) or Reduced Function Devices (RFD). The FFD node can function as a personal area network (PAN) coordinator, as a router or as a regular device. Besides, it can communicate with any other device in the network, as long as the device is in its radio range. The RFD nodes are more limited computationally and can communicate only with an FFD node.

According to IEEE 802.15.4 standard, every WPAN (Wireless Personal Area Network) has at least one FFD node operating as a PAN coordinator. The IEEE 802.15.4 standard defines also two different types of topology: star and peer-to-peer (Figure 2.3). In the star topology, the communication is established directly to the coordinator, and the ordinary nodes (RFD) cannot communicate with each other. In the peer-to-peer topology, the FFD node can communicate with all other nodes, while the RFD node is able to communicate only with their coordinator. The network capacity depends on the chosen frequency band, it can be 20Kb/s (868MHz), 40Kb/s (902-928MHz) and at most 250Kb/s(2.4GHz).

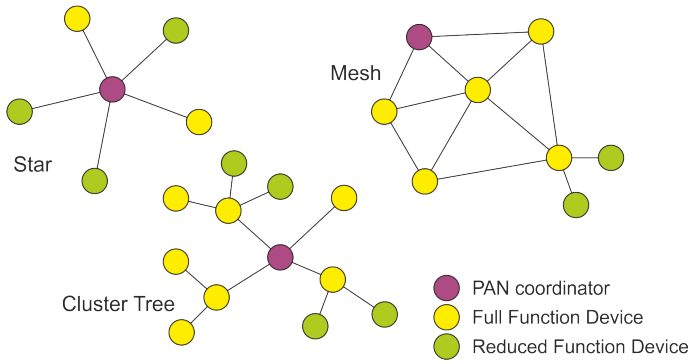


Figure 2.3: topology

Concerning the IEEE 802.15.4 MAC layer, it can operate in either beacon-enabled or non-beacon-enabled modes. The difference between the two modes lies in the fact that in the beacon-enabled one the sensor nodes work under a duty cycle mechanism. The duty cycle is a mechanism that allows the nodes to take turns changing from active mode to sleep mode, and vice-versa. The duty cycle performed by the nodes in the beacon-enabled mode of 802.15.4 allows them to save energy and increase the network lifetime. As the energy saving is one of the issues approached in this thesis, we are only interested in the beacon-enabled mode.

2.2.1 Duty cycle mechanism

As can be observed in Figure 2.4, in the beacon-enabled mode, the network operation timeline is divided into consecutive time intervals called Beacon Intervals (BI). The BI

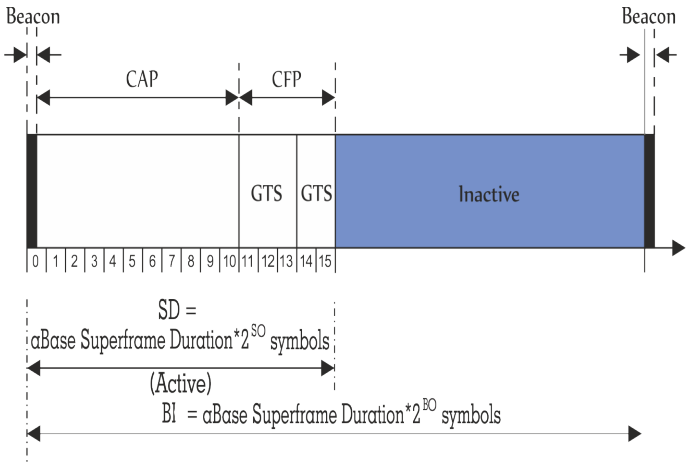


Figure 2.4: Super frame structure.

uses a structure called superframe to regulate the packet transmission and the node's state. The superframe can be divided in two periods: active and inactive. The BI defines the entire period of a superframe, and the Superframe Duration (SD) defines the active period duration. The active period (represented by the white rectangle in Figure 2.4) consists of 16 time slots with the same length which can be further divided into a Contention Access Period (CAP) and a Contention Free Period (CFP) portions. The slotted CSMA/CA protocol is used to control the node access to the channel in the CAP period. The CAP is used to send non-time-critical messages and MAC commands. In the CAP period all nodes can send their data to the coordinator, but to do that the nodes need to compete to get access to the channel.

The 802.15.4 standard uses the slotted CSMA/CA protocol to control the node access to the channel in the CAP period. On the other hand, the CFP is an optional portion used to transmit real-time messages through guaranteed time slots (GTS) allocated especially to nodes having real-time traffic demand. The GTS allocation is made by the coordinator when it receives a request from the node that wishes to transmit real-time packets. According to the standard, the CFP period can have at most seven time slots.

The BI starts with the dispatch of a beacon message, called Beacon Frame (BF), by the coordinator. The BF marks the beginning of the BI, as can be observed in Figure 2.4. It is sent in the slot 0 of the CAP period and contains information related to the superframe, such as the Beacon Interval (BI) and the Superframe Duration (SD). The BI, SD and DC values are calculated according to the Equations 2.1, 2.2 and 2.4, respectively, where BO *(macBeaconOrder)* and SO (*macSuperframeOrder*) are MAC layer parameters. The *aBaseSuperFrameDuration* is calculated using Equation 2.3, where *aBaseSlotDuration* is equal to 60 symbols and *aNumSuperframeSlot* is equal to 16, according to the standard. The standard defines the slot size in symbol number but it can be easily converted to time in seconds.

$$BI = 2^{BO} \times aBaseSuperFrameDuration, \quad 0 \leq B0 \leq 14 \tag{2.1}$$

$$SD = 2^{SO} \times aBaseSuperFrameDuration, \quad 0 \leq SO \leq BO \leq 14 \tag{2.2}$$

$$aBaseSuperFrameDuration = aBaseSlotDuration \times aNumSuperframeSlot \tag{2.3}$$

$$DC = \frac{SD}{BI} \tag{2.4}$$

Therefore, from the point of view of the lower layers, the adoption of the Beacon enabled mode allows the network to perform according to the need of energy saving in the sensors by changing the BO and SO parameters. Regarding the network layer, there are also well accepted standards such as the Ad hoc On-demand Distance Vector (AODV) routing protocol, Routing Protocol for Low-Power and Lossy network (RPL), [68] as well as Zigbee [94]. The AODV implements a data centric approach allowing the network communication through a mechanism of publishing and requesting the information taking into account the features of RPL networks, such as energy and memory restriction, high rate of packet loss, low data rate and channel instability. On the other hand, Zigbee employs a global addressing approach where each node obtains a unique address allowing it to find and be found by any other node in the network.

2.3 WSNs evolution (IoT)

Speaking from the application point of view, in the last years, the general architecture of the WSNs has evolved to adapt to new demands. So, now, it can be classified in two types [4]:

- **Semi-Automated Architecture** is an architecture similar to that traditionally used by the WSNs where all information are collected and transmitted hop by hop to a sink/gateway node responsible for sending information to the users.
- **Automated Architecture** arises from the integration of new devices called actuators in the WSNs giving essence to Wireless Sensor Actuator Networks (WSANs). In WSANs, sensors and actuators are blended with the environment creating a communicating-actuating network that allows a smart context-aware connectivity with existing networks. The actuator are more robust devices in term of memory, processing and battery, that take actions based on pre-specified application requirements after receiving an event information produced by one or multiple sensors.

The automated architecture differs from the traditional WSNs because now the devices interested in receiving the data produced in the network, the actuator, are scattered all over the network interacting with the data producers (Figure 2.5). As a consequence, this architecture promotes the emergence of a new spectrum of applications and put the WSANs as an important piece in the development of the Internet

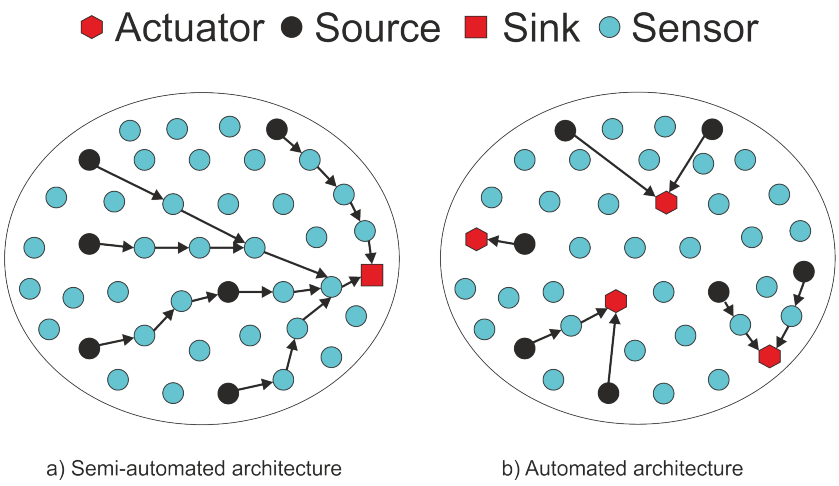


Figure 2.5: WSNs architectures.

of Things (IoT) [34] as well as some of their underlying localized applications. Indeed, the ubiquitous sensing achieved with the WSANs makes it an indispensable part of IoT.

The IoT represents the evolution of a computing paradigm that uses smartphones and portables to a computing paradigm in which all objects are connected and seamlessly embedded in the environment. In this new paradigm, an abstraction of a huge network is created allowing the iteration of the objects with the physical world (actuators), and also with different Internet standard protocols in order to provide new services that could be easily and seamlessly accessed by the users. However, by incorporating a large number of different and heterogeneous systems, the IoT promotes the emergence of applications that use the big amount of data generated to provide more and more services to the users [91]. Therefore, as the WSANs are now integrated in the IoT, the sensor nodes have to deal with the generation of a huge amount of data which have to be stored, and efficiently, easily and seamlessly delivered to the objects interested in receiving them.

2.4 Data management

In the data management performed in traditional WSNs, systems deal with data storage and retrieval in an offline mode. Data are stored in the sink node or behind it while the other sensor nodes only work as relay nodes to allow packet transmission. However, in the IoT context, the systems have to do ephemeral online data storage after receiving data from different sources and of different types, as well as, retrieve them to answer requests coming from different users. In order to deal with the new requirements present in the IoT, such as on-demand data access and a huge data generation [55], WSANs have to be able to do data management in a scenario where the collection and the processing of information become more complex, [34]. Indeed, the network need to do a shot-term storage of data, once data are now continuously retrieved to feed the demand for information coming from services and applications.

Aiming to make the WSANs capable of managing the data in this new context,

several research activities started with the aim of designing solutions for handling the underlying issues, such as energy efficient sensing, security and privacy, data mining, and data storage, [9]. As in this work we are specifically interested in the problem of doing in-network caching of IoT data, in the following we discuss the in-network data storage issues, and the mechanisms employed in this thesis to allow the WSANs performing an optimal data management.

2.4.1 In-Network data storage

According to [67], WSNs can perform three different types of data storage depending on where the data are stored: local, external and data centric. In the local storage, the sensing nodes that detect an event store the data; in the external storage, the data are sent to be stored into a base station usually located outside the network; and in the data-centric storage, the data are stored in the sensor nodes that are as close as possible to the source nodes [35]. In the data-centric storage, although being based on the local storage, all sensor nodes can store data, regardless if the data were sensed by them or not. Besides, the data-centric storage outperforms the other two, among other things, regarding energy consumption [6, 49], and the reduction in the communication overhead in the network [7].

In the context of the IoT realm, the in-network data storage performed by the WSANs employs a data-centric approach, and plays a fundamental role to allow the data consumers, which are now blended in the environment, to receive the data which they are interested in. Thus, the function of the in-network data storage consists in temporarily storing the most *fresh* data, and making easier its retrievability by the consumers [82]. This is achieved by performing decentralized caching of ephemeral data within the network. This solution, besides improving the data availability, decreases the packet loss by employing shorter multi-hop paths between source and destination [14]. To do so, however, the network has to support the data access by a large amount of users in a reliable and decentralized way. In addition, as we are talking about WSANs, the energy shortage, typical of this network, has also to be taken into account. Therefore, the major two issues in this domain are:

Energy constraint The need of handling the huge amount of data generated to suit the foreseen IoT applications leads to an increase of power consumption given that much more transmission will be performed to store and deliver the data [10]. Indeed, according to [84], sensor communication rather than computation is the most expensive operation in terms of energy consumption in WSANs.

Data reliability Performing a distributed data storage in WSANs has two main advantages: 1) it decreases the packet loss due to packet collision, and 2) it is more robust and fault tolerant since the data are stored in distributed way. However, distributed data storage also creates a challenge in making the data arriving at the consumers, requiring thus more reliable and efficient data retrieval mechanisms. Besides, although less packets may be lost in case of failure in the storage node, packets may be lost still.

Concerning the energy issue, the better way to reduce power consumption in the data storage scheme is to decrease the communication overload among sensor nodes.

To this end, the adoption of the Publish/Subscribe paradigm emerges as the most suitable communication model for WSANs. On the other hand, in order to overcome the drawbacks related to the data reliability in decentralized data storage, packet redundancy techniques have been used in the literature. The easiest and more common way to introduce redundancy in the network is through replication of data packets that consists in storing several copies of the same packet in different storage nodes. However, this approach incurs a high transmission cost; and yet does not solve the problem of data retrieval since one would have to know exactly in which node is stored the data packet to recover it. Filling this gap, coded-based decentralized data storage arises as a reliable data storage opportunity; and although having a considerable cost, it can also be applied as an efficient data retrieval solution.

Although several works have been proposed in the field of data storage, most of them are either focused on the problem of how to store the data in a way to facilitate the match between them and the queries sent by the user [11, 69, 7, 90, 47, 18, 16], or on the data reliability and retrievability [19, 65, 41, 24, 86, 85, 32]. As a consequence, there is a lack of works interested in a reliable data storage that seek to provide data availability from the source to the destination, ensuring the trade-off between the cost (energy consumption) and data reliability. In this thesis, we approach this problem and propose solutions in order to contribute with the progress in this field. Next, we describe the techniques used in our contributions to reach this progress by reducing the storage cost and increasing its reliability: the publish/subscribe paradigm and the coding theory, respectively.

2.5 Publish/Subscribe paradigm

The Publish/Subscribe paradigm (Pub/Sub) creates an abstraction that facilitates the data management in large scale networks offering a scalable, efficient and reliable dissemination of information. With the development of the Internet of Things and the growth in data generation and consumption, the demand for a more flexible communication model based on a loosely coupled iteration point out the pub/sub paradigm as a more adequate communication model to be used in a data-centric network [23]. Actually, according to [8], the data-centric storage provides a low level support to a successful implementation of the Pub/Sub paradigm which results, among other things, in the decrease of energy consumption in the data access [63].

The pub/sub system is composed of autonomous devices distributed all over the network where they can perform either as data consumers or as data producers [73]. In such a system, the communication is set by the transmission of three primitives which are [32]:

- **Advertise:** primitive sent by the producer to announce the type of data it can produce.
- **Publish:** primitive used by the producer to sent the date to all consumers that are interested in receiving it.
- **Subscribe:** primitive sent by the consumer to announce its interest in receiving a determined information.

2.5.1 Publish/Subscribe architecture categories

Regarding the architecture of a pub/sub system, they can be classified in two categories: client-server and peer-to-peer [53].

Client-Server In this model, the entities are classified under two roles, either they are event servers or event clients. The event server receives events (publication or subscription) while the event client acts as publishers, subscribers or both. The topologies accepted in this model are: star, hierarchical, ring and irregular polygon.

Peer-to-Peer In this model there is no differentiation in the role performed by the entities in the network, every node can act as a publisher or a subscriber.

2.5.2 Data dissemination

The routing protocols for sensor network are designed for synchronous communications. However, if one assume the pub/sub model the data should be routed in an asynchronous way given that publishers and subscribers do not know each other and are not aware of the production of the data. The routing in the context of data-centric network is thus classified as follow [78]:

- Centralized-based - All messages, query and event, are sent to a central broker that is the only destination in the network.
- Broadcast-based - The query is broadcasted by the subscriber to all nodes in the network. Upon receiving a query, the publisher, that has a matching event, publishes the data.
- Gossip-based - In this solution, publisher and subscriber use random walks to disseminate their queries and events as a gossip. In this method, each query and event are replicated on all nodes visited by the random walk to allow that an event finds all the matching queries.

The Pub/Sub paradigm characteristics make it suitable in the context of WSANs. However, in this context, the introduction of a novel concept, called Virtual Broking (VB), has been explored in order to decrease the cost in energy consumption associated with the dissemination of queries and events. 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. In the basic operation of this model, the data producers as well as the data consumers send their events and queries, respectively, to the VB. As an instance of such approach, [51] proposes a VB-based architecture where the network is divided into a number of small partitions similar to the one illustrated in Figure 2.6. In each partition, virtual broker nodes (depicted in black) are located at the center; and the remaining nodes, the publishers and subscribers, are spread all over the network. For the remainder of this thesis, we follow the realistic assumption of a WSAN with a VB-based architecture made up of several partitions, such as the one illustrated in Figure 2.6, that operates under the Pub/Sub model described above.

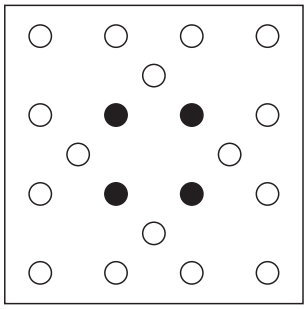


Figure 2.6: Network partition.

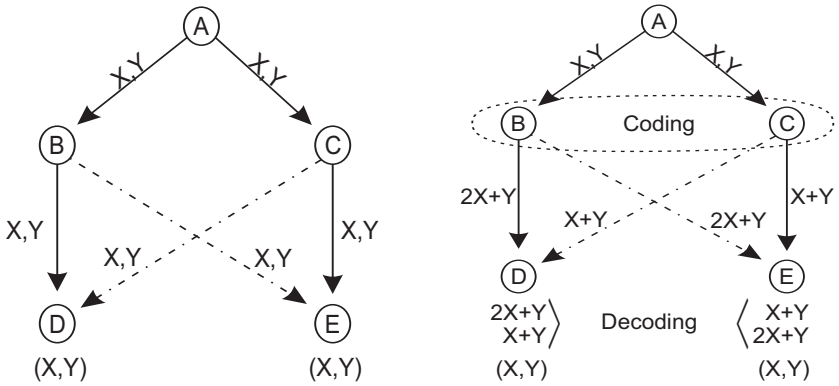
Issue: one should notice that although the Pub/Sub paradigm, when employed in a VB-based WSNs, can be efficient as an architecture to implement a distributed data storage scheme, it does not offer any guarantee concerning the data reliability.

2.6 Coding techniques

In wireless networks, as many packet are lost or corrupted due to the mobility and instability of the communication channels [33], coding-based schemes are employed as a solution to increase data reliability through the storage of redundant packets. By using coding schemes, the nodes introduce packet redundancy in the network at a lower overhead than using replication. This happens because using coding technique to store redundant packets, the receiver can recover the packets even that some packets are lost. The quantity of lost packets born in the network depending on the redundancy level employed. With the replication technique, on the contrary, for each packet lost, a packet retransmission is requested overloading the network. By exploiting the advantages of the coding techniques, Network Coding (NC) and Erasure Coding (EC) have been proposed to improve reliability in distributed data storage schemes in WSNs [19, 65, 28, 93, 43, 89, 29, 30, 81, 37].

Erasure coding (EC) is a method of data replication created more than 50 years ago. It divides a data in i blocks that are used to generate n coded packets that are transmitted to a set of storage nodes. The goal of an erasure coding scheme is to allow the recovery of any data packet from any subset of i coded packets. Therefore, even if $n - i$ coded packet are lost or corrupted, the original data can still be recovered. There are different types of erasure coded such as the most common Reed-Solomon codes, the Low-Density Parity-Check (LDPC) and the Fountain Codes.

Network coding (NC) was originally introduced to improve the throughput obtained by multicasting in a point-to-point network [3], but it has since found applications in several other tasks such as network tomography, robustness enhancement and security [60], as well as data storage [93, 89, 81]. NC replaces the traditional store-and-forward procedure, performed in the intermediary nodes of the network, by a store-code-and-forward procedure. That is, the intermediate



(a) Without network coding. (b) With network coding.

Figure 2.7: Example of Network Coding Operation.

nodes forward codifications of the packets that arrive to them, and the end-receiver collects packets that arrive through different paths and decode them to recover the information that was transmitted at the other end.

Both NC and EC improve packet reliability in data storage schemes. The difference between them lies on the fact that NC allows the encoding of packets coming from multiple sources in intermediate nodes whereas in EC only the source nodes can encode packet. This difference allows the NC to take advantage of the overhearing characteristic of the shared wireless medium in WSNs to increase the throughout and decrease the delay. In order to appreciate the advantages advocated by the NC, in Figure 2.7, we show an example of its operation in a broadcast scenario. In this figure, the source node A broadcasts two packets, x and y, which arrive to intermediate nodes B and C. In Figure 2.7a, we have the store-and-forward operation mode where, upon receiving x and y, the nodes B and C broadcast each packet at a time making a total of 6 transmissions in order to deliver x and y from A to D and E. In Figure 2.7b, we have the NC operation. In this case, nodes B and C combine the received packets x and y into a single packet before forwarding them. At the destination nodes D and E, the packets x and y are recovered by solving a linear equation system using the combinations of the received packets. The combination operation performed by nodes B and C makes each of them saves one transmission. Thus, in this example, the NC delivers x and y to D and E at a cost of only 4 transmissions. This example shows the case of one-hop NC. Others NC schemes exists. Among these, the Linear Network Coding (LNC) for multi-hop NC is described hereafter.

2.6.1 Linear Network Coding (LNC)

In Linear Network Coding (LNC), packets are regarded as vectors $\mathbf{x} = (x_1, \ldots, x_l)$ of elements on the finite field $GF(q)$ of size 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). Then, 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 $\mathbf{x_1}, \ldots, \mathbf{x_N}$. The decoding process is only possible if N linearly-independent coded packets are received. Hence, we 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 $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

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

Now, assuming that the coefficient matrix $\mathbf{C}$ has rank N, we thus obtain the following system of linear equations

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

where $\mathbf{Y} = [\mathbf{y_1}, \ldots, \mathbf{y_N}]$ and $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_N}]$, can be solved uniquely in order to recover the packets $\mathbf{x_1}, \ldots, \mathbf{x_N}$. In practice, if the computational cost is an issue, one should use a sparse coefficient matrix $\mathbf{C}$ where most entries are zero, so that the coding and decoding processes can be implemented in an efficient way. In this case, one must only be careful to guarantee that the matrix $\mathbf{C}$ has the required rank N. Linear Network Coding is based on this set of precepts.

Issue: the coding techniques achieve reliability allowing the data to be retrieved with high probability. However, as the coding-based schemes proposed in the literature assume an unstructured network, the use of a coding technique results in a high communication overhead.

2.7 Conclusion

WSANs may be employed in a wide range of applications which range from simple environment monitoring to advanced services proposed through the IoT, such as all home automation and smart cities applications. In order to efficiently support these services, WSANs have to face the problem of managing the data produced in the network. However, as discussed in this chapter, the solutions proposed in the literature for data management in WSANs, do not take into account the new requirements that arise from the vision where WSANs are an integral part of the IoT realm allowing the access to a wide-range of novel applications and services. Among the new challenges that the WSANs have to face, in this thesis we are interested in those related to in-network data storage, data reliability and retrievability, communication overhead and energy consumption. How to handle a big volume of data within WSANs is one of the important challenges of the Future Internet of Things. Fault-tolerance is another problem of the IoT: ubiquitous networking requires redundancy in several levels and ability to automatically adapt to changed conditions in order to guarantee robust communications [56]. Aiming at solving some of these related problem, we use the Pub/Sub paradigm

and the Network Coding technique in our contributions. The Pub/Sub paradigm, assuming a VB structure, improves the data storage process reducing its communication cost, but does not ensure reliability in the situation of packet loss and failures in the VBNs. On the other hand, storing redundant packets using the Network Coding technique makes the data storage process reliable, but, as most of works that use coding technique consider an unstructured network, incurs in high communication cost. So, in next chapter, we envisage to take advantage of both approaches - i.e. data storage reliability from coding technique and the low communication cost from a VB-based publish/subscribe architecture - in order to develop a reliable and robust data storage system.

Chapter 3

VBC: Virtual Broking Coding data storage scheme

Contents

4.1 Motivation 54
4.2 State of the art 55
4.3 System model 56
4.4 Theoretical analysis: a generalization 60
4.5 Optimization problem formulation 63
4.6 Performance evaluation 64
4.7 Conclusion 69

In this chapter, we are interested in doing reliable in-network data storage and data retrieval in Pub/Sub-based Wireless Sensor Actuator Networks (WSANs). In the process of data storage many packets can be lost before arriving at the storage node or in their way from the storage node to the data consumer. In order to ensure data reliability and an efficient mechanism for data retrievability, we propose the Virtual Broking Coding data storage scheme (VBC). VBC uses a Virtual Broker (VB) structure responsible for storing redundant coded packet (which are generated through the Linear Network Coding (LNC) technique) and intermediating the communication between producer and consumer nodes. The performance of VBC is validated by a theoretical analysis and simulated experiments demonstrating that VBC achieves a good trade-off between the cost and the benefit of storing redundant coded packets.

This chapter is organized as follows. Section 3.1 introduces our motivation and the context of this work followed by the related works in Section 3.2. In Section 3.3 we present our solution, the Virtual Broking Coding (VBC) data storage scheme. Section 3.4 shows a theoretical analysis of the performance of VBC while Section 3.5 theoretically compares the communication cost of VBC with that of our benchmarks. Section 3.6 discusses our simulation results and, finally, Section 3.7 concludes this chapter.

3.1 Motivation and context

In the context of the Internet of Things (IoT), any informational device is a potential element of a global data communication network. As a consequence, we are led to change the way we understand and design WSANs. Important examples of the employment of WSANs 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 WSAN itself. Sensors, actuators and controller devices work as smart devices which take decisions and/or trigger actions. The problem of reliable in-network data storage then arises from the necessity of making the collected data available at any time to these controllers, sensors and actuators used by localized IoT applications. Despite all the advances underwent by the sensors, controllers and actuators, the WSANs remain resource lacking networks with respect to energy, memory and computation power. Therefore, the challenge is to design a robust and reliable data storage scheme for such constrained and automated WSANs.

Two different approaches to tackle the data storage problem exist in the literature: those based on the Publish/Subscribe paradigm [59, 80, 77, 74, 75, 88, 71], 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 [27].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 lead to packet loss, 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 [19, 93, 43, 29, 37]. These solutions do improve the performance of a data storage system, but they still present a considerable communication overhead for implementation on WSANs.

Aiming to overcome the shortcoming of both approaches, we propose an in-network data storage scheme 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. With this solution, besides improving the overall communication cost, VBC also offers a more flexible access to the stored data, where the information consumers can have direct access to any portion of data produced by the sensors, at a proportional cost. This last feature makes VBC a particularly well suited solution for the data storage problem on WSANs in the context of the Internet of Things realm.

3.2 Related works

The Pub/Sub paradigm has been pointed out by [27] as an encouraging solution to face the problem of content delivery in the Internet as envisaged in our days. However, despite all researches on the Pub/Sub paradigm on WSNs [36, 22, 2], the reliability of

the data storage has not received much attention so far. In order to address the reliability problem, several works propose the implementation of efficient data redundancy schemes using coding technique, such as erasure codes [19, 43, 37]. These solutions, however, adopt a strategy of distribution of coded packets based on random walks, which leads to a high communication cost.

Next, we present some works found in the literature and discuss their shortcomings regarding the reliability and the communication cost of their solution.

3.2.1 Pub/Sub-based storage schemes

The Pub/Sub paradigm has been employed in several solutions looking for solving the in-network data storage related issues. However, most of them, such as in [72, 59, 79, 75, 17], propose new architectures and protocols in order to set an efficient communication model between publishers and subscribers, and neglects the data reliability.

Other problems are also approached, in [36] the author care about the access to the data by the mobile subscribers while in [80] the authors are interested in finding the optimal way to disseminate the data stored through a multicast scheme considering a real-time application. Other works are more focused in solving high level problem concerning the data access by the subscriber, such as the heterogeneity of consumers' requirements in [27], and the way of setting the communication between publishers and subscriber in [40]. In [2], another high level issue is approached. In this case, the authors address the problem of user's data security, such as confidentiality, authentication and authorized access. Nonetheless, also in these works, the data reliability is not faced up.

The work we found that is closer to this topic is undertaken by Liu et al. [52] 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 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.

3.2.2 Reliable coding-based storage schemes

In order to address the reliability problem some kind of redundancy technique should be implemented within the WSANs. The simplest redundancy solution consists of replicating the same information in several nodes of the network. This method, however, requires too much memory and is not suitable for WSANs. Several works propose the use of coding techniques in order to implement data redundancy in WSNs in an efficient way, such as in [19, 43, 37]. In these works, they propose different solutions

based on Erasure Codes on unstructured networks. The Erasure Codes based algorithms divide a data object into k packets that are coded into n new packets, where n > k. In the best algorithms, the k original packets can be decoded by a collector accessing any k coded packets when almost all storage nodes store coded packets. The problems in these solutions are, first, the cost of the algorithm used to forward the n coded packets to the storage nodes that, although smaller than in other redundancy techniques, is still not negligible for allowing an efficient use in WSANs. And second, the fact that practically all storage nodes in the network (which are all nodes but the sources and the consumers) must to store the coded packets to allow the consumer to decode the packets after requesting packets at any subset of k storage nodes.

In the attempt to reduce this cost, Dimakis et al. [29] propose a mechanism of distributed erasure coding (DEC), inspired from the network coding technique presented in [3]. In the 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 precomputed 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. In order to ensure the packets decoding with high probability, the main result in [29] states that, the number of storage nodes associated with each source node must be about $5\frac{n}{k}ln(k)$. Although DEC represents a step forward with respect to previous coding-based solutions through reducing the number of storage nodes to be requested, it still presents a considerable communication overhead for implementation on WSANs since they assume a large scale network. Other 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.

3.2.3 Summary

It is worth noting that in all previous works that use coding technique is assumed an unstructured network. As a consequence, they have higher communication and encoding cost. On the other hand, the data storage schemes based on a Publish/Subscribe architecture are unreliable although more efficient in terms of communication cost. So, next we present our contribution whereby we envisage to take advantages of the benefits that come from both approaches, the data storage reliability from coding technique, and the low communication cost from a publish/subscribe architecture, in order to develop a reliable and robust data storage system. As we are interested in ensuring the recoverability of the data stored in the VBNs at the network level, the pub/sub matching problem wont be addressed in this work, some solution in this field can be found in [58], [75] and [88].

3.3 VBC: Virtual Broking Coding data storage scheme

Our proposal, VBC, is a scheme based on Linear Network Coding (LNC) through which VBC stores coded packets in a redundant way. In order to introduce redundancy in our scheme, VBC needs to generate M coded packets from N data packets, where $M > N$. Just to recall, in LNC the 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. So, formally assuming an $N \times M$ coefficient matrix $\mathbf{C}$ we are allowed to compute a collection of M coded packets as linear combinations of the N original packets $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

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

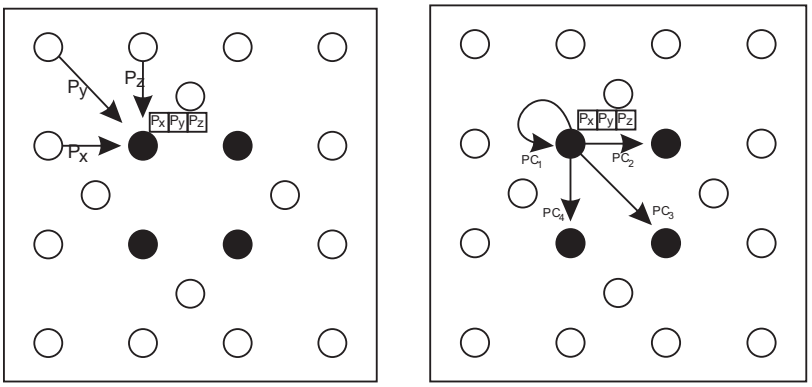
In this context, the desirable property of the redundant coding scheme is that any subset of N coded packets can be used to recover the original packets. In mathematical terms, this requires that every $N \times N$ sub matrix of the coefficient matrix $\mathbf{C}$ must have rank N. Assuming this condition 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.

3.3.1 Publish operation

In the publish operation, each VBN 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 3.1 illustrates the steps involved in the publish operation using redundancy level 4/3 where a VBN receives packets P_z, P_y and P_x that are coded together, and generates four coded packets (PC_1, PC_2, PC_3 and PC_4) distributed among the VBNs.

3.3.2 Subscribe operation

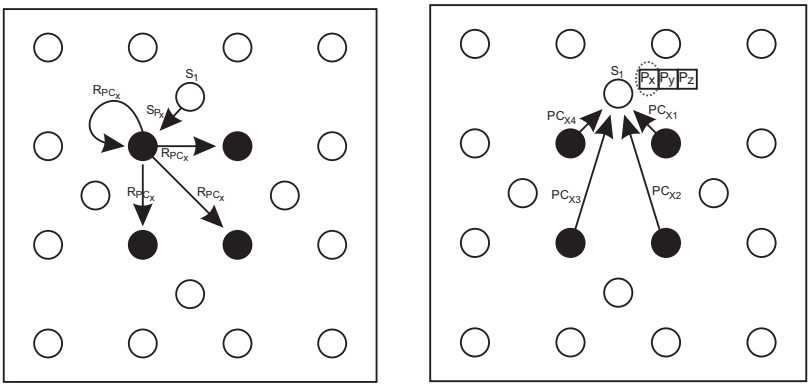
In the subscribe operation, when a VBN receives a request for a given packet p, it broadcasts the request to the other VBNs. Then, any VBN that holds a coded packet that was produced from p in its memory, sends this coded packet directly to the subscriber. The subscriber is responsible for performing the decodification of the coded packets, as described in Chapter 2 (Section 2.6.1), which is always possible if at least N coded packets are received. Figure 3.2 illustrates the steps involved in the subscribe operation using a redundancy level of 4/3. In Figure 3.2, the subscriber S_1 requests packet P_x at the closest VBN. The VBN, in its turn, broadcasts the request for all



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

Figure 3.1: Publish operation

VBNs. Then, all VBNs that hold a coded packet with P_x among the original packets used in its generation send the coded packet to the subscriber.



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

Figure 3.2: Subscribe operation

3.3.3 Discussion

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 [52]. 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, these design choices has a considerable impact on the ratio of requests completed by the proposed solution compared to other existing schemes.

3.4 Theoretical packet delivery ratio analysis

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) situation - where besides the packet loss probability σ, one VBN fails.

3.4.1 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 [52]. So, we focus on a single partition (see Figure 2.6), 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).

3.4.2 Analysis

We start calculating the probability of losing packets in the publish operation. Here, when the packet p arrives at VBN_1, it will be coded together with another packet p' to produce the coded packets c_1, c_2 and c_3. One of these packets will be stored in VBN_1 itself, and the other two will be transferred to two randomly chosen VBNs. 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 definitively 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. Here, we assume that the request arrives at a VBN that holds a coded packet produced from p, and the other two coded packets are in neighboring VBNs. Again, there are three cases and standard calculations that give us the following:

- 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 arriving at the VB for a packet which has also reached the VB is delivered to the subscriber is, in the EPCC situation,

$$\sum_i a_i b_i = \frac{-2\sigma^8 - 20\sigma^7 + 83\sigma^6 - 184\sigma^5 + 233\sigma^4 - 164\sigma^3 + 53\sigma^2 - 3}{3} \tag{3.1}$$

and in the FPVB situation,

$$\sum_i a'_i b'_i = \frac{2\sigma^7 - 14\sigma^6 + 37\sigma^5 - 41\sigma^4 + 4\sigma^3 + 32\sigma^2 - 27\sigma + 7}{9} \tag{3.2}$$

3.4.3 Comparison of results: VBC vs BS

Figure 3.3 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 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. In this situation, an argument similar to that used for VBC analysis gives us that 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 about:

$$\frac{9\sigma^4 - 39\sigma^3 + 66\sigma^2 - 52\sigma + 16}{16} \tag{3.3}$$

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 3.3 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.

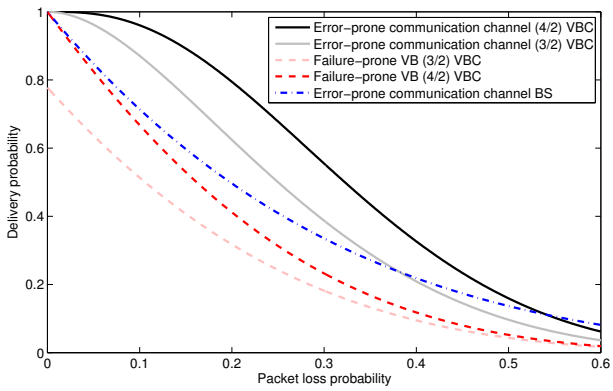


Figure 3.3: Packet delivery probability.

3.5 Theoretical communication cost analysis

3.5.1 Comparison of Coding Schemes: VBC vs DEC

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 WSAN 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 [29] are supposed to hold when the number of nodes is very large (see Theorem 3, pg 7 in [29]). VBC, on the other hand, was designed to operate on networks organized into small partitions, as

described in [52], and it takes advantage of this structure to: (1) offer a more flexible data storage service, and (2) a lower communication cost.

The important issue here is that, although DEC performs a decentralized coding procedure, the code which 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, while DEC implements an all-or-nothing model of access to the data, VBC allows the user to have access to any portion of the stored data incurring just a proportional cost.

3.5.1.1 Assumptions

In order to compare the communication costs of VBC and DEC, we will consider the simplified scenario discussed in [29]: a grid sensor network with N nodes, and $k = O(\sqrt{N})$ data sources producing one piece of information each. We will analyze the costs of the operations of publish and subscribe separately.

3.5.1.2 Communication costs 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 and may depend on the specific characteristics of the application. During the publish operation, each data source will send its data 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.

Next, we consider the communication cost of VBC in the same scenario. In the publish operation, each of the $O(\sqrt{N})$ data sources will send one data 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 will produce local codes, each corresponding to a small generation of packets, and will 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 will have 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 $O(NlnN)$ and $O(N)$, respectively. What we would like to emphasize, however, is the flexibility offered by VBC, which allows data collectors located in any place of the network to have direct access to their desired data, incurring a cost which is proportional to the amount of information retrieved. This feature makes VBC more suitable for the localized IoT applications, which are predominant in the IoT realm and where collectors are also spread out around the network and typically interested in local information.

3.5.2 Comparison of VB-based data storage schemes: VBC vs BS

Here, we aim to analyze the cost of both schemes, VBC and BS, through a metric we named the cost-benefit metric. The cost-benefit is defined as the ratio of the communication overhead to the number of packets delivered to the subscribers. As we already have estimated the packet delivery ratio of VBC and BS (Section 3.4.2 and 3.4.3, respectively), now we calculate only their cost. In this analysis, we do not specifically asses the energy waste because we assume that the energy consumption is directly related to the communication cost. Indeed, according to [83], 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 scenarios where the subscribers are more powerful devices, such as cellphones and actuator sensors. This way, the computational overhead does not have much impact in the VBNs that perform relatively simple operations with low costs. Thus, obviously, it is the transmission cost that makes the difference in the energy consumption.

3.5.2.1 Assumptions

As we made in Section 3.4, we perform the analysis considering two different situations: the EPCC and the FPVB. We also continue to assume the scenario showed in Figure 2.6 with one publisher, one subscriber, and time of observation τ. In this study, as we also analyze the FPVB situation, τ is divided in τ' and τ'', where τ' is the time interval without failure in the VB and τ'' is the time interval that the VB remain with a broker node out of work. 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 to BS, we further assume the cost of the load balancing procedure as a cost of the publish operation. This includes the cost of distributing the packets arrived at the VBN_1 among all VBNs. Moreover, the control messages broadcasted at each 3 seconds by each VBN to its neighbors. In the subscribe operation, we assume the cost of answering the τr_2 requests generated by the collector t and sent to VBN_3. Besides, here for the same reasons discussed in Section 3.4.3 we analyze BS only in the EPCC situation.

3.5.2.2 Error-prone communication channel (EPCC)

In this scenario, 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 is $\tau r_1 \cdot (1-\sigma)$. Starting with the communication cost of VBC in the publish operation, we have that 2 coded packets are distributed with costs x. Thus, the cost is defined as $\tau r_1(1-\sigma)x$.

In the subscribe operation the cost is given by the addition of two different costs. The first one is the cost associated to the broadcast sent by the VBN that received the request from the subscriber, which costs z. The other one is the cost associated to the answers 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 packet loss 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 communication cost of VBC is:

$$\tau r_1(1-\sigma)x + \tau r_2(1-\sigma)\Big(z + 3y(1-\sigma)^2\Big) \tag{3.4}$$

Next, we realize the BS analysis with the communication cost. In the publish operation, as only 3/4 of the packets that arrive in the VB will be distributed, we have a cost of $3\tau r1(1-\sigma)x/4$. 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 $\tau r2(1-\sigma)/4$ requested packets are delivered with a cost y as they are directly transmitted from the VBN that received the request. To the other $3\tau r2(1-\sigma)/4$ requested packets, we have to add the cost z related to the broadcast of the request to the VB. We also add the cost y of answering the request when a VBN has the requested packet, which means $3(\tau r2(1-\sigma)^3)$ requests. Thus the total communication cost of BS is given by:

$$\frac{3\tau r1(1-\sigma)x + \tau r2(1-\sigma)y + 3\tau r2z + 3\tau r2(1-\sigma)^3 y}{4} + \frac{4\tau}{3} \tag{3.5}$$

Finally, by computing the ratio between the cost and the delivery probability computed in Section 3.4, we obtain the cost-benefit of both schemes in the EPCC situation. Table I presents the values of the parameters used to produce the curves in Figure 3.4. 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. In fact, 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 3.1: Parameters of the cost-benefit estimation.

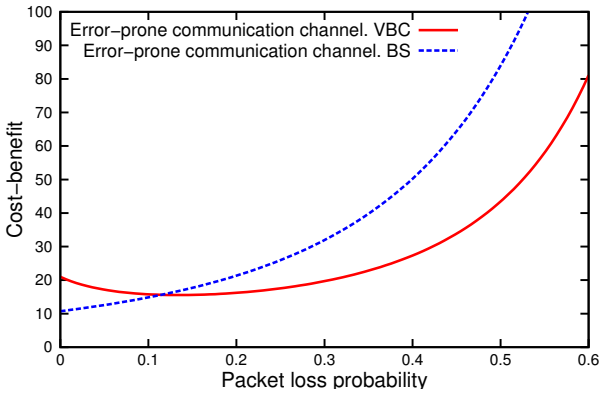


Figure 3.4: Cost per packet delivered.

3.5.2.3 Failure-prone VB (FPVB)

Although we cannot present a comparison between VBC and BS in this scenario (BS cannot be analyzed under the FPVB situation, as discussed in Section 3.4.3), we finalize this analysis computing only the cost of VBC. Here, we introduce a failure in one VBN. As during τ'' less messages are transmitted, we compute the cost for τ' and τ'' separately. In this case, we have that during τ the situation is the same as that presented in the Section 3.5.2.2 for the cost of publish and subscribe operations. So, during τ, these costs are $\tau r_1(1-\sigma)x$ and $\tau r_2(1-\sigma)\Big(z + 3y(1-\sigma)^2\Big)$, respectively. Besides, we have that the failure of a VBN does not affect the communication cost in

the publish operation, so during τ'' the cost of publish operation also remains the same presented during τ. This way, during τ'', we only have to compute the communication cost for the subscribe operation.

Following the same argument used in the last scenario, we first compute the cost of the broadcast sent by a VBN after receiving a request, that is $\tau'' r_2(1-\sigma)z$. Next, the probability of a VBN answers the request assuming that it had received the coded packet with a $(1-\sigma)$ probability. Then, the probability of a VBN to receive the request, that is $(1-\sigma)$; and, finally, the probability of being active, that is 3/4, as we assume that only one VBN fails. So, we have that the total communication cost of VBC in the FPVB is:

$$\tau r_1(1-\sigma)x + \tau' r_2(1-\sigma)\Big(z + 3y(1-\sigma)^2\Big) + \tau'' r_2(1-\sigma)z + \frac{\tau'' r_2 9y(1-\sigma)^2}{4} \quad (3.6)$$

3.6 Performance evaluation

3.6.1 Simulation settings

In the following, we replicate the situation proposed to the Balanced Storage (BS) scheme in [52] in order to assess the performance of VBC and fairly compare it to the one of BS. As Liu et al. [52], we assume that all nodes have a transmission range of 16m and are placed in a partition that has a dimension of 50m x 50m. Besides, the partition has 20 nodes whose 4 make up the virtual broker responsible for storing the data (see Figure 2.6). 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.

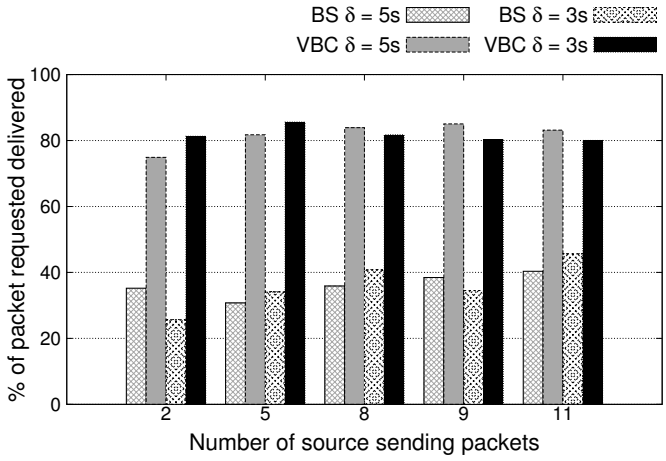
3.6.2 Simulation results

We consider in this evaluation study, the EPCC situation that is the only situation considered in [52]; 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 takes into account five metrics, the packet delivery ratio, the VB communication cost, the cost-benefit, the storage load balance and the delay in delivering the requested packet. In all graphics presented in the evaluation, the dashed lines represent BS and the solid lines VBC results, as well as the forms square and circle represent the values set to the packet generation ratio δ, 3 and 5 seconds, respectively.

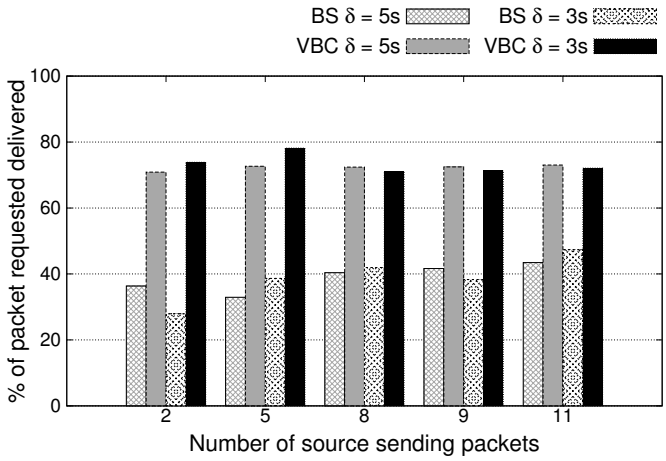
3.6.2.1 Packet delivery ratio

This metric assesses the performance of both schemes in the task of delivering the packets requested by the subscribers. More specifically, in Figure 3.5a 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 3.5b, 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.



(a) Packet delivery ratio (EPCC)



(b) Packet delivery ratio (FPVB)

Figure 3.5: Packet delivery ratio.

3.6.2.2 VB communication cost

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

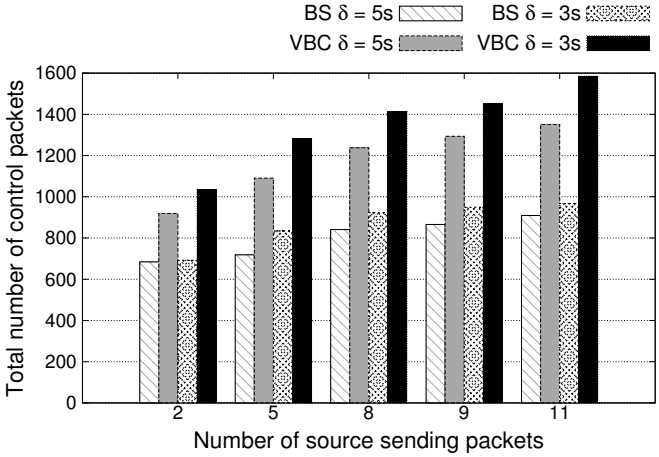
and data packets, but also includes the control messages and data transfers associated with the load balancing procedure. In VBC, the main impact on the communication cost is due to the introduction of redundancy and the fact that packets are coded together and distributed to other VBNs. In particular, 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 3.6 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, in Figure 3.6a, 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. In addition, in Figure 3.6b, it is observed that the cost is decreasing for both schemes when there are faults in the VBNs, this happens because when it has less active VBNs, less packets will be retransmitted. Thus, we have that in order to ensure the scheme's reliability, VBC pays with the increase of the overhead in the network communication. However, as we see next, it turns out that this cost is a small penalty in view of the reward in VBC performance.

3.6.2.3 Cost-benefit trade-off

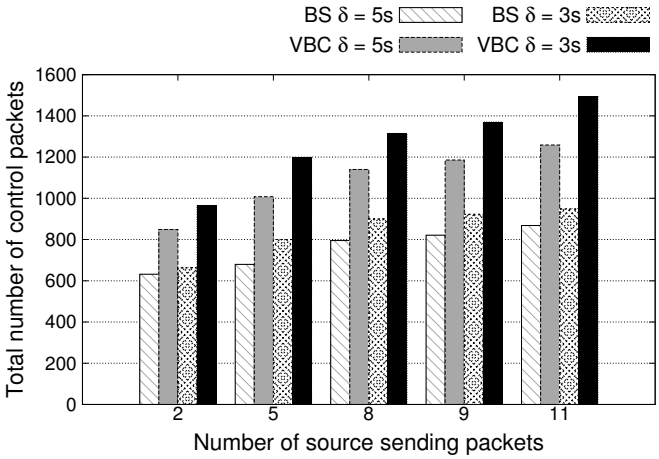
The idea here is to integrate the results of the two previous measures in order to compute the cost-benefit metric. Through this metric we estimate the communication cost per delivered packet associated with each protocol. The results displayed in Figure 3.7a show a clear advantage for VBC in the situations 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, let 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 3.7b shows that even in the case of failure in a VBN (FPVB situation), VBC also has a better performance than BS in all studied cases.

3.6.2.4 Storage load balancing

As mentioned in Section 3.3, the codification scheme and coded packets distribution implemented in VBC naturally leads to an even distribution of the data storage load among the VBNs in the same partition. In order to confirm this fact, we have reproduced the experiment described in [52], to compare the performances of VBC and BS and see which protocol is more effective from the load balancing point of view. The study presented in [52] focus on the absolute difference (in number of packets) between the maximum and minimum storage levels among the VBNs. However, in our case, we have defined our metric in terms of the relative difference: $(maxload - minload)/minload$. Due to the introduction of redundancy, VBC has to



(a) Communication Cost (EPCC)

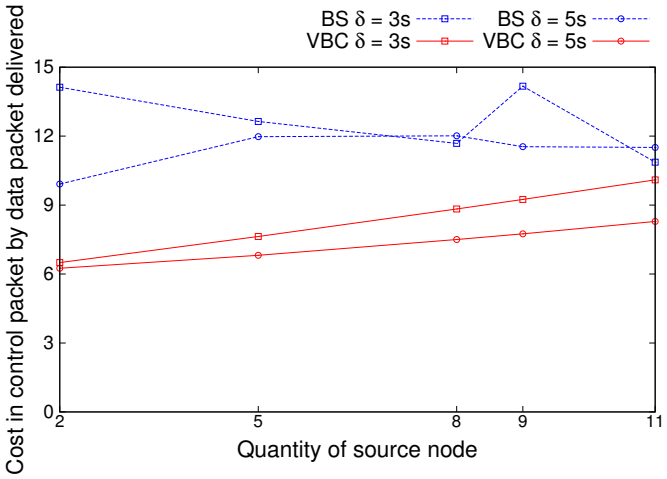


(b) Communication Cost (FPVB)

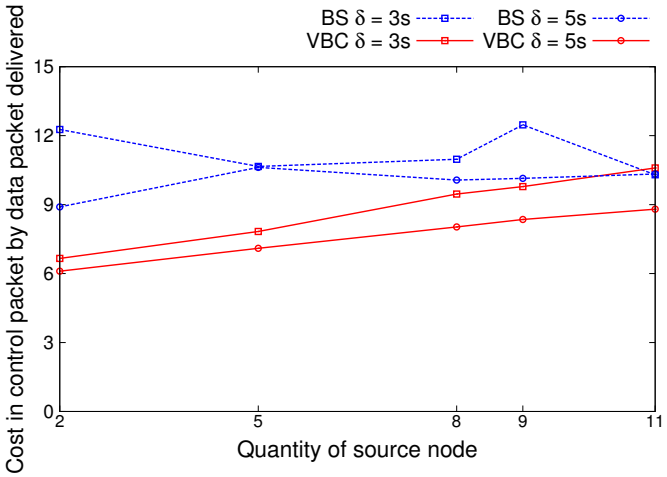
Figure 3.6: Total Communication Cost.

handle a larger amount of packets than BS and, for this reason, the relative difference is a more appropriate metric to compare the performance of the two protocols.

Figures 3.8 and 3.9 show the variation of this metric as the simulation time progresses in the cases with 2 and 11 data sources, respectively, in both simulated scenarios: EPCC and FPVB. In the case with 2 sources and a lower packet arrival rate ($\delta = 5s$) the two protocols have a similar performance: in the beginning of the simulation, the value of the chosen metric is higher due to the small number of packets in the system; but, as the simulation time advances, the relative difference between the maximum and minimum loads consistently decreases, reaching a value around 10% at the end of the simulation. With a higher packet arrival rate ($\delta = 3s$) however, the performance of VBC is clearly superior: already in the beginning of the simulation the metric has a small value, and it remains stable at a such small value during the entire simulation time. The explanation for the worse performance of BS in this case is that packets arrive at a higher rate than the distribution mechanism of the protocol



(a) Cost-benefit (EPCC)



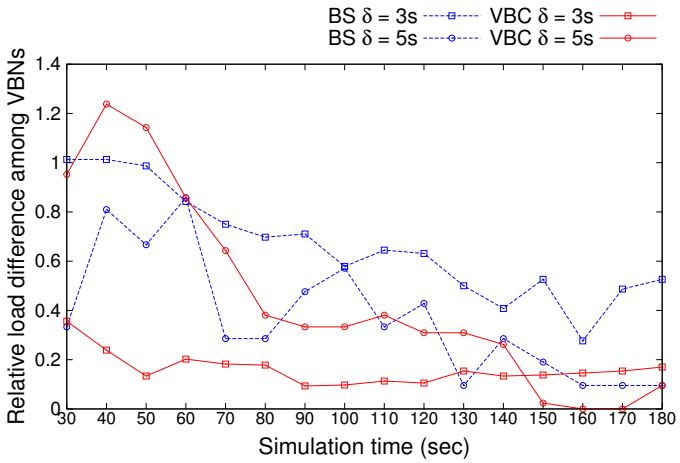
(b) Cost-benefit (FPVB)

Figure 3.7: Cost-benefit

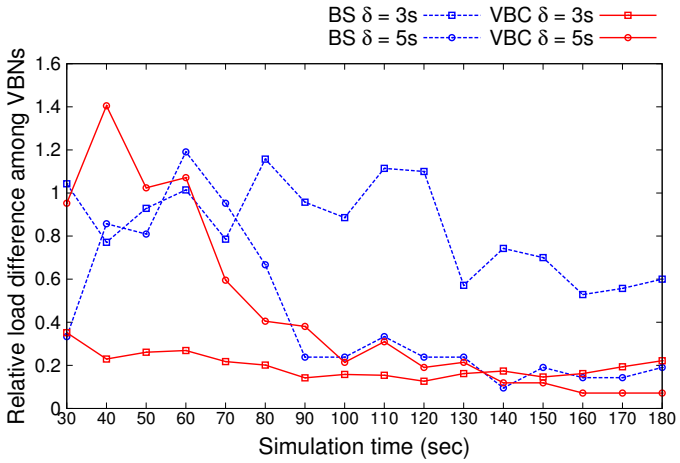
can cope with, resulting in an imbalance among the storage loads of the VBNs. In Figure 3.9 we see that with 11 data sources the situation becomes even worse for BS. The increased number of packets, on the other hand, only improves the performance of VBC, since the variation between the maximum and minimum loads gets reduced due to statistical effects.

3.6.2.5 Delay

This metric is defined as the interval of time between the time a subscription arrives at a VBNs and the time the subscriber receives the requested packets. Figure 3.10 shows the delay of the packet arrival in scenarios with 2, 5, 8, 9 and 11 data sources and δ assumed equal to 3 and 5s, respectively. From this Figure, we notice that VBC has a greater delay than BS. This happens due to the fact that the subscriber has to wait for the arrival of all coded packets needed to perform the decoding process in the VBC scheme. Unlike, BS only sends one packet if it is stored in the VB. Similarly to



(a) 2 data sources (EPCC)

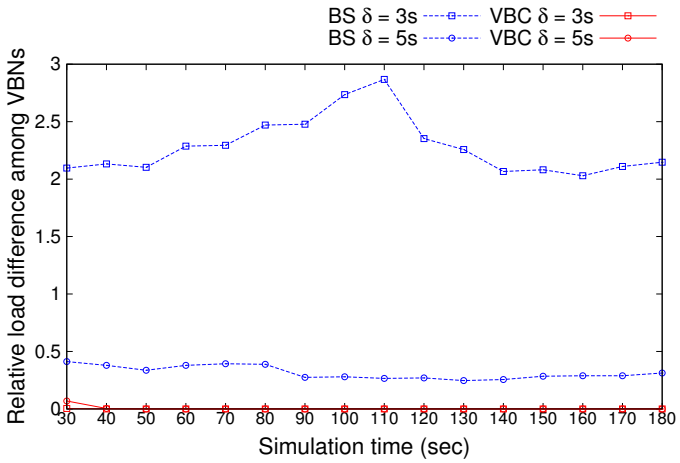


(b) 2 data sources (FPVB)

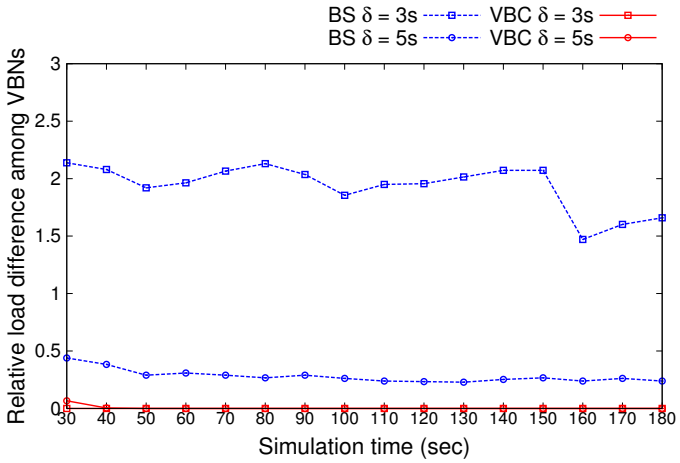
Figure 3.8: Relative load balance between VBNs 2 data sources

the communication cost, the increasing in the delay is a price to pay to keep the data reachable in the network.

However, we have that VBC delivers N packets instead of one delivered by BS. So taking this into account, if one considers the delay per packet delivered, VBC actually take 0.6s to deliver one packet in the worst case (11 source nodes in Figure 3.10a) which means that VBC keeps the same delay presented by BS in the EPCC scenario. On the other hand, when we observe Figure 3.10b, we note that following the same train of thought, in the scenario with 2 source nodes, BS take 120% more time to deliver one packet. Another aspect that also impacts this results is the fact that in this experiment we are not taking into account specific application needs so that any treatment has been done in the network regardless of any application requirement. As, for instance, giving priority to sending first the packets containing the requested coded packets. Besides, the flexibility of the coding scheme proposed by VBC can be used according to the application requirements to compensates the delay. For instance, if the delay has



(a) 11 data sources (EPCC)



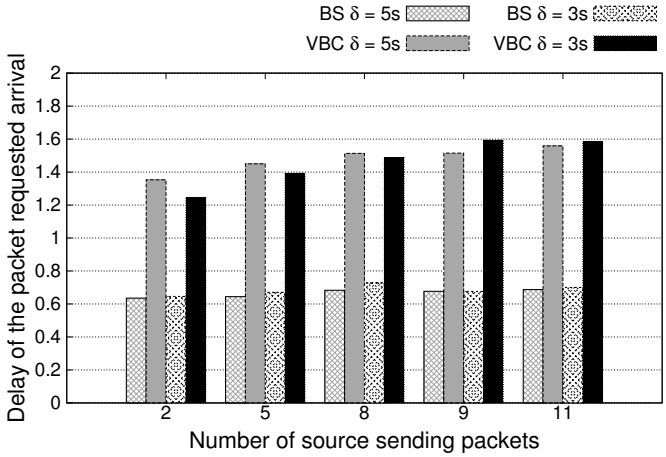
(b) 11 data sources (FPVB)

Figure 3.9: Relative load balance between VBNs 11 data sources

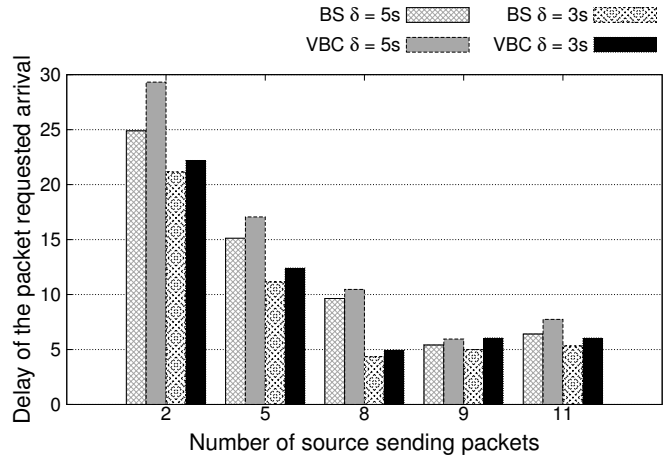
to be prioritized, simply reduce N, thus the VBN and the receiver do not have to wait too long to distribute the coded packets and recover the requested packet, respectively. Therefore, even though providing a reliable and robust data storage scheme incurs in some costs, VBC can be adapted to the application in order to compensate its cost taking advantage of its operation mode.

3.7 Conclusion

In this chapter, 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



(a) Delay (EPCC)



(b) Delay (FPVB)

Figure 3.10: Cost-benefit

using a virtual broker structure, VBC accomplishes to overcome this cost. Besides, the cost-benefit metric defined in our evaluation 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. Indeed, we saw through an analytical discussion that introducing a coding scheme in a VBN based network with a Publish/Subscribe communication model is less expensive than, instead of, assuming an unstructured network with a completely decentralized communication model.

However, 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 (σ), VBC can do even better. The optimization of the redundancy level used by VBC according to this, and the VBC evolution and integration in the IoT realm are the subject approached in the next chapter.

Chapter 4

DA-VBC: Dynamic Adaptive VBC Data Storage Scheme

Contents

5.1 Context and issues 72
5.2 Related works 73
5.3 Duty cycle adaptation in traditional WSN 74
5.4 Duty cycle in DA-VBC-based WSANs 84
5.5 Conclusion 90

Previously we addressed the problem of ensuring data reliability while doing in-network data storage in WSANs. To do so, we proposed an in-network data storage scheme called Virtual Broking Coding (VBC) that accomplishes data reliability by storing and retrieving the data assuming a new data coding model in a virtual broker structure. The success of VBC is due to the redundancy introduced in the VB using a network coding technique. However, VBC assumes a redundancy level set initially to perform during all the network lifetime without taking into account the changes in network conditions.

In this chapter, where we present our main contribution, we propose the evolution of VBC to a dynamic and self-adaptive data storage scheme (DA-VBC). DA-VBC is able to perform with the optimal redundancy level regarding to the current network behavior, as well as to be well integrated in the IoT realm. In order to make it possible, we first model the communication iteration between the entities of the IoT architecture and VBC. Then we model the problem of choosing the optimal redundancy level as a Markov Decision Process (MDP) problem, and propose the DA-VBC algorithm to solve it. 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 overloading the energy consumption.

The remainder of this chapter is organized as follows. In Section I we introduce the context of the addressed problem and in Section II we present the state of the art of the topic. 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.

4.1 Motivation

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 processes 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 Actuator Networks (WSANs) domain, [54, 62]. However, these new services and applications bring new issues to the WSANs in view of their integration within the IoT realm, such as the data access management.

Among the research trends raised by the data management in the IoT, this chapter addresses the in-network data storage and data retrieval when performed in WSANs 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 [6, 25, 19, 93, 43, 29, 37], do not take into account the new challenges and needs imposed by the integration of the WSAN in IoT. In the previous chapter, we proposed, as a first attempt, an in-network data storage scheme called VBC that assumes an IoT-based localized application scenario in order to overcome this shortage.

In Chapter 3, 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 packets. 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 network conditions in order to keep both, its reliability as well as a transparent and seamless access to the resources.

In this chapter, aiming to make VBC an adequate solution to the IoT realm, we propose its evolution through the design of a system that defines a communication model that integrates the WSANs 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 packet redundancy is to minimize the cost-benefit of our data storage scheme. 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. Hence, our contributions in this chapter are:

- A communication model that integrates the VBC-based WSANs in the IoT realm, and allows a local (localized application) and global (non-localized application) data access by building the abstraction of the information claimed in IoT.
- 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 trade-off.
- A formulation and solution to the optimization problem that consists in choosing the redundancy level in a dynamic and adaptive way. 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 the defined optimization problem.
- A simulation study showing 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.

4.2 State of the art

4.2.1 WSAN integration in IoT realm

In the literature we can find several proposed solutions for the problems related to WSANs integration to the IoT realm, [39, 31, 92, 87, 48, 20]. However, we could not find a proposal that targets both the in-network data storage and the data retrieval. In [61], 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 [31, 92, 87, 44, 5] and [20], the authors propose different models for the WSANs integration in IoT, as well as for the access to the data produced by 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 [82], 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, all the approaches proposed in these related works reinforce the results obtained by us in Chapter 3, 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 direct communication between source and consumer nodes.

4.2.2 Data storage in IoT/WSANs

On the other hand, the works that perform in-network data storage do not assume the WSAN integration in the IoT realm, [6, 25, 19, 93, 43, 29, 37]. In order to overcome this shortage, we proposed in Chapter 3 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 [52], a scenario that is favorable to localized IoT applications only. This choice has been made for a fair comparison with the solution proposed in [52], the Balance Storage (BS). As a reminder, 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 the 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 [52] 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 relationship with the entities allowing the integration of WSANs into the foreseen bigger picture of the IoT, concentrating only on localized IoT applications.

4.3 System model

In this section we recall the VBC scheme, as proposed in Chapter 3, and present the communication model proposed here for its integration to the IoT realm.

4.3.1 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 physical process 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, controllers 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 4.1.

In our system the access to the data can be performed at 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



Figure 4.1: System model: an example

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 example, a multi-source data or measurements summarizing some value in a short period of monitoring. Figure 4.2 illustrates the relationships among the entities in the VBC system model.

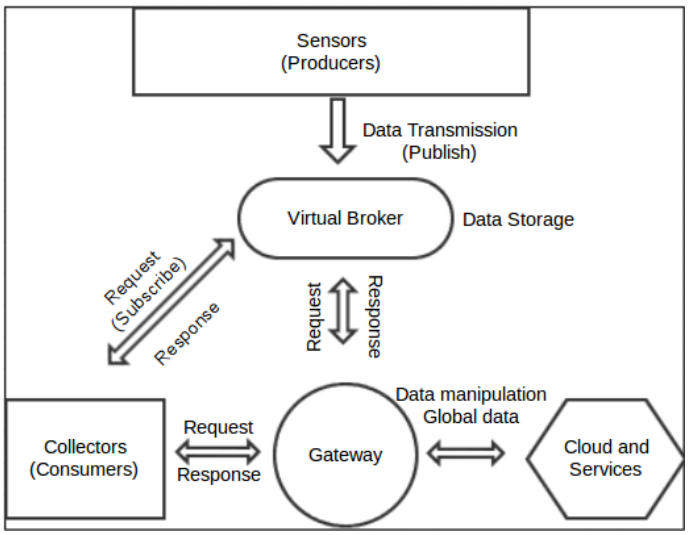


Figure 4.2: System model: iteration among entities

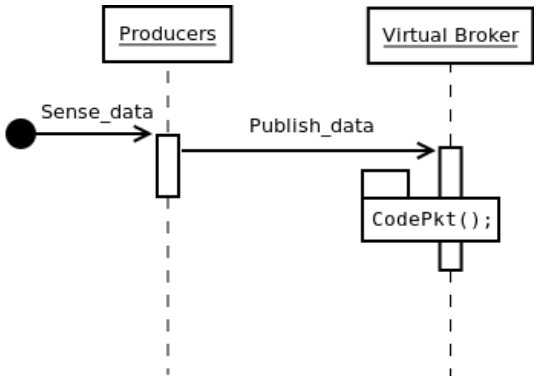


Figure 4.3: Sequence diagram of the publish operation

4.3.2 Storage scheme

The VBC storage scheme proposed in Chapter 3 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 Chapter 3, 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 $\mathbf{x_1}, \ldots, \mathbf{x_N}$ and are represented as a coefficient matrix $N \times M$:

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

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.

4.3.2.1 Publish operation

This operation is not modified comparatively to the one proposed in Chapter 3. 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 4.3.

4.3.2.2 Subscribe operation

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 consider now 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. The complete sequence diagram of the subscribe operation is presented in Figure 4.4.

- The devices subscribe for information to the closest VBNs
- The VBN forwards the request to the VB
- The VBNs that have the requested information send an answer directly to the device

The second case is similar to the one described in Chapter 3 for the subscribe operation.

4.3.3 Application Example

As an example of small scale we assume a scenario of a health application where this model can be used. We assume a home equipped with sensors that make environment and physiological monitoring, a gateway that intermediate the communication between the house network and a hospital through the cloud, the sensors that make health monitoring of the patient, and the VBNs which store the data gathered by all producers. In this case, as we have a small network we can see the entire network as one partition with only one VB. In this type of application the data availability is crucial to the patient's heath. Although it may be a small network, the data reliability is associated with high priority which is why our model fits well with the application requirement.

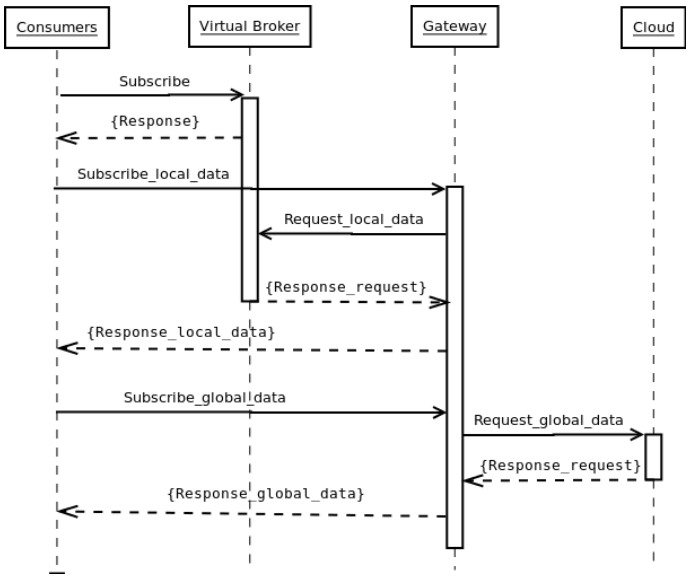


Figure 4.4: Sequence diagram of the subscribe operation

Indeed, the idea of VB combined with the redundancy provided by the coding technique used to store the data, instead of using a cluster based model or a VB without redundancy, ensure the availability and reliability of data even when some node are unavailable either due to failure or as a result of a node activity scheduling. When it is considered node activity scheduling in the network, despite of the data availability, our model also saves nodes energy and improves the network lifetime. Besides, our approach allows the access to a group of packets per request which is an advantage for this application, since a doctor in a hospital have to check many measurements at time to be able to diagnose a problem.

After describing the systemic view of the data access management in WSAN 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 while ensuring an optimal cost-benefit trade-off.

4.4 Theoretical analysis: a generalization

In order to choose the redundancy level that results in the optimal cost-benefit trade-odd, we need to derive the expressions that defines this metric that are: the communication cost and the packet delivery ratio. In Chapter 3, we presented a detailed analysis of the communication cost and the packet delivery ratio of VBC assuming the scenario proposed in [52] with a virtual broker of a fixed size of 4 VBNs. In this work, after making a few simplifications in the assumptions adopted in this chapter, we derive general expressions assuming 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.

4.4.1 Packet delivery probability

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:

$$P = \sum_{j=N}^{M} \binom{M}{j} (1-\sigma)^{3j} (1-(1-\sigma)^3)^{M-j} \tag{4.1}$$

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:

$$f_p = \frac{Tr_2}{N} \cdot (1-\sigma) \cdot P \cdot N \;=\; Tr_2(1-\sigma)P \tag{4.2}$$

4.4.2 VB communication Cost

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 $(1-\sigma)^2 M$ 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:

$$f_c = \frac{M(Tr_1)}{N} + (V + M(1-\sigma)^2)(\frac{Tr_2}{N}) \tag{4.3}$$

4.4.3 Cost-benefit trade-off

Equations 4.2 and 4.3 allow us to compute the cost-benefit metric 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 4.5 shows the cost-benefit performance of VBC with a VB of size 6 and several redundancy levels, varying the parameter σ.

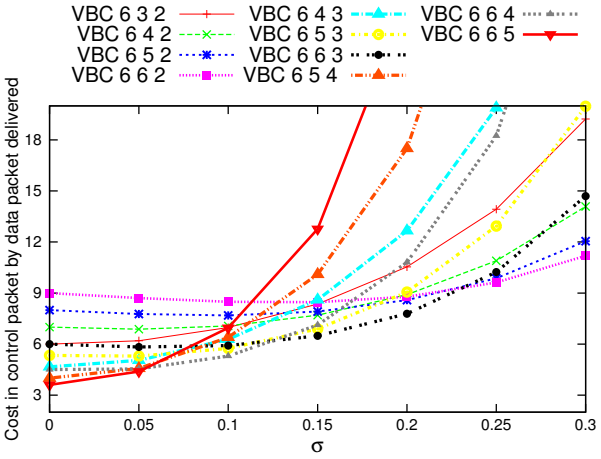


Figure 4.5: Cost-benefit with virtual broker of size 6

We clearly see in Figure 4.5 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, in order to provide an in-network data storage scheme with optimal performance (i.e. optimal cost-benefit trade-off) across all network conditions, there is clearly a need to propose a dynamic optimization of the redundancy level used in VBC. To do so, in the next section we formulate the problem of choosing the best redundancy level such as an optimization problem and present the technique we used to solve it. What makes this optimization 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 usage, 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.

4.5 Optimization 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.

4.5.1 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 [15]. 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:

$$S = \{(m, n, \sigma) \mid 1 \leq m, n \leq V, m \geq n, \sigma \in O\} \tag{4.4}$$

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:

$$\mathcal{A} = \{(M, N) \mid M \leq V, N \leq M\} \tag{4.5}$$

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 [57] and defines the transition matrix of the Markov chain that represents the network behavior.

$$P_\sigma = \begin{bmatrix} P_{BB} & P_{BN} & P_{BG} \\ P_{NB} & P_{NN} & P_{NG} \\ P_{GB} & P_{GN} & P_{GG} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.7 & 0.3 \end{bmatrix} \tag{4.6}$$

Finally, the reward function $r(s, a)$ is basically the cost-benefit metric obtained from Equations 4.2 and 4.3. 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:

$$r(s, a) = \frac{[f_c(s)]^{p_1}}{[f_p(s)]^{p_2}} \tag{4.7}$$

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:

$$v^{\pi}(s) = \lim_{T\to\infty} E\Big[\sum_{t=1}^{T} \lambda^{t-1} \cdot r(X_t, A_t^{\pi})\Big] \tag{4.8}$$

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 4.8, in the sense that $v^{\pi^*}(s) \leq v^{\pi}(s)$, for all π, s.

4.5.2 Dynamic Adaptive VBC algorithm (DA-VBC)

As it is well-known, the optimal expected discounted reward satisfies the following recurrence relation, called the Bellman's Equation:

$$v(s) = \min_{a\in\mathcal{A}} \Big\{ r(s,a) + \sum_{j\in S} \lambda p(j|s,a) v(j) \Big\} \tag{4.9}$$

On the right-hand side of Equation (4.9), 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 (4.9) 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 (4.9) such as value iteration, policy iteration and linear programming. In our proposed DA-VBC algorithm (see Algorithm 1), we use the value iteration algorithm defined in [15].

Algorithm 1 is performed at the gateway that is also responsible for monitoring the network in order to detect any change and allow the adaptation of the redundancy level adopted by the VBNs. The pre-computation phase is performed off-line (although could be performed online given that the gateway is a more robust device) and results in the optimal police π^*. The latter is recorded as a look-up table and determine the optimal action defined to each state in the Markov chain according to the current network conditions. Thus, after receiving the initial action (the redundancy level) sent by the gateway, the VBNs start coding packets with the redundancy level that ensures the optimal cost-benefit trade-off assuming that initially the network performs under good conditions. Thereafter, the gateway performs the operational phase, where according to the observed behavior (σ) in the network, such as the increase or decrease in the packet loss, it chooses the action $\pi^*(s) = (m, n)$ based on both the current state s and the optimal deterministic policy π^*, and broadcasts to the VBNs the updated optimal redundancy level.

4.6 Performance evaluation

4.6.1 Simulation settings

In this section, we evaluate the performance of DA-VBC based on the the expected total reward defined in Equation 4.7 to find the optimal police π^*(represented in the

1: **Input:** $s_0 = (n_0, m_0, \sigma_0)$, $\varepsilon > 0$
2: **Output:** π_t^*
3: **Pre-computation 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** $t \leq T - 1$ **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: DA-VBC Algorithm

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 WSAN, we do not have another similar method to compare with DA-VBC. Therefore, we compare it with the performance of our benchmark BS [52], the previous original VBC scheme, and with two heuristic-based policies. The heuristics are: the policy π^{Cost} which always choose the state with the lowest cost, and the policy π^{Benef} which always choose the state with the greatest packet delivery ratio. We also show the impact of different application weight factors assigned to $\pi^{mdpCost_2}$ and $\pi^{mdpBenef_2}$, metrics used to calculate the expected total reward of the DA-VBC MDP-based optimal policy.

We perform our evaluation 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 4.6, with four publishers and four subscribers producing data packets and requests, respectively, at each 3s. The node in the center represents the gateway defined in our system model (responsible for finding the optimal policy applied in the WSAN) 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 4.10.

$$P_\sigma = \begin{bmatrix} P_{BB} & P_{BN} & P_{BG} \\ P_{NB} & P_{NN} & P_{NG} \\ P_{GB} & P_{GN} & P_{GG} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.7 & 0.3 \end{bmatrix} \tag{4.10}$$

In our simulations, we define $\lambda = 0.99$ and we have a VB of size 8. We defined this

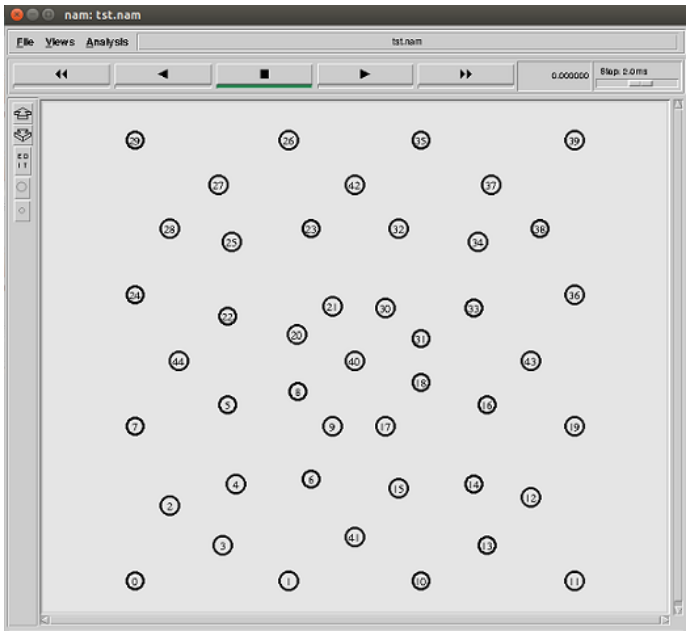


Figure 4.6: Scenario of simulation.

VB size because we assume that the network is partitioned according to the algorithm proposed in [51] where they find the best VB size according to the nodes' density in the partition, that is 8 in our case. 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 with $M \geq N$. 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 $M \geq N$.

4.6.2 Simulation results

Now, let us discuss the impact of π^{mdp} and the other heuristics regarding to three studied 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 the 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 as well as BS.

4.6.2.1 Comparison of π^{mdp} and fixed redundancy levels

Figures 4.7, 4.8 and 4.9 show the performance of π^{mdp} compared to BS and VBC with different redundancy levels. In Figure 4.7, we clearly see that 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 (4.8) and the cost-benefit (4.7). Figure 4.8 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 4.9 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 Chapter 3 that, although introducing redundancy we achieves the delivery of more packets with a lower cost.

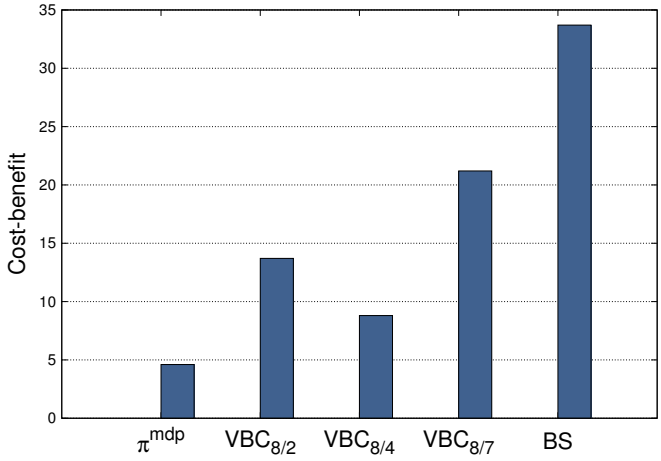


Figure 4.7: Cost benefit

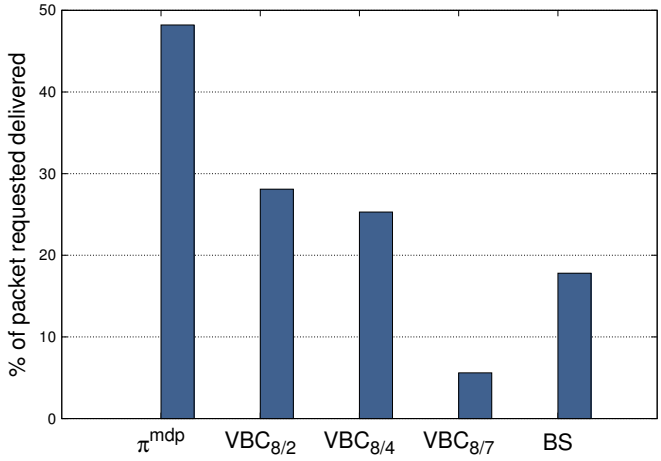


Figure 4.8: Packet delivery

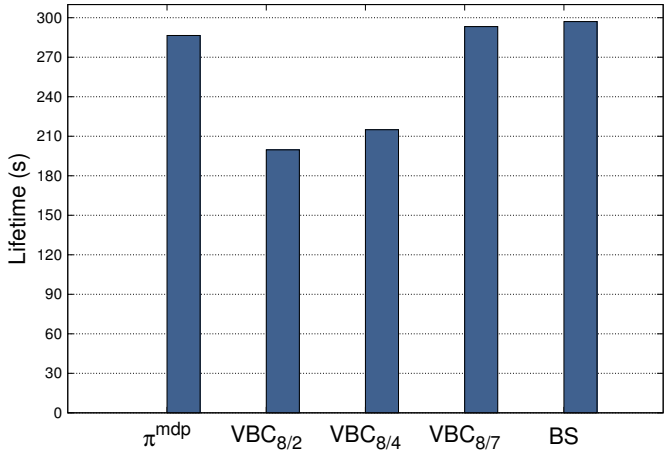


Figure 4.9: lifetime

4.6.2.2 Comparison of π^{mdp} and heuristic-based redundancy levels

Now, we discuss the results of π^{mdp} when compared to the heuristics π^{Cost} and π^{Benef}. Observing Figures 4.10, 4.11 and 4.12 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 requested packets delivered. As we can see in 4.10, 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} and 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 all the time, 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 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 4.12 the improvement in the network lifetime. Besides, we observe the decrease of the cost-benefit compared to π^{mdp} in Figure 4.10 and the increase of the packet delivery compared to π^{Cost} in 4.11. 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}.

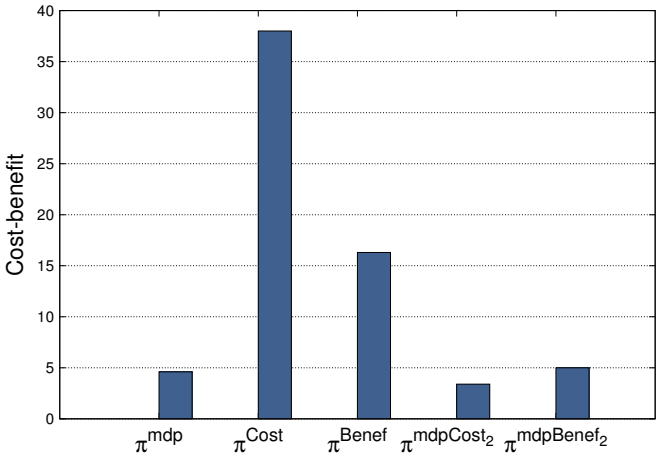


Figure 4.10: Cost benefit

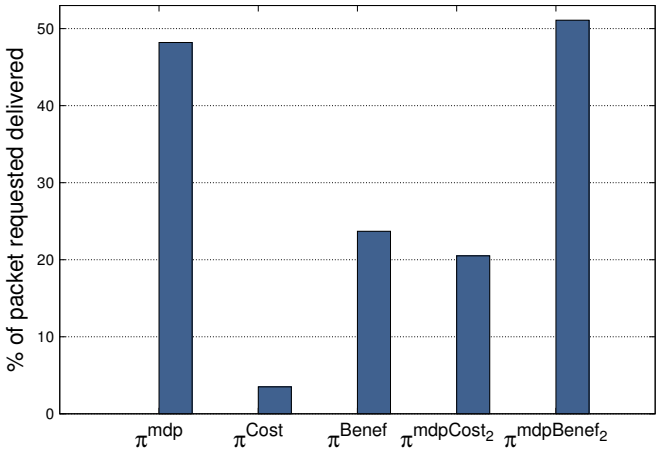


Figure 4.11: Packet delivery

4.7 Conclusion

In this chapter, we proposed a communication model that integrates the VBC data storage scheme introduced in Chapter 3 in an IoT usage 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 conditions represented by σ, the average loss probability of end-to-end communications. In order to achieve 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 our 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 changes in the state of the network.

The simulation results corroborate those presented by our theoretical analysis con-

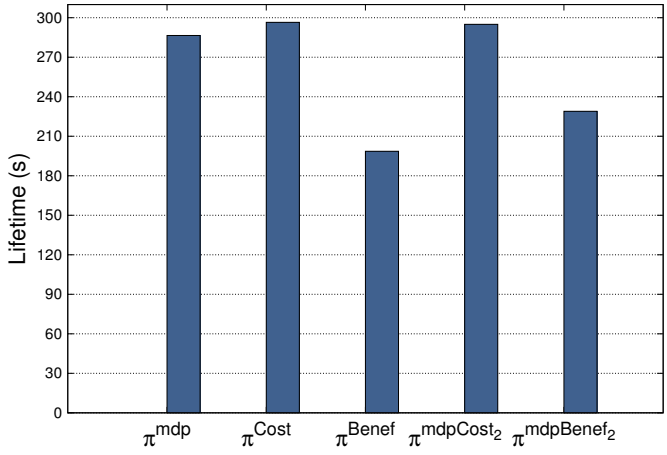


Figure 4.12: lifetime

firming that performing 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 usage.

In this chapter, as well as in the previous one, we considered the case of always-on WSANs (i.e. WSANs without node activity scheduling). However, this is not always the case in practice. In the next chapter, we are interested in evaluating our solution in a not-always-on WSAN. To do so, we start assessing the benefit of an adaptive duty cycle mechanism (i.e. a node activity scheduling) in a general case by improving the duty cycle mechanism defined in the IEEE 802.15.4 standard. Then, in order to evaluate the application of such a mechanism in the specific case of DA-VBC-based WSANs, we propose a duty cycle mechanism to be introduced in the VB, which is the part of the network that consumes the more energy.

Chapter 5

Node Activity Scheduling and In-Network Data Storage: Impact of Introducing duty-cycling on DA-VBC

Contents

6.1 Perspectives 94

In the previous two chapters we were interested in two problems: 1) the problem of data reliability in in-network data storage systems, for which we proposed the VBC scheme; and 2) the problem of enabling VBC to perform in its optimal state, adapting itself according to the network condition, for which we proposed the DA-VBC. Nonetheless, both of them are evaluated assuming always-on WSANs. However, as most of WSANs use the IEEE 802.15.4 standard and it proposes a duty cycle mechanism as an option in its operation mode, it happens that, in some situations, the nodes in a WSAN may performing under a nodes' activity scheduling resulting in a not-always-on WSAN.

In this chapter, in order to understand the impact of using duty cycling in the WSAN's performance, we first present a performance study of this mechanism while in the general case of use of WSANs, i.e. not specifically the case of data storage. To do so, we start proposing an improvement in the IEEE 802.15.5 duty cycle mechanism in order to make it dynamic and adaptive, as well as an evaluation of its performance assuming the general case of use of WSANs handling diverse traffic patterns. Then, inspired by the good results attained in the general case, we evaluate the impact of a node activity scheduling when introduced in the specific case of DA-VBC-based WSANs. Given that the nodes in the VB structure are those most used by the data storage schemes proposed in this work, we aim at assessing the impact of scheduling these nodes' activity on the cost-benefit trade-off. The objective here being to evaluate the use of DA-VBC in not-always-on WSANs.

The remainder of this chapter is organized as follows. In Section 5.1, we present

the context and the issues of this contribution and in Section 5.2 we discuss the related works. Then, in Section 5.3, we present our approach to the node activity scheduling in the WSANs' general case, as well as the performance evaluation of the adaptive duty cycle algorithm that we proposed for this case. We continue this chapter with Section 5.4, where we present the duty cycle mechanism introduced in the specific case of DA-VBC-based WSANs as well as a discussion about the performances of DA-VBC assuming a not-always-on WSAN. Finally, Section 5.5 concludes this chapter.

5.1 Context and issues

In the last few years, the rapid and growing progress in the development of modern and new devices allowed the use of Wireless Sensor Actuator Networks (WSANs) in a large range of applications. According to [12], the use of WSNs in real-life applications is also due to the advent of the IEEE 802.15.4 standard, which defines the physical and Medium Access Control (MAC) layers of the WSAN's protocol stack; as well as the ZigBee specification, which covers the WSAN's network and application layers. As a consequence, WSNs have been largely employed in many application domains, such as public safety, environmental monitoring and smart homes. This type of networks is characterized by its resource constraints, mainly concerned with the nodes energy. To make it even more suitable to networks with energy limitations, the 802.15.4 protocol is proposed to implement node's activity scheduling through a mechanism named duty cycling.

While keeping in mind our ultimate objective of studying the operation and performance of DA-VBC while used in VBNs performing under a duty cycle mechanism, we first study the performance of such a mechanism in the 802.15.4-based WSANs without the DA-VBC scheme. The duty cycle consists in performing a node activity scheduling. The mechanism is employed by the 802.15.4 protocol when it operates in the beacon-enabled mode. In this operation mode, a superframe structure (see Figure 2.4) is used in order to manage the duty cycle mechanism. The timeline is divided in Beacon Intervals (BI), which in turn, are divided into two periods (i.e. each BI): an active period, which represents the Superframe Duration (SD), and an inactive period. The SD defined in each BI and the BI itself are calculated based on two MAC protocol parameters, the Superframe Order (SO) and the Beacon Order (BO), respectively. In this beacon mode, the relation between SD and BI defines the network duty cycle (See Chapter 2 Section 2.2.1, for more details). However, although this protocol works well with respect to the energy, memory and processing consumption; it does not take into account the dynamic nature of the sensors activity in the case where multiple application usage, with different traffic patterns, are involved. Without considering the external stimuli, the duty cycle adopted initially can lead to a decreased application quality.

Thus, in the first part of this chapter, we propose a dynamic and self-adaptive algorithm named Dynamic Beacon interval and Superframe Adaptation Algorithm (DBSAA) aiming at improving the performance of the 802.15.4 duty cycle mechanism in the above mentioned case. DBSAA gives the network the ability to handle the trade-off between saving energy and meeting the application demands, in terms of delay and throughput. To do so, DBSAA changes the duty cycle through the adjustment of two

MAC protocol parameters, BO and SO. Its main objective is to provide more flexibility in duty cycle adaptation in order to increase the network throughput when needed and decrease the end-to-end delay while keeping saving energy.

Motivated by the results reached with DBSAA in the nodes energy saving assuming the general case of WSAN's (Section 5.3), we move forward in our study by addressing the performance of DA-VBC assuming a not-always-on WSAN. To do so, we define a new version of DA-VBC named DCDA-VBC that performs with VBN activity scheduling. DCDA-VBC takes advantage of the flexibility offered by the DA-VBC in the redundancy level adaptation to offer the possibility of putting some broker nodes to sleep. It expects with that to be more compliant with the IEEE 802.15.4 operations without compromising the cost-benefit trade-off.

5.2 Related works

The literature presents different approaches for an adaptive implementation of the duty cycle mechanism proposed in the IEEE 802.15.4 standard. Some of them change the duty cycle without analyzing the values of the parameters that define the duty cycle itself, as can be seen in [46]. On the other hand, there are a few works that adapt the duty cycle by changing only one of the MAC layer parameters, either BO or SO, [64] and [45]. As we explain next, this leads to limited impacts while targeting improving the throughput usage, end-to-end delay, and energy usage.

In [46], the authors propose a congestion control mechanism that implements an adaptive duty cycle to do resource and traffic control. The duty cycle adaptation takes place when the algorithm detects the network congestion. In this situation, the adaptation procedure takes into account the required service time and two thresholds that define the minimum and maximum nodes active time allowed. The service time is calculated in order to know the active time necessary to transmit the data decreasing the network congestion. Although the congestion control is accomplished, as they do not consider the BO and SO values in their duty cycle adaptation and the active time is limited by thresholds, this solution may impose, in certain cases, delay in the packet delivery. Also, in [64], the authors adapt the duty cycle by changing the BO value. The only goal of the adaptation is to save energy through the modification of the time that the nodes remain in sleep mode. Consequently, the need for more or less active time (defined by the SO) to handle the network traffic is not considered. Thus, as the SO is unchanged, when the traffic load increases, the application quality decreases.

Alternatively, the Dynamic Superframe Adjustment Algorithm (DSAA) presented in [45] changes only the SO parameter intending to decrease the energy consumption and to improve the channel utilization. The proposed algorithm makes its decision based on the superframe occupation ratio, the nodes collision ratio, and in two thresholds defined to limit the maximum and the minimum occupation and collision ratio accepted by the network. Relying on these values, the proposed algorithm decides either to increase or decrease the SO value. In this case, the application needs are taken into account as the active time is adapted. However, the BO value defines how much time the nodes wait before sending their new packets. In summary, the solution allows the network to save energy and to deliver the data; but as the BO is not modified, the packet delivery delay is increased.

There are also theoretical approaches proposed to define the optimal operation of the duty cycle. In [42], a Markov decision process-based theoretical analysis is proposed with the aim of saving more energy and meeting the application requirements. Looking for defining the optimal operation of the duty cycle dynamically, a solution based on reinforcement learning is presented in [26]. These solutions based on mathematical models show good accuracy with respect to the optimal values defined for both, the BO and SO, parameters. However, they are computationally complex solutions and, to the best of our knowledge, they have not been implemented in real networks to prove their practical feasibility.

Despite the fact that the works described above perform duty cycle dynamic adaptation, they are limited either by the choice of just one parameter, either the BO or the SO in the duty cycling definition, or by the complexity of their practical implementability. Therefore, in order to assess the benefit allowed by the duty cycle mechanism, we propose a dynamic and self-adaptive algorithm to control the duty cycle in an 802.15.4 wireless sensor network that is simple and acts on both BO and SO. This study on adapting duty cycling in traditional WSNs is preliminary to the introduction of the duty cycling in the Dynamic Adaptive Virtual Broking Coding (DA-VBC) scheme.

Concerning the duty cycle in a virtual broker structure, for the best of our knowledge, we are the first to study such an approach. For this reason, the study that we present in Section 5.4 can be considered as the state of the art in this domain.

5.3 Duty cycle adaptation in traditional WSN

5.3.1 Problem description

The IEEE 802.15.4 standard was developed to take into account the energy, memory and processing limitations of WSNs. If we consider the energy problem, the protocol saves energy putting the network nodes into sleep mode during some periods of time through the duty cycle mechanism. Nevertheless, the 802.15.4 standard does not offer means to adapt its duty cycle mechanism which is statically set up.

Choosing a duty cycle setting in runtime is a complicated task and depends on the network objectives. Superficially, we know that setting a low duty cycle means a low energy consumption whereas a high duty cycle means that a bigger number of packets can be transmitted but with greater overall energy consumption. It is thus clear that there is a need for adapting the duty cycle to better accommodate the throughput needs and the energy constraints of the network. As the duty cycle is defined in function of the BO and SO values, these two parameters play a significant role in this self-adaptation.

With fixed BO and SO values, we can easily compute the maximum energy amount which will be consumed, the maximum number of packets that can be served, as well as the achievable end-to-end delay. For instance, if a duty cycle of 50% with BO = 2 and SO = 1 is set, as the difference between BO and SO is small, the nodes are able to save energy. Besides, as BO is also small, important delays in the packet delivery are avoided. However, if a duty cycle of 50% with a BO = 10 and SO = 9 is defined, we have a long sleep time and, even if all packets are delivered during the active time, there will be a significant delay between packets delivered which can be problematic

for the intended application. Further, it can still bring on high contention between the nodes at the beginning of the following beacon interval causing energy waste.

Aiming at avoiding the problem of adapting the duty cycle only thinking about the percentage of active time, we consider in this work a duty cycle adjustment that finds appropriate values for the BO and SO parameters such that the energy consumed by the nodes is minimized while the application requirements are fulfilled.

5.3.2 DBSAA: a self-adaptive duty cycle approach

This section describes our solution called DBSAA (Dynamic Beacon interval and Superframe Adaptation Algorithm). Before describing this solution, let us first define the considered network assumptions. Next, the methodology used to estimate the variables that characterize the network traffic are expound. Finally, we describe DBSAA operation.

Assumptions In order to develop our algorithm, we made some assumptions on the network operation mode and configuration. First, the algorithm is proposed for 802.15.4 network using the beacon-enabled operation mode in a star topology. Second, we decided to consider only the CAP portion of the active period. This choice was based on the same justifications as those presented in [50], such as, the limitation in the number of GTS slots, which can be allocated in each beacon interval (at most seven), and the fact that the GTS slots are optional in an 802.15.4-based network.

Methodology The methodology chosen in this work is based on two premises. The first one consist in the assumption that the packet collision ratio is directly related to the number of sensor nodes sending data. The second one specifies the relationship between the number of the packets received by the coordinator and the superframe occupation. The DBSAA algorithm deliberates upon these two premises to decide when and how to change the BO and SO values. Besides, the algorithm is executed only by the coordinator and utilizes the information estimated in it. Therefore, it is not necessary to create new messages and modify the 802.15.4 standard.

In order to get the required information to make a decision, DBSAA follows the three steps described below:

- Step 1: DBSAA estimates the network load based on the number of packets received by the coordinator and on the number of nodes that have sent packets during a measurement window enclosing the N last beacon intervals.
- Step 2: DBSAA determines if there were changes in the network load during the N last BI and delineates the type of the variation which has arisen.
- Step 3: DBSAA calculates a factor α which assigns the number of beacon intervals that the coordinator should wait before triggering the DBSAA algorithm. The factor α is defined to avoid either a too long or a too short waiting time to perform DBSAA.

The Step one is easily performed through simple measurements made by the coordinator over the last N intervals. Then in the Step two, the coordinator computes the superframe Occupation Ratio (OR) and the Collision Ratio (CR) using the information collected in the first step. In DBSAA, OR and CR, which estimate the current network load, are computed by the equations (5.1), (5.2) and (5.3):

$$OR = \frac{numPkt \times Tmp_{trans}}{2 \times SD} \tag{5.1}$$

$$Tmp_{trans} = \frac{tamPkt}{250} + T_{ack} + ACK + IFS + (2 \times CCA) \tag{5.2}$$

$$CR = \frac{(numN \times NumPSD \times 2) - numPkt}{numN \times NumPSD \times 2} \tag{5.3}$$

In Equation (5.1), $numPkt$ is the sum of packets received by the coordinator during the last N beacon intervals, and Tmp_{trans} is the time necessary to transmit a data packet. In Tmp_{trans} we do not consider the backoff time because it has no influence in the overall occupation ratio, since when the nodes are performing a backoff, they stay in idle mode which allows the other nodes to send their packets. In Equation (5.2), $tamPkt$ is the packet size in bits, T_{ack} is a turn around time waited by a node before sending an ACK, IFS is the time that separates two successive frames transmitted from a device, and CCA is the time taken to realize the clear channel assessment procedure that verifies if the channel is available. In Equation (5.3), $numN$ is the number of nodes who sent packets in the last N intervals, and $NumPSD$ is the number of packets that the source nodes generate during a SD. The variable $NumPSD$ is calculated based on the data rate used in the network.

In addition, in Step two, a couple of other variables have their values set, they are: $change_numPkt$ and $change_numNodes$. These variables represent the direction of changes associated with the number of nodes and packets. At the end of Step two, these variables can assume two different values: 1 when the number of packets/nodes increases and 0, otherwise. Consulting these values, the coordinator can detect the traffic load changes.

In Step three, the factor α is computed as indicated in Equation (5.4). This factor is responsible for adapting the number of waited intervals ($num_interval$) by the coordinator to alter the BO and SO values. The factor α has a different impact in the network performance depending on the BO value. If we have a large BO, the BI also assumes a large value. Considering the computed value of α, it may take a long time until the coordinator makes the duty cycle adaptation, which can result in delay in the packet delivery. On the other hand, if BO assumes a small value, the BI becomes very short, and the frequent checking of the network load may result in unnecessary consumption of energy. The α adaptation is thus an important step in DBSAA performance. In addition, Equation (5.4) considers

Table 5.1: β values.

OR	0% - 25%	26% - 50%	51% - 75%	76% - 100%
β	4	3	2	1

an OR factor, β, which represents the occupation ratio and helps decreasing the α value when there is an intensive traffic in the network. See Table 5.1 .

$$\alpha = \frac{(15 - BO) + \beta}{BO} \tag{5.4}$$

5.3.3 DBSAA algorithm

After accomplishing the steps described above, the coordinator can call the DBSAA algorithm to select the BO and SO adjustments that are the more adequate to the current network state. In order to take the application requirements into consideration, DBSAA uses two thresholds: the *THoccupation* and the *THcollision*. These thresholds are used in the algorithm to characterize the traffic behavior and are defined by the application according to its needs. They represent the application sensitivity to delay and packet loss. The detailed operation of DBSAA is presented in Algorithm 2.

Initially, the *num_interval* value is checked (line 2). If the *num_interval* reaches α, the network load is verified by examining the *change_numPkt* variable (line 3). When *change_numPkt* value is 0, we may have two different situations: either the nodes had no packets to send, or they could not send any because of collisions. First, we try to eliminate the possibility of collision. For that, the channel state is checked out (line 4). If the channel is busy, this means that collision probability is high. In that case (lines 5-6), if the $BO - SO = 1$, the BO is decremented to avoid a high contention after a large interval. Otherwise (lines 8-10), if the $BO - SO > 1$, DBSAA not only decreases the BO value but also increases the SO value to give more time for packet transmissions. On the other hand, if the channel is not busy, the OR is verified to assess the traffic conditions (lines 14-16). If the OR is smaller than THoccupation, the BO and SO values are not modified. Indeed, in this case the DBSAA assumes that we do not expect more packets to arrive, as the traffic load is tending to become smaller. In case OR is larger than THoccupation, and if SO is smaller than BO, SO is increased. Indeed, increasing SO and thus SD, gives more time for the nodes to send their packets, which improves the delivery rate.

Considering the case where *change_numPkt* is 1 (line 18), the next step is to monitor the *change_numNodes* value (line 19). If *change_numNodes* is 1, this may indicate an increase in the traffic intensity due to a growth in the number of nodes sending data. In this case (lines 20-24), the CR is compared to THcollision. If THcollision is not exceeded, BO and SO remain unchanged. Otherwise, SO is updated according to the difference between BO and SO (since SO is limited by the BO value). If BO is larger than SO, SO is increased in order to decrease the contention among the nodes to access the channel. Therewith, the nodes will have more time to send their packets and, as the active time becomes longer, they will need a fewer number

of slots to send their packets. As a result, the collisions will decrease. In the case where the *change_numNodes* is 0 (line 25), the DBSAA assumes that the number of packets increased due to the rise in the rate of sent packets by the sources. In this case, the active time in the beacon interval has to be increased to meet the application's requirements. To do so, the algorithm ascertains the CR and OR with relation to the THcollision and THoccupation to have an evaluation of the network load (lines 26). If one of the thresholds is exceed, the DBSAA checks if SO is smaller than BO, and in that case SO is increased by 1. In this way, the nodes stay active for more time whereas the BO remains the same to prevent an increase of delay in the packet delivery. If not, as SO is limited by BO, BO and SO are both increased, giving more active time to the nodes to transmit their data.

5.3.4 Performance evaluation

This section describes the DBSAA performance evaluation using NS-2 (Network Simulator 2). We simulate the IEEE 802.15.4 standard and the DSAA algorithm [45] to compare with DBSAA. In our implementation, DSAA underwent certain improvements as some details are not described in the original paper where it is proposed. These improvements are restricted to the definition of its collision and occupation ratio. Apart from these, the algorithm operation remains exactly the same as presented in [45].

5.3.4.1 Simulation setting

The DSAA ratios simulated are the same than those used in DBSAA, except for the fact that the DSAA ratios are based on the information of every interval time while, in DBSAA, they rely on the feedback of the N last intervals. In the DBSAA simulation, N is set to 2. We chose to keep the history of the two last BI because our simulations showed that this duration was sufficient to detect a significant change in the network load.

The performed simulations consider a network with a maximum link throughput of 250Kb/s. The network is configured in a star topology. Our traffic model tries to model the dynamism and the unpredictability present in the traffic generated by WSN applications in the context of smart-* (home, building, etc.). In order to achieve that, we simulate networks with 4 different number of nodes (10, 15, 20 and 25), where the number of source nodes changes dynamically over the simulation time (the start and stop time of each traffic is chosen randomly which results in different number of nodes sending packets during the network lifetime), and the simulation of each network is performed 33 times. The value 33 is chosen because, according to [38], 30 is the minimum number of samples required to find a reliable confidence interval. The variation of the source nodes is done in a random way and it is different in each simulation execution. The default values of the 802.15.4 standard variables are maintained and the power consumption in each mode operation utilized by the NS-2 simulator is showed in the Table 5.1.

Aiming to demonstrate the benefits of adapting the BO and SO parameters to achieve more appropriate duty cycles with regard to the application requirements and the power saving, we compare the performance of the three algorithm with respect to

```
1: At each beacon interval
2: if num_interval = α then
3:   if change_numPkt == 0 then
4:     if channel_status = BUSY then
5:       if BO − SO = 1 then
6:         BO ← BO − 1
7:       else
8:         if BO − SO > 1 then
9:           BO ← BO − 1
10:          BO ← SO + 1
11:        end if
12:      end if
13:    else
14:      if OR >= THoccupation and BO <= 14 and SO < BO then
15:        SO ← SO + 1
16:      end if
17:    end if
18:  else
19:    if change_numNodes == 1 then
20:      if CR > THcollision then
21:        if BO − SO >= 1 then
22:          SO ← SO + 1
23:        end if
24:      end if
25:    else
26:      if OR > THoccupation and CR > THcollision then
27:        if SO < BO then
28:          SO ← SO + 1
29:        else
30:          SO ← SO + 1
31:          SO ← BO + 1
32:        end if
33:      end if
34:    end if
35:  end if
36: end if
```

Algorithm 2: DBSAA Algorithm.

three metrics: the average energy consumption, the data delivery ratio and the end-to-end delay. In order to consider a general scenario, not so restrictive, the simulations are realized using a THoccupation value of 75% and a THcollision value of 30%.

5.3.4.2 Simulation results

Here we show the simulation results for the scenario with 10 and 20 nodes experiencing different duty cycles.

- **Packet delivery ratio** - Figure 5.1 displays more precisely the packet delivery ratio resulted when the WSN perform under a duty cycle of 50%. In this case, we have BO=2 and SO=1. This Figure shows that the 802.15.4 Std and the DBSAA perform better than DSAA. This can be explained by the fact that DSAA operations are based on the SO adjustment. As DSAA cannot adapt the BO value to save energy, it tries to adapt the SO. However, the SO is responsible for defining the active time and, for that, some changes in this parameter result in packet loss. Besides, in this scenario the difference between BO and SO is too small to allow good improvements by only adjusting the SO value.

  In addition, we remark that in a situation where the network has a high duty cycle and the gap between the BO and SO values is small, DBSAA performs well keeping the same delivery ratio presented by the 802.15.4 Std. The drop in the curve of delivery rate presented by the DBSAA in the Figure 5.2 is explained by the drop in the energy curve in the Figure 5.6. As can be seen in this curve, the nodes in the DBSAA run out their energy few seconds earlier than in the 802.15.4 std. In the Figures 5.1, 5.2, 5.7 and 5.8, which represent a scenario where the duty cycle is reduced and the gap between the BO and SO increases, the dynamic adaptation of BO and SO leads to important improvements related to the application demand. In Figures 5.3 and 5.4 we can see that DBSAA and DSAA deliver two times more packets than the 802.15.4 Std. This result is explained by the fact that the delivery ratio is directly related to the SO value, and both, DBSAA and DSAA, perform SO adaptation. So, the SO adjustment gives more capacity to handle the requested data load.

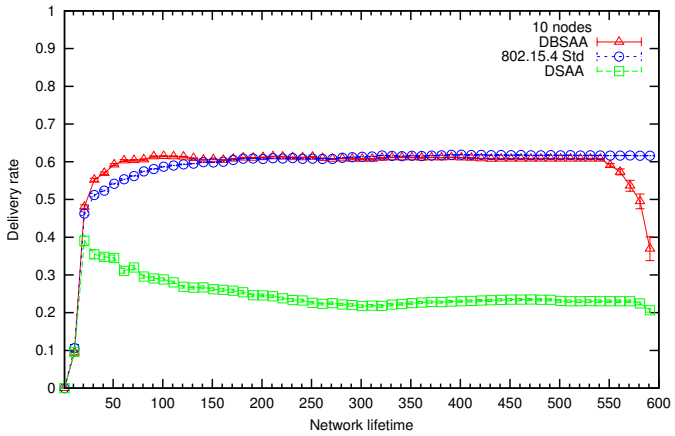


Figure 5.1: Packet delivery ratio DC 50% (BO = 2, SO = 1) 10 nodes

- **Energy consumption** - Regarding the energy consumption, Figures 5.5, 5.6, 5.7 and 5.8 confirm that the dynamic adjustment of the MAC parameters does not imply much energy waste. Analyzing these Figures, we can see that DBSAA and DSAA, in both scenarios, with DC = 50% and 6%, may waste a little more energy than the 802.15.4 Std in some cases. However, DBSAA compensates this energy waste by the accomplished low end-to-end delay presented in Figures 5.9 and 5.10.

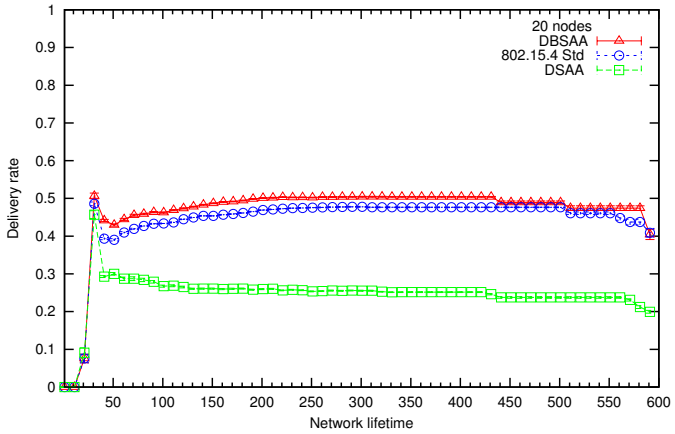


Figure 5.2: Packet delivery ratio DC 50% (BO = 2, SO = 1)20 nodes

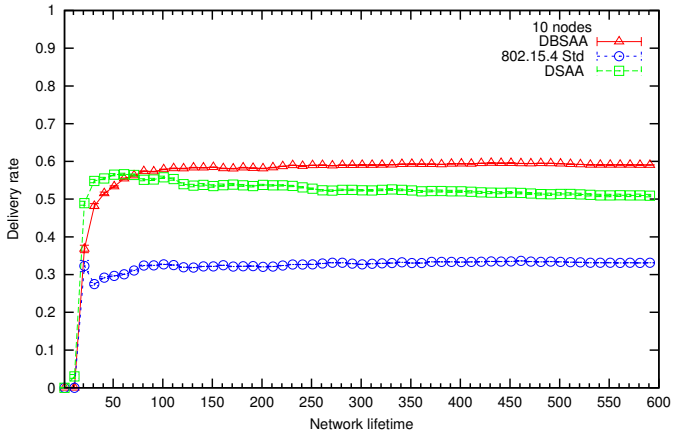


Figure 5.3: Packet delivery ratio DC 6% (BO = 6, SO = 2) 10 nodes

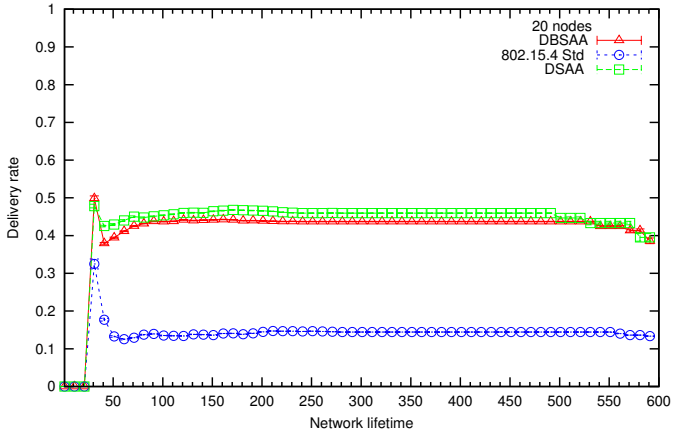


Figure 5.4: Packet delivery ratio DC 6% (BO = 6, SO = 2) 20 nodes

- **Delay** - Indeed, Figures 5.9 and 5.10 make it clear that adapting just the SO can produce a higher end-to-end delay than the result obtained through the adapta-

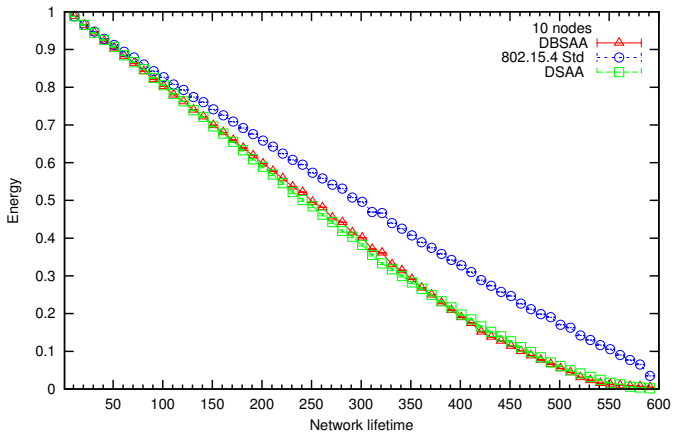


Figure 5.5: Energy average consumption DC 50% (BO = 2, SO = 1).10 nodes

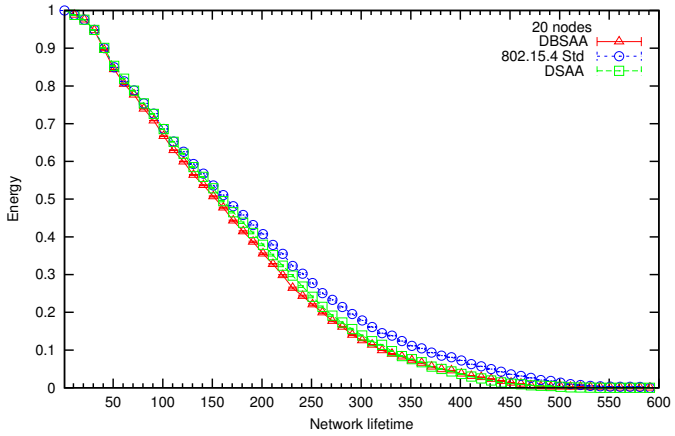


Figure 5.6: Energy average consumption DC 50% (BO = 2, SO = 1). 20 nodes

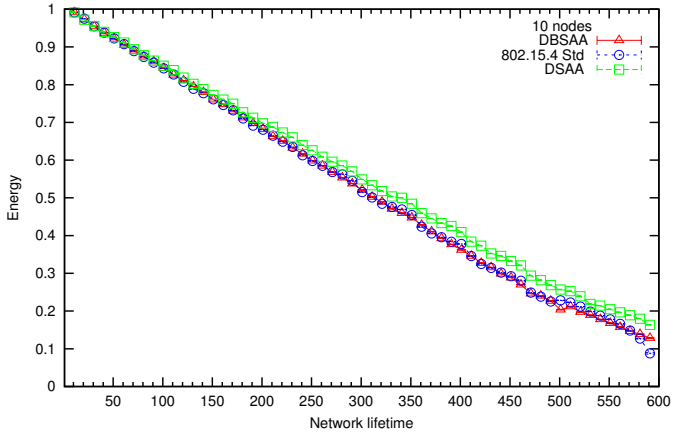


Figure 5.7: Energy average consumption DC 6% (BO = 6, SO = 2) 10 nodes

tion of the two parameters. In brief, the graphs show that even in a situation where the BO and SO cannot vary widely (Figure 5.9), DBSAA achieves good

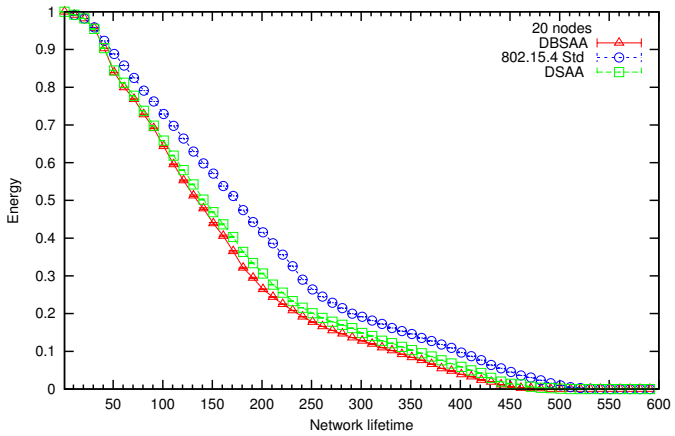


Figure 5.8: Energy average consumption DC 6% (BO = 6, SO = 2) 20 nodes

results in terms of energy consumption and end-to-end delay, whereas DSAA suffers the consequences of taking into account only the SO parameter in the duty cycle adaptation. This fact is even more evident in Figure 5.10, which shows the end-to-end delay result in a scenario where the BO and SO can vary in a larger range of values. DBSAA shows the relevance of looking for a balance between the BO and SO to fulfill the application needs by depicting the smaller end-to-end delay among the three protocols.

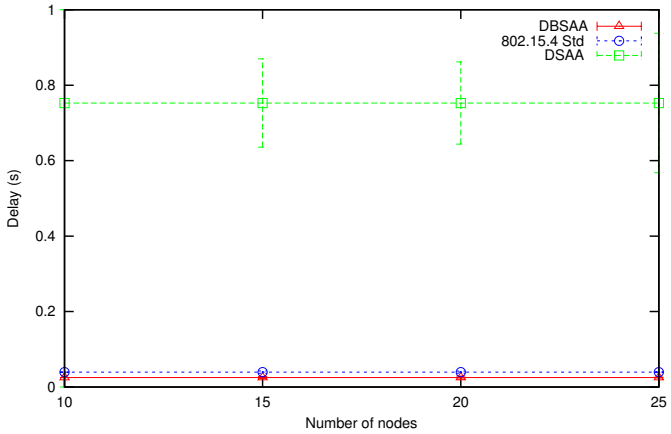


Figure 5.9: Delay DC=6% (BO=2,SO=1)]

- **Summary** - To sum up, we can say that DBSAA achieves a very good cost-benefit by performing duty cycle adaptation (i.e. in terms of the trade-off between the energy consumption and end-to-end delay). Therefore, after having showed the benefiting in adapting the duty cycle mechanism according to the traffic variation in the traditional WSNs case, lets discuss now the impact of using a node activity scheduling assuming the specific case of DA-VBC-based WSANs.

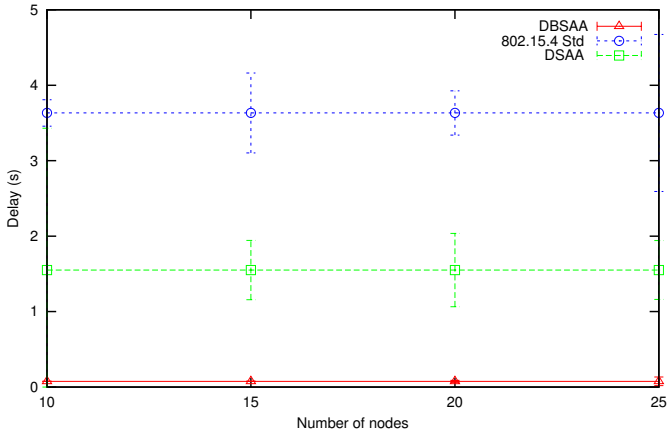


Figure 5.10: Delay DC=50% (BO=6,SO=2)

5.4 Duty cycle in DA-VBC-based WSANs

As shown in the previous section, putting some nodes to sleep in the 802.15.4-based networks, not only saves energy, but also improves the performance of the network, even in front of variable traffic condition. This fact encourages the use of duty cycle in WSANs and as such we found it necessary to see how the operation and performance of DA-VBC, our previously proposed scheme, evolves. Given that the broker nodes are the nodes that are the most sensitive to duty cycling in the VB architecture assumed by DA-VBC, we aim at scheduling these nodes' activity and evaluating the impact of this mechanism on the network's energy saving as well as on the cost-benefit trade-off of DA-VBC.

5.4.1 Theoretical analysis of the impact of duty cycling in a VB structure

We start with an analytical evaluation of the node activity scheduling integrated in the operation mode of the broker nodes. To do so, we assume that all VBNs may sleep with a probability γ, where γ defines, on average, the proportion of broker nodes that are in the sleep mode in a time interval t. We further assume the VBC scheme (i.e. without the optimization) in order to allow the evaluation of the impact of the duty cycling according to the fixed redundancy level adopted by the VBNs. For example, suppose that we have a VB with 6 VBNs and redundancy level of 4/2. In this case, as there are 2 redundant packets stored in the VB, we have that, in principle, up to two nodes can sleep to save energy without compromising the data reliability in the network. In this case, the VB may operate with $\gamma = \lfloor 1/3 \rfloor$ or $\gamma = \lfloor 1/4 \rfloor$, which means, in this example, one or two brokers nodes in the sleep mode, respectively. It is important to observe, however, that assuming a situation where at each point in time two nodes on average are sleeping, it may happen that two packets from the same generation of coded packets are lost and the other two are stored in those nodes that are sleeping. In this case, if some packets of this generation are requested, the delivery of the packets is affected.

Aiming to evaluate the performance of VBC assuming the VB working with duty

cycling γ, we realized some experiments where we assume a VB with 6 VBNs and all possible redundancy level combinations, the communication channel condition σ varying from 0 to 0.3, and γ varying from 0.25 to 0.33. In order to calculate the communication cost (which in this analysis also represents the node's energy consumption) and the packet delivery probability, we introduce γ in the expressions derived in Chapter 3 as well as generalize them (following the same train of thought used in the expression derivation made in Chapter 3) to work with a VB of size 6. We also plot the results obtained with the Balance Storage (BS) scheme used as a VBC benchmark in the Chapters 3 and 4, to have a reference scheme without redundancy.

We see in Figures 5.11 - 5.16 that, although the cost (energy consumption) decreases when we increase γ in the VB, the delivery ratio decreases even more. Indeed, when we observe the performance of VBC without duty cycle in Figure 5.12, we see that only two combinations of M/N (the redundancy level), those with low redundancy level, do not perform better than BS when $\sigma = 0.1$. However, when we look at Figure 5.14 with duty cycle, $\sigma = 0.1$ and $\gamma = 0.25$, VBC with almost all possible redundancy levels performs worse than BS. This means that, even with a redundancy level of one packet, when VBC works with a broker node in the sleep mode, the packet delivery ratio reduces significantly whereas the cost decreases slightly, a behavior that gets even worse if we increase σ to 0.33, as can be observed in Figures 5.15 and 5.16. Thus, we have that even though putting broker nodes to sleep saves energy, it results in a high packet loss ratio which impacts the overall performance of VBC.

Although, it was expected a decrease in the performance of VBC in delivering the requested packets, we would not expected such a big decrease for such small reduction in the communication cost. As we observe in Figures 5.11 - 5.16, even in the case where the duty cycling presents its biggest reduction regarding the cost, which means 6% with $\gamma = 0.25$ at $\sigma = 0.1$, and 13% with $\gamma = 0.33$ at $\sigma = 0$, the impact of the duty cycling is much bigger, in the negative way, on the packet delivery ratio. This latter presents, for the same values of γ and σ, a reduction of 11% and 22%, respectively. Assuming the worst case, the reduction in the packet delivery ratio can reach until 66% and 300% when $\gamma = 0.25$ and $\gamma = 0.33$, respectively.

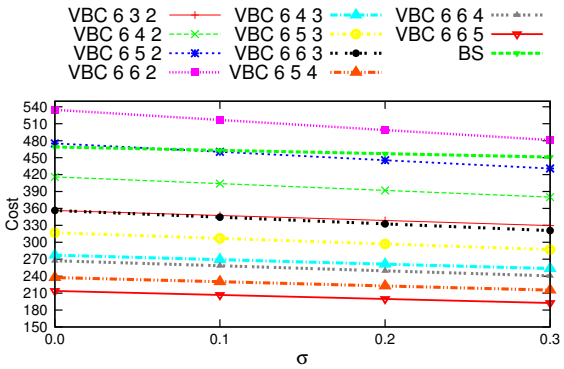


Figure 5.11: Cost without DC

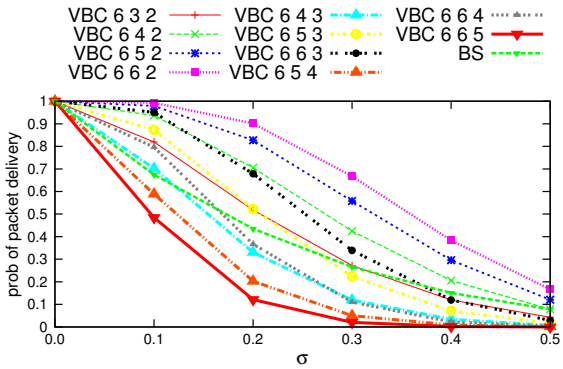


Figure 5.12: Packet delivery without DC

Nevertheless, although the duty cycle mechanism indeed decreases the energy consumption in the broker nodes, the most interesting aspect observed in this analysis is the fact that increasing γ does not means saving more energy if one is interested in

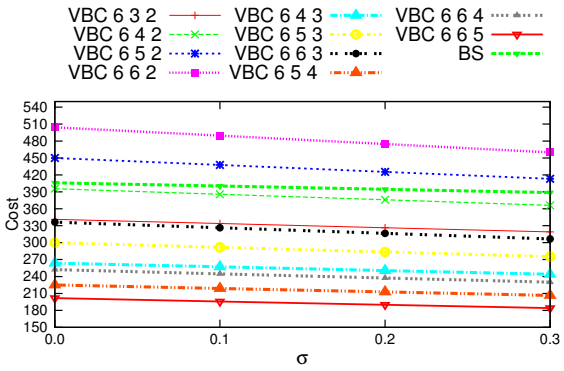


Figure 5.13: Cost with $\gamma = 0.25$

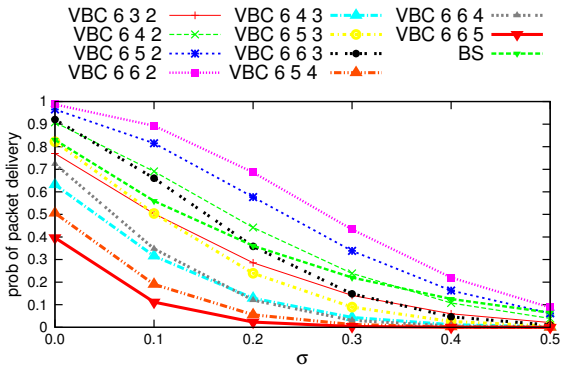


Figure 5.14: Packet delivery $\gamma = 0.25$

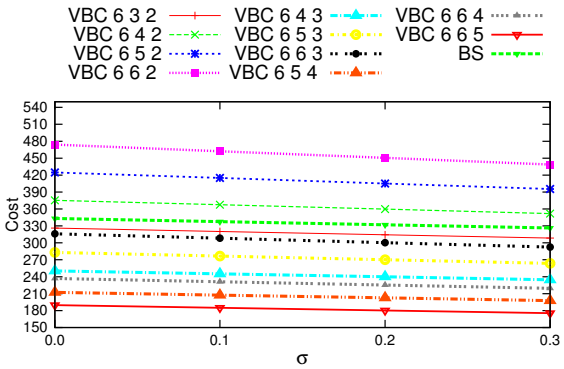


Figure 5.15: Cost with $\gamma = 0.33$

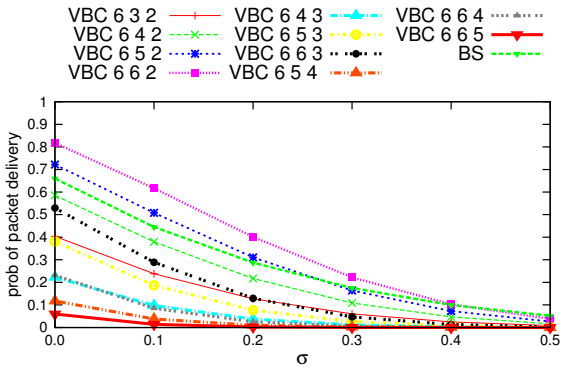


Figure 5.16: Packet delivery with $\gamma = 0.33$

reducing cost while keeping the same packet delivery ratio. Contrary to what one can intuitively expect, it is possible to keep the same delivery ratio (or very close) and save node's energy only adjusting γ and the redundancy level in the network. For instance, still observing Figures 5.11 - 5.16, we have that without duty cycle and redundancy level of 6/2, 100% of the packets are delivered with a cost of 530 messages. Now, assuming a duty cycle where on average one VBN is in the sleep state ($\gamma = 0.25$) at a time, we have that with the same redundancy level 6/2, approximately 100% of packets are delivered with a cost of 500 messages. Now, if one considers to put one more node to sleep (by setting $\gamma = 0.33$) keeping the same redundancy level, the delivery ratio drops to 80% and the cost reduces to 470 messages. Clearly, we see a reduction of 30 messages in the communication cost which results in energy saving. However, we have that if one assumes a redundancy level of 5/3 and $\gamma = 0.25$, the delivery ratio is 80% with a cost of 300 messages. Thus, one can keep the same delivery ratio achieved in the scenario with two nodes in the sleep mode ($\gamma = 0.33$) but with a much smaller cost. We conclude then that increasing γ is not necessarily the best way to minimize the communication cost while a data storage scheme is being used. The most important, though, is that we got an understanding of the way we have to manipulate the duty cycle and the redundancy level in the VB in order to reach the best cost-benefit of VBC. For this reason, in the next section, we discuss the broker nodes activity scheduling assuming that γ can be used as a parameter in the optimization model used by DA-VBC as defined in Chapter 4.

5.4.2 The DCDA-VBC approach

Now, let us present our node activity scheduling to be employed in the VB structure adopted in the DA-VBC-based WSANs. In order to attempt to introduce node's activity scheduling in the VB taking into account the performance of the packet delivery metric, we introduce γ as a parameter in the optimization problem modeled as a MDP in Chapter 4, and name this resulting algorithm as DCDA-VBC. The state of the MDP is now made up of 4 parameters: M , N, σ and γ. Using this approach we intend not to force the duty cycle, but, instead, allow its use when the situation in the network provides the necessary conditions for the duty cycle be worth it.

The MDP responsible for finding the best state according to the current network condition remains basically the same as defined in Chapter 4 (Section 4.5.1), changing only the definition of state and the reward functions. The state now is defined as a quadruple $s = (m, n, \sigma, \gamma)$ where the variable γ takes values in the set $G = \{0.5, 0.25, 0\}$, which represent the fraction of nodes which are put to sleep, associated respectively, with the good, normal and bad condition of the network. Formally, the *state space* is now defined as:

$$S = \{(m, n, \sigma, \gamma) \mid 1 \leq m, n \leq V, m \geq n, \sigma \in O\, \gamma \in G\} \tag{5.5}$$

Regarding the reward functions, we have to consider now that, in the publish and the subscribe operations, a packet can be lost when sent to a broker node in the sleep mode. Thus, besides the packet loss probability σ, we add in the reward function the variable γ that defines the probability of a broker node be sleeping. Taking this into account, the reward functions for cost and packet delivery probability are defined as:

$$P = \sum_{j=n}^{m} \binom{m}{j} ((1-\sigma)^3(1-\gamma)^2)^j (1 - ((1-\sigma)^3(1-\gamma)^2))^{m-j} \tag{5.6}$$

$$f_p = Tr_2(1-\sigma)(1-\gamma)P \tag{5.7}$$

$$f_c = \frac{m(Tr_1)}{n} + (V + m(1-\sigma)(1-\gamma))(\frac{Tr_2}{n}) \tag{5.8}$$

We present next the background procedure performed in the VB to allow the execution of the duty cycle mechanism, as well as the operations triggered when the MDP chooses as the next state a new state where some nodes are put to sleep.

As it happens in the DA-VBC, the DCDA-VBC is performed at the gateway, which is also responsible for putting the broker nodes in the sleep mode. When there is a change of state, and in the new state the broker nodes perform in a duty cycle mode, the first action is to choose the nodes which will sleep. The number of broker nodes sleeping is determined by the variable γ, and the nodes are chosen according to the residual energy level of each node. The interval of time that a node stays in the sleep mode is defined as the decision epoch t used in the MDP.

In order to be able to determine the broker nodes that need the most to save energy, the gateway receives from the broker nodes messages sent at each interval t with their energy level. Thus, when a new state performs in a duty cycle mode, the gateway executes the two steps below:

- Choose the broker nodes with the lowest energy level to go to sleep mode
- Send a message to the chosen nodes warning them about the change in their operation mode

Once the nodes that will sleep in the next time interval t receive the gateway message, they broadcast a message to update their neighbors about their new operation mode. As the sleep time interval is defined by the duration of a decision epoch, at the beginning of each decision epoch all broker nodes come back to the active operation mode. For this reason, the nodes that were in the sleep mode do not need to send a new message to update their active operation mode.

5.4.3 Performance evaluation

5.4.3.1 Simulation setting

In order to evaluate the performance of DCDA-VBC, we assume the same simulation scenario and configuration used in Chapter 4. We continue using the Network Simulator version 2 (NS-2) to perform our experiments with a simulation time of 300s, and a VB of 8 broker nodes. We also introduce a background traffic in order to change the network condition. The value of γ depends on the σ values, thus the number of broker nodes that operate in the sleep mode is: half of them if σ is good, 1/4 of them if σ is normal, and 0 otherwise. The sleep time as well as each decision epoch continues to be 20s. We simulate DA-VBC and DCDA-VBC (VB without and with duty cycle, respectively) considering the optimal police π^*(represented in the curves by π^{mdp}), the policy π^{Benef} which always chooses the state with the greater packet delivery rate, and the policy π^{Cost} which always chooses the state with the lower cost.

5.4.3.2 Simulation results

We proposed the introduction of a duty cycle mechanism in the broker nodes of our scheme by adding γ as parameter in the state definition of the MDP defined in DA-VBC. Our objective is to verify the behavior of DA-VBC when its broker nodes are able to operate whether in active or sleeping mode. We start evaluating the lifetime metric represented in Figure 5.17. We see that, although having the possibility of putting some broker nodes to sleep, the DCDA-VBC presents the same lifetime than DA-VBC. This happens because even though DCDA-VBC could put until half of the broker nodes to sleep to save a lot of energy, it would also impact in the packet delivery ratio compromising the cost-benefit trade-off. Thus, as DCDA-VBC has as main objective to ensure the optimal cost-benefit trade-off, it only puts broker nodes to sleep when this does not affect its optimality, limiting its gain in energy saving.

One could had wished with this approach to improve the performance of VBC reducing the energy consumption in the broker nodes. However, a decrease in energy consumption with DCDA-VBC would also means the decrease in the packet delivery rate. The most important, here , is the fact that although DCDA-VBC does not improve the network lifetime, it is correctly designed to ensure that the network performs with the optimal cost-benefit trade-off and packet delivery ratio, as can be seen in Figures 5.18 and 5.19. We conclude then that DA-VBC performs well regardless of

whether the nodes in the network are always-on or not. Indeed, DCDA-VBC has as objective to offer a flexible algorithm able to allow the energy savings in the broker nodes when it is worthwhile, but above all, to guarantee an optimal network performance when node's activity scheduling are used in the network.

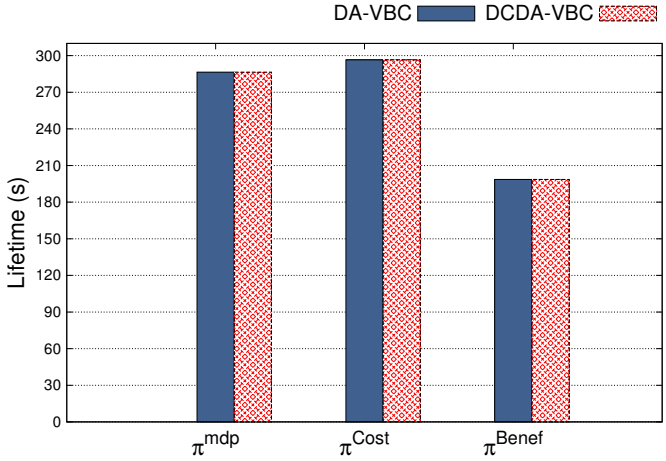


Figure 5.17: lifetime

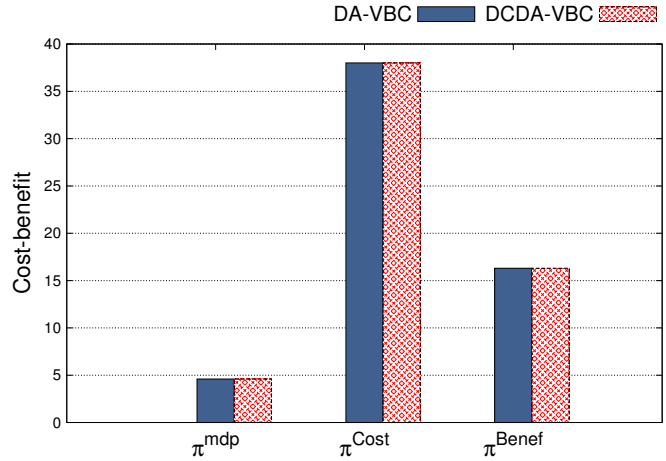


Figure 5.18: Cost benefit

5.5 Conclusion

In this chapter we proposed to study the energy saving duty cycle mechanism in the traditional sensor networks as well as in the virtual broker structure used in the data storage schemes presented over this thesis. In the first part, we studied the possibility to realize autonomous adaptation of the duty cycle mechanism as defined in the IEEE 802.15.4 standard for the traditional WSNs. In this solution the duty cycle is adapted through the dynamic adjustment of the BO and SO values. Our algorithm,

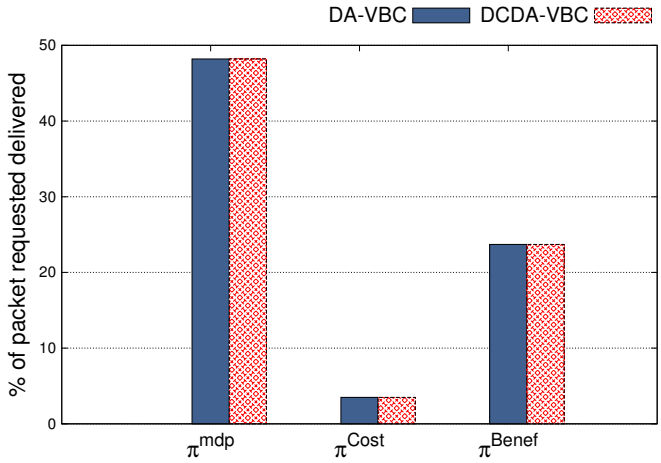


Figure 5.19: Packet delivery

called DBSAA, takes into account the trade-off among the application requirements in terms of bandwidth and the energy consumption in the duty cycle definition. DBSAA does not imply any modification to the 802.15.4 standard and the dynamic adaptation at the coordinator level is performed based on traffic load estimations. Performance evaluations were realized considering the WSN energy consumption as well as two other metrics that determine the application quality: the data delivery ratio and the end-to-end delay. Simulation results showed that our algorithm performs better than the legacy 802.15.4 Std as well as another algorithm from the literature (DSAA) considering different scenarios.

The energy savings obtained by DBSAA showed that the node's activity scheduling can be adapted dynamically to network condition, and as such motivated us to evaluate the impact of a duty cycle mechanism implemented in the VB of our data storage scheme. We started with a theoretical analysis of the duty cycle introduced in the operation mode of the broker nodes. As a result, we claim that to best use the energy resource, through using duty cycle without presenting the drawback of a huge reduction in the packet delivery ratio, is necessary to combine the duty cycle mechanism with the redundancy level adaptation. Thus, we proposed to use the duty cycle as a parameter in the solution presented by DA-VBC. For that, we created the DCDA-VBC version where we change the MDP model used by DA-VBC through adding the duty cycle, represented by γ (variable that defines the number of broker nodes sleeping), as a parameter in the definition of a state in the MDP. The simulation results showed that, although DCDA-VBC does not improve the energy saving in the network, our dynamic and adaptive data storage scheme ensures the optimal cost-benefit in both, always-on and not always-on, WSANs. This means that, although DCDA-VBC looks for saving energy in the broker nodes, it aims primarily to achieve the best performance in the network even if to attain that the energy in the broker nodes are not specially preserved.

Chapter 6

Conclusions

In this thesis, we were interested in solutions to the problem of data management in wireless sensor and actuator networks integrated in the IoT realm through proposing an efficient, cost-effective and reliable in-network data storage scheme. We mean as efficient, a scheme able not only to store the data but also to keep them available and easily retrieved by the data consumers; as cost-effective, a scheme that performs with the best trade-off between its communication cost (energy consumption) and its benefit (packet delivery ratio); and as reliable, a scheme that ensures the delivery of requested packets. To this end, we organized our works into three contributions that evolve over this thesis to solve the data management problem. We started proposing a coding-based data storage scheme that assumes a VB-based Pub/sub architecture as the communication model between data producers and consumers. We intended to provide reliability to our scheme through performing the storage of redundant coded packets whereas reducing its communication cost (which in our work is also perceived as the energy consumption incurred thereof). This is performed while assuming the VB-based pub/sub architecture in the WSN. We realized the performance evaluation of our solution analytically as well as by simulation. Both of them showed that we accomplished a significant increased in delivering the packets requested by the subscribers, reaching to deliver 200% more packets than a data storage scheme that is not based on a coding technique. Moreover, we achieved a better cost-benefit that reached values of at least 42% smaller than our benchmark.

After showing the well worth of using linear network coding in a data storage scheme according to the scheme we proposed, we stepped forward and realized a generalization of our scheme as well as designed a system that allows its integration in a WSAN/IoT scenario. In order to make our solution suitable to this scenario, we proposed DA-VBC, an optimization algorithm based on a MDP formulation that allows our solution to perform with the optimal reliability level. The latter is dynamically and autonomously adapted according to the application requirements and changes in the network conditions. The results obtained through simulations demonstrated a better performance in the packet delivery ratio that increased by 70% compared to VBC, our scheme with a fixed redundancy level, while the cost-benefit trade-off is reduced by 197% and 360% compared to our scheme assuming the minimal and the maximum fixed redundancy levels, respectively.

In our two first contributions, we evaluated the performance of our data storage scheme assuming always-on WSANs. However, thanks to the widespread IEEE 802.15.4 standard for WSNs, one has that many WSNs may perform with a node activity scheduling, which means they perform as a not-always-on WSN. Thus, in our third contribution, we had as goal to go further in the DA-VBC evaluation in order to prove that it works well also when this type of mechanism is adopted in the network. To do so, we started with a study about the duty cycle mechanism in the general case of WSANs. In this first part, we proposed an algorithm that improves the duty cycle mechanism already existent in the IEEE 802.15.4 standard making it self-adaptive. The simulation results were heartening in the means that our algorithm not only saves more energy as it also doubles the packet delivery ratio. Then, we used the same concept of scheduling the nodes activity, but this time only applied in the broker nodes of our data storage scheme. This is realized in order to assess the DA-VBC's performance assuming a not always-on WSAN. To this end, we introduced the duty cycle mechanism in the MDP used to optimize our scheme. The simulation results showed that our modified scheme, DCDA-VBC, kept the same performance achieved in the always-on WSANs making it a good solution to all WSANs regardless the nodes' behavior adopted in the network operation.

To sum up, we can say that throughout our three contributions we reached to answer the questions we posed initially about the data management in the WSANs when they are integrated into the IoT realm. Our first approach was to prove that it is worth using coding-based solution to do in-network data storage in order to ensure data reliability. Then, we generalized and optimized our scheme making it dynamic and self-adaptive while ensuring reliability and optimal cost-benefit trade-off assuming always-on WSANs. And for last, we focused our attention in evaluating the DA-VBC performance also in not-always-on WSANs still confirming its effectiveness.

6.1 Perspectives

The work realized all over this thesis allows the emergence of several interesting thinking. In our two first contributions, we were interested in providing reliability to the in-network stored data at the network level, while ensuring an optimal performance in the data storage by using a Markov Decision Process. We chose a MDP to model our optimization because this technique allows our scheme, through its probability table, to consider the possible future in its decisions. In fact, more than that, it is also able to take into account the impact of its decision in the future condition of the network. However, assuming only one general probability table our scheme does not take advantages of all that MDP can offer. In order to remediate it, we think to determine and add in our MDP specific probability tables to each combination of M and N (the parameters that define the redundancy level). The probability corresponding to each combination could be found empirically through a parameterization study followed by exhaustive simulations resulting in the choice of even more adequate redundancy level adopted in the data storage.

Other prospect, which also concerns the in-network data storage, relates to the query matching problem in the data retrieval. Our scheme ensures data reliability and retrievability at the level of the network layer but does not treat the higher layers

problem thereof. However, in the WSANs/IoT scenario, where the sensors can detect multiple types of attributes that, in their turn, can belong to different events, efficient solutions to identify the data stored and match them to the data queries are very important for the performance of a data storage scheme. The matching between data and query performed at the application level is a complex task that requires, among others, solution to the identification and organization of the data in the virtual broker storage. Hence, extending our data storage scheme to a cross-layer solution in order to approach the problems related to the application layer is a natural step forward for the contributions proposed in this thesis.

Finally, in this thesis, we are interested in providing an efficient way of doing in-network data storage assuming an IoT/WSANs realm. However, an important point of view to be considered in the integration of WSANs in the IoT realm remains to be studied. It consists of taking into account the problem of unauthorized data access. One of our contributions, the IoT/WSAN's integration system, allows the direct access to the data produced in the WSAN by any user around the world. Thus, as the data can be accessed remotely by any user, security measures have to be taken by the network in order to keep the WSNs data confidential to non-authorized users. A standard way to face this problem is to encrypt the information. However, as our scheme presents the facility of already storing coded data, we intend to investigate possibilities of taking advantages of this in order to perform data access control in our integration system. Indeed, in this case the coding strategy would not have as a sole objective to optimize the storage and retrieval but also to secure data access.

Previously Published Material

Journals

Miguel F. de Castro,Levi B. Ribeiro and **Camila H. S. Oliveira**, "An autonomic bio-inspired algorithm for wireless sensor network self-organization and efficiente routing", Elsevier Journal of Network and Computer Applications, vol 35, no. 6, pp. 2003-2015, 2012.

International Conferences (Full Papers)

Camila H. S. Oliveira, Yacine Ghamri-Doudane, Carlos E. F. Brito and Stéphane Lohier "Optimal Network Coding-based In-Network Data Storage and Data Retrieval for IoT/WSNs", IEEE International Symposium on Network Computing and Applications (NCA), Cambridge, MA USA, 2015. (To appear)

Camila H. S. Oliveira, Yacine Ghamri-Doudane, Carlos E. F. Brito and Stéphane Lohier "Virtual Broking Coding for Reliable In-Network Storage on WSANs", IEEE Symposium on Computers and Communications (ISCC), Larnaca, Cyprus, 2015.

Camila H. S. Oliveira and Miguel F. de Castro, "Autonomic Duty Cycling for Target in a Bio-Inspired Wireless Sensor Network", International Conference on Systems (ICONS), Barcelona, Spain, April 2015.

Camila H. S. Oliveira, Yacine Ghamri-Doudane and Stéphane Lohier, "A duty cycle self-adaptation for the 802.15.4 wireless sensor networks", Global Information Infrastructure and Networking Symposium (GIIS), Trento, Italy, Octobre 2013.

Résumé de la thèse en Français

Fiabilité et efficacité énergétique pour le stockage et l'accès aux données dans l'IoT/WSANs : une approche basée sur la théorie du codage

Dans cette thèse, nous étudions la gestion des données dans les réseaux de capteurs sans fil intégrés dans un contexte IoT (Internet of Things). Le manuscrit est organisé de la manière suivante :

Introduction Générale

Nous sommes témoins ces dernières années d'une révolution dans nos interactions avec la technologie. Le développement de périphériques communicants intégrés dans les objets que nous utilisons tous les jours change ces interactions à travers la création d'applications et de services précédemment inconcevables. Les réseaux de capteurs sans fil sont au cœr de cette révolution qui bénéficie de l'évolution des technologies de miniaturisation des équipements basés sur les microsystèmes électromécaniques (MEMS). Ces équipements peuvent être utilisés pour mesurer une grande variété de grandeurs physiques, chimiques ou de biomasse telles la pression, l'intensité lumineuse, le champ magnétique, le pH ; ou encore pour prendre une décision à partir des valeurs mesurées et suivant un ensemble de règles prédéfinies. Toutes ces capacités permettent l'évolution des WSNs (Wireless Sensor Networks) envers les réseaux de capteurs et d'actuateurs sans fil (WSANs), ainsi comme leur utilisation dans différents domaines tels que l'automatisation industrielle, les soins de santé et le bien-être, l'agriculture, les applications militaires, les véhicules automatisés et le transport intelligent, le contrôle de l'environnement et les prévisions, etc.

La variété de scénarios d'application dans lesquels les WSNs peuvent être utilisé et l'évolution vers des dispositifs de captage plus performants font des WSNs un élément primordial dans l'émergence de l'Internet des objets (IoT). Cependant, avec les nouvelles caractéristiques des micro- équipements qui permettent aux nœuds capteur une interaction autonome et directe avec l'environnement, les applications développées dans le contexte IoT sont devenus beaucoup plus complexes que celles prévues pour les WSNs initiaux. Pour pouvoir assurer ces nouveaux services, les WSANs doivent changer leur manière de gérer les données. Ils doivent désormais assurer une communication fiable et à faible coût entre les producteurs et les consommateurs de données pour permettre ces nouveaux services. Ces applications doivent aussi avoir accès aux données les plus récentes générées dans le réseau. Ainsi dans le but de rendre les données rapidement disponibles pour les consommateurs, le stockage au sein du réseau est considéré comme la solution la plus adaptée. Le problème d'un stockage fiable et intégré est donc lié à la nécessité de rendre les données collectés disponibles à tout moment pour les consommateurs et pour les actuateurs. Cependant, malgré les progrès réalisés sur les capteurs, les contrôleurs et les actuateurs, les WSNs restent des réseaux à faibles ressources en ce qui concerne l'énergie, la mémoire et les capacités de traitement. C'est pourquoi, l'un des enjeux majeurs est de concevoir un système de stockage des données robuste et fiable pour ces réseaux contraints.

Stockage de données dans les WSANs

D'après [67], trois différents type de stockage des données existent dans les WSNs suivant la localisation du stockage : local, externe et orienté données (data centric). Dans le stockage local, les nœuds capteurs qui détectent un événement mémorisent la donnée ; dans le stockage externe, les données sont transmises pour être stockées dans une base à l'extérieur du réseau ; dans le stockage orienté données, les données sont mémorisées dans les nœuds les plus proches possibles des nœuds source [67]. Dans ce dernier type de stockage, bien qu'il soit basé sur du stockage local, tous les nœuds capteur peuvent mémoriser des données, que celles-ci aient été captées par eux ou non. Par ailleurs le stockage orienté donnée surpasse en performances les deux autres types en ce qui concerne la consommation d'énergie [6, 49] et la surcharge de trafic occasionnée dans le réseau [7]. Pour ces raisons, dans le contexte IoT, le stockage des données au sein du réseau utilise une approche orientée données.

Le stockage dans le réseau utilisé dans les WSNs intégrés dans un contexte IoT consiste à stocker temporairement les données les plus récentes, tout en les rendant plus facilement récupérables pour les consommateurs [82]. Cette solution, en plus d'améliorer la disponibilité des données, améliore le taux de perte de paquets en utilisant des chemin multi-saut plus courts entre la source et la destination [14]. Cependant, dans le but d'améliorer les performances du stockage au sein du réseau, celui-ci doit pouvoir gérer certaines contraintes. La première est de supporter l'accès aux données par de nombreux utilisateurs de manière fiable et décentralisée. La deuxième est liée aux caractéristiques des WSNs qui présentent par nature une faible capacité énergétique. Les deux problématiques majeures dans ce domaine sont donc :

La fiabilité des données Pour améliorer cette fiabilité, les données sont stockées dans les nœuds répartis. Cependant, ce stockage de données dans des nœuds répartis présente un inconvénient pour la transmission des données vers les consommateurs finaux qui demandent plus de fiabilité et des mécanismes performants de recouvrement de données.

La contrainte en énergie Le besoin de traiter la très grande quantité de données générée dans des réseaux basés sur l'IoT implique une augmentation de la consommation énergétique dans la mesure où davantage de transmissions seront nécessaires pour le stockage et la transmission des données [10]. En effet, d'après [84], la transmission entre les capteurs des WSNs est davantage consommatrice d'énergie que le traitement.

Concernant la problématique énergétique, la meilleure façon de réduire la consommation dans le système de stockage de données est de réduire la surcharge de communication entre les nœuds capteur. Dans ce but, nous retiendrons le modèle Publish/Subscribe qui est le plus adapté aux WSNs dans le contexte IoT. Par ailleurs pour améliorer la fiabilité des données dans le stockage distribué des donnés, la redondance de paquets est introduite dans le réseau en utilisant des techniques de codage.

Publish/Subscribe Paradigme

Un système pub/sub est composé d'équipements autonomes répartis sur tout le réseau dans lequel ils peuvent jouer le rôle de producteurs ou de consommateurs [73]. Dans un tel système, la communication est mise en place par la transmission de trois primitives :

- **Advertise :** primitive envoyée par le producteur pour annoncer le type de données produites ;
- **Publish :** primitive utilisée par le producteur pour envoyer les données à tous les consommateurs intéressés pour les recevoir ;
- **Subscribe :** primitive envoyée par le consommateur pour annoncer sont intérêt à recevoir une information donnée.

Les caractéristiques de ce paradigme le rende particulièrement adapté au WSANs. Cependant, dans le contexte des WSANs, l'introduction du concept de négociation virtuelle (VB - Virtual Broking) dans le but de diminuer le coût en consommation énergétique dans la dissémination des requêtes et des évènements est également étudiée. Le VB est implémenté par un certain nombre de nœuds agent (VBN - Virtual Broker Nodes) qui stockent toutes les données produites dans le réseau, et négocient la communication entre les annonceurs (publishers) et les souscripteurs (subscribers). Dans l'opération basique de ce modèle, les producteurs de données ainsi que les consommateurs envoient respectivement leurs évènements et requêtes aux VBNs. Comme exemple de ce type d'approche, [51] propose une architecture basée sur le VB dans laquelle le réseau est divisé en petites partitions similaires à celle illustrée à la figure 1. Dans chaque partition, les VBNs (qui apparaissent en noir) sont localisés au centre ;

et les nœuds restants, les annonceurs et les souscripteurs, sont repartis sur tout le réseau. Dans le travail présenté dans cette thèse, nous faisons le choix d'un WSAN avec une architecture de type VB constituée de plusieurs partitions, comme celle de la Figure 1, qui fonctionnent suivant le modèle Pub/Sub décrit précédemment.

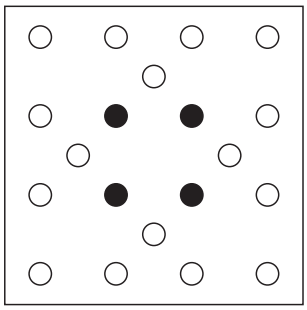


FIGURE 1 – Partition.

Technique de codage

Pour traiter le problème de la perte de paquets dans le stockage reparti de données, les solutions de codage introduisent généralement de la redondance de paquets dans le

réseau. Cette simple solution de redondance consiste à répliquer la même information dans quelques nœuds du réseau. Cependant, cette méthode nécessite trop de mémoire et n'est pas adaptée aux WSNs. En utilisant des techniques de codage, les nœuds introduisent de la redondance de paquets avec une plus faible surcharge que dans le cas la simple réplication. Dans cette thèse, nous tentons de résoudre le problème de la fiabilité dans le stockage des données en utilisant des techniques de codage résumées ci-dessous. Le codage réseau (NC - Network Coding) a été introduit au départ pour améliorer le débit dans les réseaux multicast point-à-point [3]. Il a ensuite été utilisé pour d'autres applications comme la tomographie réseau, l'amélioration de fiabilité, la sécurité [60] et le problème du stockage des données [93, 89, 81]. Le NC replace la traditionnelle procédure store-and-forward mise en place dans les nœuds intermédiaires du réseau par une procédure de store-code-and-forward. Cela signifie que les nœuds intermédiaires relaient et codent les paquets qu'ils reçoivent et que le récepteur final collecte les paquets qui arrivent de différents chemins et décode ceux-ci pour retrouver l'information émise à l'autre bout. Dans cette thèse, nous utilisons le codage réseau linéaire (LNC ' Linear Network Coding) pour la génération des paquets codés. Dans le LNC, les paquets sont considérés come des vecteurs $\mathbf{x} = (x_1, \ldots, x_l)$ d'éléments dans l'ensemble fini $GF(q)$ de taille q qui peut être manipulé et combiné par des opérations arithmétiques définies dans l'ensemble. Supposons que nous avons N packets $\mathbf{x_1}, \ldots, \mathbf{x_N}$ de même taille l, et prenons $c_1, \ldots, c_N$ comme éléments arbitraires de GF(q). Nous pouvons ainsi calculer un paquet codé $\mathbf{y}$ comme la combinaison linéaire des paquets $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

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

Intuitivement, le paquet $\mathbf{y}$ a l'information de tous les paquets $\mathbf{x_i}$ qui sont associés à un coefficient non nul c_i, mais le paquet $\mathbf{y}$ seul n'est pas suffisant pour retrouver tous les paquets originaux$\mathbf{x_1}, \ldots, \mathbf{x_N}$. Considérons ensuite une matrice $\mathbf{C}$ de taille (N, N) d'éléments de GF(q), qui permettent de calculer une collection de N paquets codés comme une combinaison linéaire de $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

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

Supposons maintenant que le coefficient de la matrice $\mathbf{C}$ a le rang N, le système d'équations linéaires

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

avec $\mathbf{Y} = [\mathbf{y_1}, \ldots, \mathbf{y_N}]$ et $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_N}]$, peut être résolu uniquement dans le but de retrouver le paquet $\mathbf{x_1}, \ldots, \mathbf{x_N}$. En pratique, si le coût de traitement est un problème, nous pouvons utiliser une matrice avec peu de coefficients $\mathbf{C}$, dans laquelle la plupart des entrées sont à zéro, ce qui permet d'implémenter le codage et le décodage de manière efficace. Dans ce cas, nous devons faire attention à garantir que la matrice $\mathbf{C}$ a le rang requis N.

Contributions

Système de codage Virtual Broking

L'objectif est de réaliser un stockage et un recouvrement fiable des données dans les réseaux de capteurs et d'actuateurs sans fil (WSANs) utilisant un modèle Pub/Sub. Dans le processus de stockage des données, beaucoup de paquets peuvent êtres perdus en arrivant au nœud de stockage ou sur le chemin entre le nœud de stockage et le nœud consommateur. Pour assurer la fiabilité des données et un mécanisme efficace pour le recouvrement des données, nous proposons le système de stockage avec codage et négociation virtuelle (VBC - Virtual Broking Coding data storage scheme). VBC utilise une structure VB en charge de stocker les paquets codés de données redondants (qui sont générés à travers le codage réseau linéaire ' LNC) et de jouer l'intermédiaire entre les nœuds producteurs et consommateurs. Les performances de VBC sont validées par une analyse théorique et des simulations qui démontrent que VBC offre le bon compromis entre le coût et le bénéfice lié au stockage des paquets codés redondant.

Pour introduire la redondance dans notre solution, VBC doit générer M paquets codés à partir de N paquets de données avec $M >$N. Rappelons que dans LNC, les paquets sont considérés comme des vecteurs $\mathbf{x} = (x_1, \ldots, x_l)$ d'éléments de GF(q), qui peuvent être manipulés et combinés à travers différentes opérations arithmétiques. Donc formellement, si nous considérons une matrice $\mathbf{C}$ de $N \times M$ coefficients, nous pouvons calculer un ensemble de M paquets codés comme une combinaison linéaire de N paquets originaux $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

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

Dans ce contexte, la propriété recherchée du système de codage redondant est que chaque sous-ensemble de paquets codés puisse être utilisé pour retrouver le paquet original. D'un point de vue mathématique, cela nécessite que chaque $N \times N$ sous-matrice $\mathbf{C}$ de la matrice de coefficients $\mathbf{C}$ ait un rang N. En respectant cette condition et avec un modèle Pub/Sub basé une architecture VB, nous implémentons notre solution de codage redondant dans le VB au lieu de l'implémenter dans tous les nœuds, ce qui est le cas des approches de codage traditionnel.

DA-VBC

Dans la suite de notre travail, nous proposons une évolution de VBC vers une solution dynamique et auto-adaptative de stockage de données (DA-VBC). Cette solution offre un niveau de redondance optimal par rapport au comportement courant du réseau tout en restant intégrée dans les scénarios IoT.

Il s'agit de la contribution majeure de cette thèse. Après avoir démontré qu'il était bénéfique d'utiliser un codage réseau linéaire pour assurer la fiabilité des données avec un modèle Pub/Sub fondé sur une architecture VB, notre objectif est d'aller pus loin en faisant de notre système de stockage des données une solution adaptée pour les services et applications IoT. Dans VBC, nous considérons un niveau de redondance fixe dans la mesure où nous cherchons seulement à prouver les avantages de notre méthode.

Cependant, dans un contexte IoT, nous sommes dans un environnement décentralisé et dynamique qui sert de base à l'implémentation d'une variété de services qui requièrent différents besoins. Pour tenir compte de ce dernier point, nous commençons par concevoir un système qui définit un modèle de communication qui intègre notre solution dans un scénario WSANs/IoT (Figure 2), ainsi qu'un mécanisme de cache dynamique et auto-ajustable dans le réseau nommé Dynamic Adaptive Virtual Broking Coding (DA-VBC) qui assure un stockage et un recouvrement fiable des données prenant en compte tout changement dans le réseau.



Figure 2 – Modèle de communication : un exemple

Nous définissons un système qui réalise le stockage local des données tout en permettant le recouvrement local et global des ces données. Ce système permet l'accès à la fois aux données locales et globales à travers l'interaction entre les quatre entités composant le système qui sont : le producteur, le nœud agent (broker node), la passerelle IoT et le consommateur. Les consommateurs sont les capteurs qui réalisent le contrôle de l'environnement et génèrent les données devant être mémorisées. Les consommateurs sont les équipements susceptibles de recevoir les données ; ils peuvent être de deux type : actuateurs ou périphériques mobiles comme les téléphones cellulaires, les tablettes, les PC portables et tout autre périphérique connecté. Les nœuds agent sont les équipements responsables du stockage des données produites. Les passerelles sont des équipements plus robustes servant d'intermédiaire dans la communication entre le VB et le *cloud*. Un exemple de système VBC est décrit à la Figure reffig :exapp.

L'algorithme VBC réalise l'adaptation dynamique du niveau de redondance, et également le choix du niveau optimal de redondance en considérant l'état actuel du réseau. Dans ce but, nous généralisons d'un point de vue théorique l'expression pro-

posée dans la première contribution. Ensuite nous modélisons le problème du choix du niveau optimal de redondance, c'est à dire le niveau de redondance qui permet au réseau le meilleur rapport coût-bénéfice, comme un problème de processus de décision Markovien (MPD - Markov Decision Process). Rappelons que la réduction de coût optimale attendue dans un MDP satisfait la relation de récurrence suivante, appelée équation de Bellman :

$$v(s) \;=\; \min_{a\in\mathcal{A}} \Big\{ r(s,a) + \sum_{j\in S} \lambda p(j|s,a) v(j) \Big\} \tag{1}$$

Dans la partie gauche de l'équation (1), le premier terme représente la réduction obtenue à l'instant de la décision courante, lorsque l'action a est effectuée, alors que le second terme représente la future réduction de coût attendue lorsque l'action a est effectuée. La solution de l'équation (1) ne donne pas seulement la réduction minimum de coût attendue $v(s)$, mais permet également de trouver une stratégie optimale π^* pour obtenir la réduction optimale $v(s)$.

L'algorithme DA-VBC est déployé sur la passerelle responsable de la détection des performances du réseau. Suivant le comportement observé (σ), par exemple l'augmentation ou la diminution des pertes de paquets, la passerelle diffuse aux VBNs le niveau optimal de redondance correspondant au nouvel état. Pour cela, DA-VBC opère en deux phases : la phase de calcul préalable et la phase opérationnelle. Dans la première phase, la passerelle trouve la stratégie déterministe optimale π^* basée sur l'algorithme d'itération de valeur et l'enregistre dans la table de consultation. Dans la phase opérationnelle, la passerelle choisi l'action $\pi^*(s) = (m, n)$ (le niveau de redondance) basée sur l'état courant et la stratégie déterministe optimale π^*.

Prise en compte de l'influence du cycle d'activité des capteurs (Duty Cycle) sur notre solution

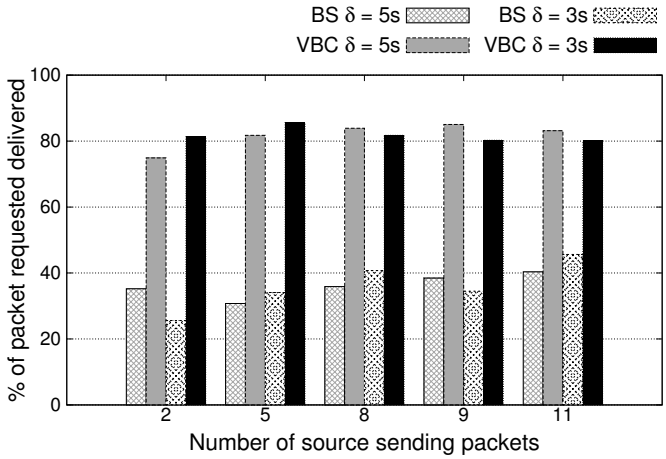
Bien que la contrainte d'énergie soit prise indirectement en compte par DA-VBC (dans la mesure où nous considérons la surcharge des communications comme un paramètre qui influe sur la consommation énergétique de notre solution), celle-ci a été évaluée sans la mise en place des mécanismes standards d'économie d'énergie dans le réseaux, à savoir les cycles d'activités des capteurs impliquant l'endormissement et le réveil de ces derniers. C'est pourquoi nous proposons, dans notre dernière contribution, une étude, réalisée en deux parties, sur les réseau de capteurs utilisant un ordonnancement de l'activité des nœuds et limpact que cela peut avoir sur notre solution. Les mécanismes d'ordonnancement de l'activité des nœuds (duty cycle mechanisms), ont pour objectif de permettre une meilleure distribution de la consommation énergétique, ce qui entraine une augmentation de la durée de vie du réseau. Par contre, cela peut détériorer les performances des communications si l'ordonnancement de l'activité des n oeuds n'est pas optimisé par rapport à l'activité du réseau. Ainsi, dans la première partie de cette étude, nous adaptons le mécanisme de cycle d'activité présent dans le standard 802.15.4 en proposant un algorithme d'intervalle dynamique entre balises et d'adaptation de la supertrame (DBSAA - Dynamic Beacon interval and Superframe Adaptation Algorithm). Dans la seconde partie, nous introduisons le cycle d'activité dans le MDP définit dans notre deuxième contribution dans le but d'évaluer son impact sur le compromis obtenu par DA-VBC.

Résultats

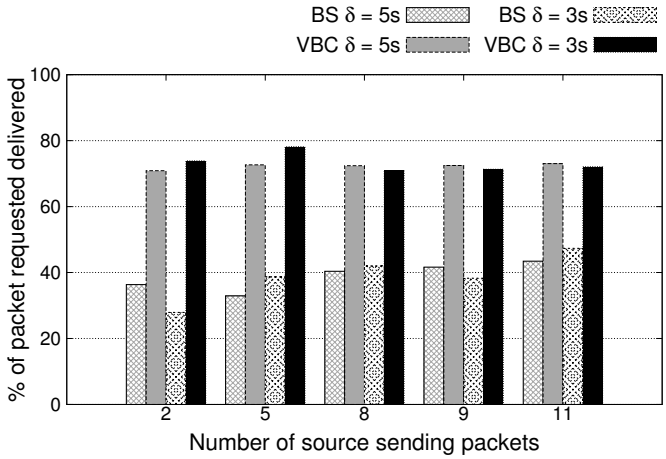
Dans cette thèse nous nous sommes intéressés aux solutions au problème de la gestion des données dans les réseaux de capteurs sans fil intégrés dans un scénario IoT en proposant des solutions efficaces et performantes pour le cout et la fiabilité du stockage des données au sein du réseau. L'efficacité concerne non seulement le système de stockage des données mais aussi la disponibilité et la facilité de recouvrement pour les consommateurs de ces données ; concernant le coût, la solution proposée offre le meilleure compromis entre la consommation en énergie et le taux de paquets délivrés ; concernant la fiabilité, la solution assure la livraison des paquets demandés. Dans ce but, nous avons organisé notre travail en trois contributions qui évoluent dans cette thèse pour résoudre le problème de la gestion des données. Nous avons commencé par proposer un système de stockage des données basé sur le codage qui utilise une architecture Pub/Sub basée sur le VB comme modèle de communication entre les producteurs et les consommateurs de données. Nous tentons de fournir la fiabilité à notre solution en réalisant le stockage des paquets codés redondants tout réduisant le coût de communication (ce qui dans notre travail est aussi considéré comme un élément de la réduction de la consommation). L'évaluation de performances de notre solution et réalisée analytiquement et par simulation. Les deux montrent une augmentation significative dans la distribution des paquets demandés par les abonnées : 200% de paquets délivrés en plus que dans le cas d'un stockage non basé sur une technique de codage (Figure 3). De plus, nous obtenons un meilleur rapport coût-bénéfice avec une diminution d'au moins 42% (Figure 4).

Après avoir démontré qu'utiliser le codage réseau linéaire dans le stockage des données, de la manière proposée, présente un grand intérêt, nous avons été plus loin en généralisant notre solution et en concevant un système permettant son intégration dans un scénario WSAN/IoT. Pour adapter notre solution à ce scénario, nous avons proposé un algorithme d'optimisation utilisant la méthode MDP qui permet d'obtenir le niveau optimal de fiabilité. Ce dernier est dynamiquement et automatiquement adapté suivant les besoins de l'application et les changements d'état du réseau. Les résultats obtenus par simulation montrent une amélioration dans le taux de distribution de paquets qui augmente de 70% par rapport à notre solution avec un niveau de redondance fixe (Figure 6). Parallèlement, le rapport coût-bénéfice est réduit de 197% et 360% par rapport à notre solution avec les niveaux fixes de redondance minimal et maximal, respectivement (Figure 5).

Dans notre troisième contribution, nous avions comme objectif d'étudier l'impact du cycle d'activité des capteurs (Duty Cycle) sur le rapport coût/bénéfice de notre solution de stockage des données. Dans ce but, nous avons commencé par une étude sur le cycle d'activité existant déjà dans le standard 802.15.4. Dans cette première partie, nous avons proposé un algorithme pour améliorer ce mécanisme et le rendre auto-adaptatif à la charge du réseau. Les résultats de simulation sont encourageants dans la mesure où non seulement notre algorithme permet des gains d ?énergie au niveau MAC mais aussi double le taux de distribution des paquets (Figures 7, 8, 9, 10, 11, 12, 13, 14). Nous avons donc utilisé le même concept d'ordonnancement de l'activité des nœuds mais cette fois seulement dans les nœuds agent de notre système de stockage des données. Pour cela, nous avons introduit le mécanisme de cycle d'activité dans le



(a) Taux de paquets délivrés (EPCC)

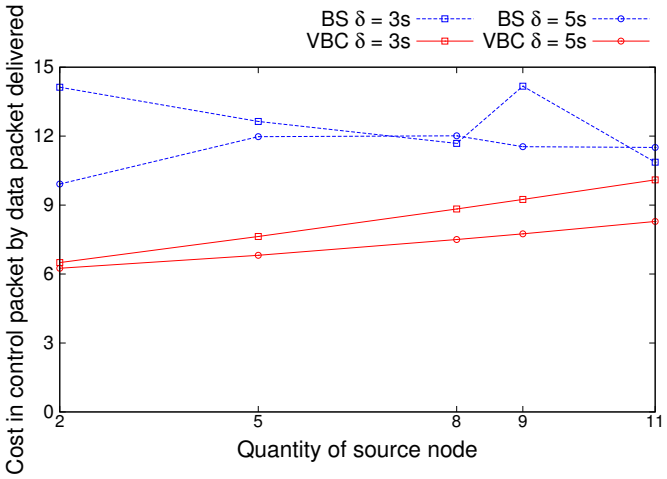


(b) Taux de paquets délivrés (FPVB)

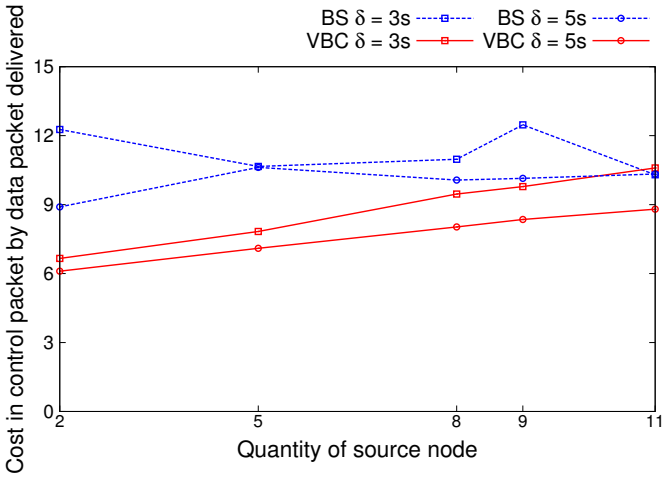
FIGURE 3 – Taux de paquets délivrés.

MDP définit dans notre deuxième contribution et utilisé pour optimiser notre solution. Les résultats de simulation montrent que le fait d'avoir des cycles d'endormissement et de réveil de capteurs n'a aucune influence sur les performances de notre solution. En effet, performances obtenues ici sont similaires en tout point à celles obtenues précédemment. En effet, cela montre qu'il n'est pas possible d'aller plus loin dans la sauvegarde de la consommation énergétique sans avoir une influence négative sur le taux de recouvrement de paquets. Ainsi, ce fait corrobore nos résultats précédents en montrant qu'en effet, notre solution permet le meilleur rapport cout/bénéfice possible.

A travers nos trois contributions, nous parvenons à résoudre la question de la gestion des données dans les WSNs lorsqu'ils font partie d'un environnement IoT. Notre première intention était de prouver qu'il est bénéfique d'utiliser solution basée sur le codage réseau pour réaliser le stockage des données au sein le réseau dans le but d'assurer le fiabilité des données. Ensuite, nous avons généralisé et optimisé notre solution en la rendant dynamique et auto-adaptative tout en assurant la fiabilité et un



(a) Coût-bénéfice (EPCC)



(b) Coût-bénéfice (FPVB)

Figure 4 – coût-bénéfice

rapport coût/bénéfice optimal. Enfin, nous nous sommes concentré sur la sauvegarde de l'énergie des nœuds à la fois dans les nœuds standard et dans les nœuds agent.

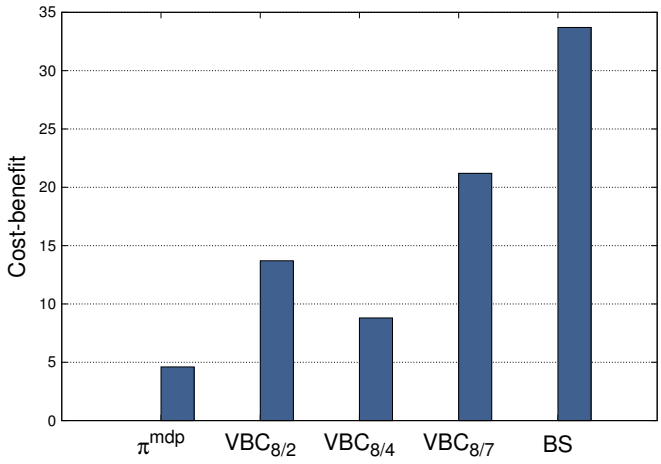


FIGURE 5 – Coût-bénéfice

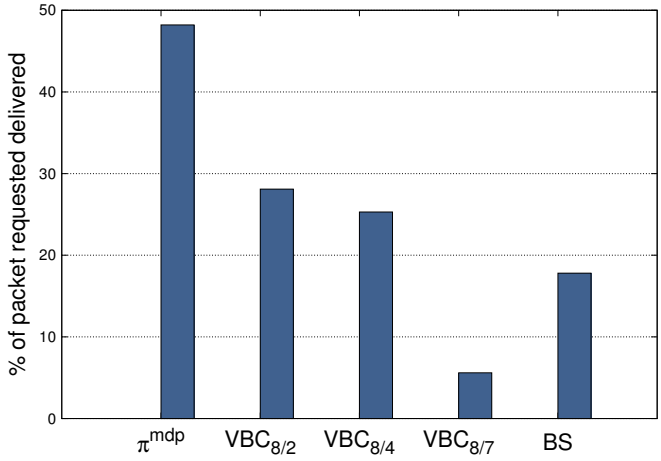


FIGURE 6 – Taux de paquets délivrés.

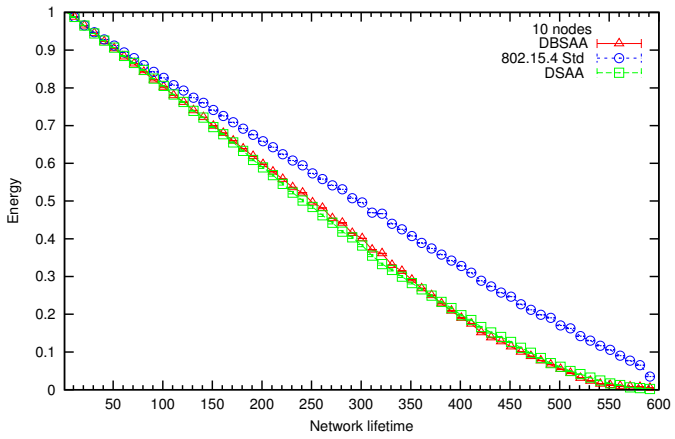


FIGURE 7 – Energy average consumption DC 50% (BO = 2, SO = 1) - 10 nodes

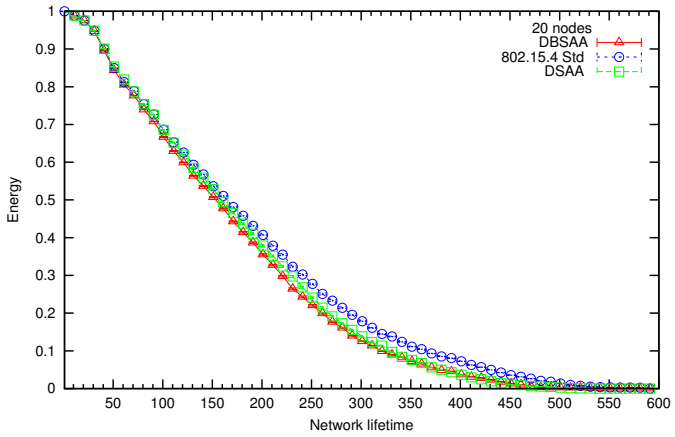


Figure 8 – Energy average consumption DC 50% (BO = 2, SO = 1)- 20 nodes

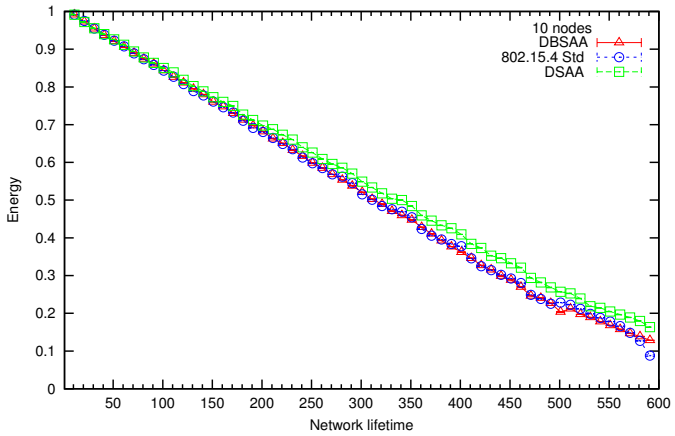


Figure 9 – Energy average consumption DC 6% (BO = 6, SO = 2) - 10 nodes

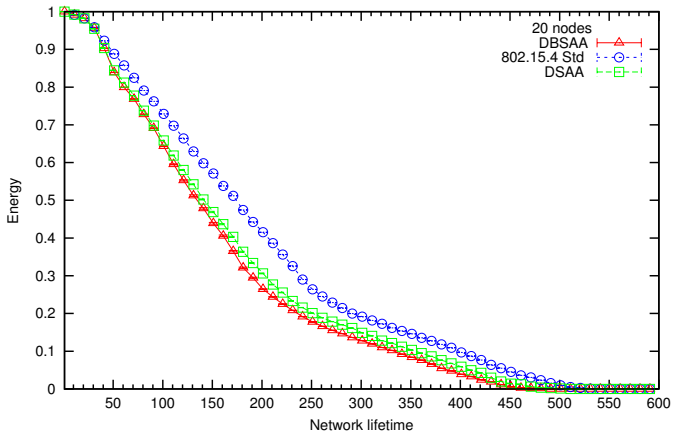


Figure 10 – Energy average consumption DC 6% (BO = 6, SO = 2) - 20 nodes

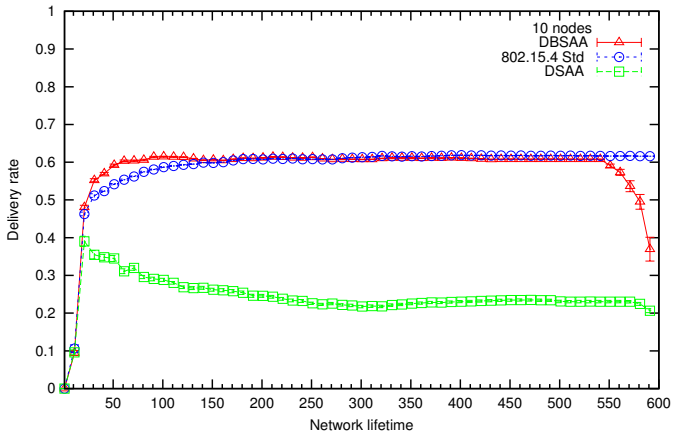


FIGURE 11 – Packet delivery ratio DC 50% (BO = 2, SO = 1) - 10 nodes

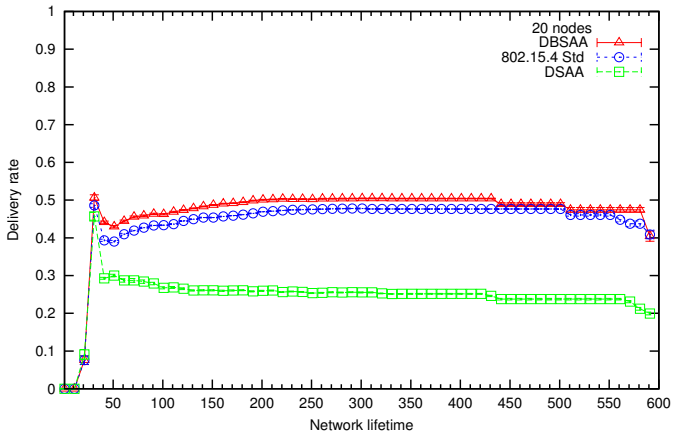


FIGURE 12 – Packet delivery ratio DC 50% (BO = 2, SO = 1) - 20 nodes

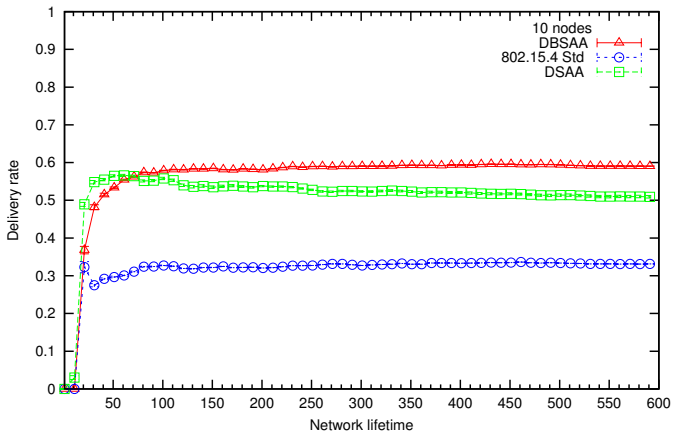


FIGURE 13 – Packet delivery ratio DC 6% (BO = 6, SO = 2) - 10 nodes

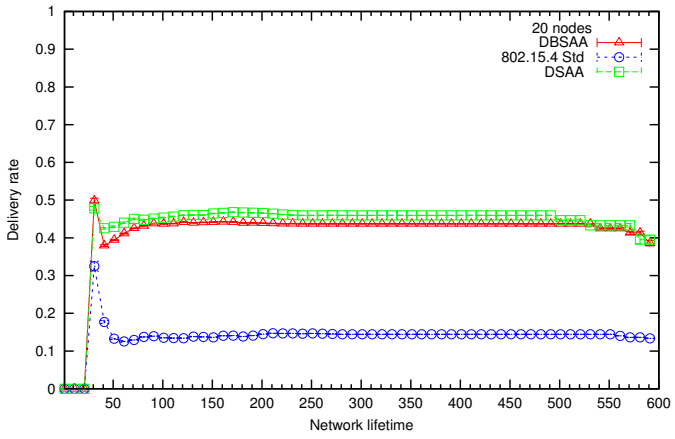


Figure 14 – Packet delivery ratio DC 6% (BO = 6, SO = 2) 20 - nodes

List of Figures

2.1 Traditional Wireless Sensor Network. 18
2.2 Ambient energy harvesting devices. 18
2.3 topology 19
2.4 Super frame structure. 20
2.5 WSNs architectures. 22
2.6 Network partition. 26
2.7 Example of Network Coding Operation. 27

3.1 Publish operation 36
3.2 Subscribe operation 36
3.3 Packet delivery probability. 39
3.4 Cost per packet delivered. 43
3.5 Packet delivery ratio. 45
3.6 Total Communication Cost. 47
3.7 Cost-benefit 48
3.8 Relative load balance between VBNs 2 data sources 49
3.9 Relative load balance between VBNs 11 data sources 50
3.10 Cost-benefit 51

4.1 System model: an example 57
4.2 System model: iteration among entities 58
4.3 Sequence diagram of the publish operation 58
4.4 Sequence diagram of the subscribe operation 60
4.5 Cost-benefit with virtual broker of size 6 62
4.6 Scenario of simulation. 66
4.7 Cost benefit 67
4.8 Packet delivery 67
4.9 lifetime 68
4.10 Cost benefit 69
4.11 Packet delivery 69
4.12 lifetime 70

5.1 Packet delivery ratio DC 50% (BO = 2, SO = 1) 10 nodes 80
5.2 Packet delivery ratio DC 50% (BO = 2, SO = 1)20 nodes 81
5.3 Packet delivery ratio DC 6% (BO = 6, SO = 2) 10 nodes 81
5.4 Packet delivery ratio DC 6% (BO = 6, SO = 2) 20 nodes 81

5.5 Energy average consumption DC 50% (BO = 2, SO = 1).10 nodes . . . 82
5.6 Energy average consumption DC 50% (BO = 2, SO = 1). 20 nodes . . . 82
5.7 Energy average consumption DC 6% (BO = 6, SO = 2) 10 nodes 83
5.8 Energy average consumption DC 6% (BO = 6, SO = 2) 20 nodes 83
5.9 Delay DC=6% (BO=2,SO=1)] 83
5.10 Delay DC=50% (BO=6,SO=2) 84
5.11 Cost without DC 86
5.12 Packet delivery without DC 86
5.13 Cost with $\gamma = 0.25$ 86
5.14 Packet delivery $\gamma = 0.25$ 86
5.15 Cost with $\gamma = 0.33$ 86
5.16 Packet delivery with $\gamma = 0.33$ 86
5.17 lifetime 89
5.18 Cost benefit 89
5.19 Packet delivery 90

7 Energy average consumption DC 50% (BO = 2, SO = 1) - 10 nodes . . 110
8 Energy average consumption DC 50% (BO = 2, SO = 1)- 20 nodes . . . 111
9 Energy average consumption DC 6% (BO = 6, SO = 2) - 10 nodes . . . 111
10 Energy average consumption DC 6% (BO = 6, SO = 2) - 20 nodes . . . 111
11 Packet delivery ratio DC 50% (BO = 2, SO = 1) - 10 nodes 112
12 Packet delivery ratio DC 50% (BO = 2, SO = 1) - 20 nodes 112
13 Packet delivery ratio DC 6% (BO = 6, SO = 2) - 10 nodes 112
14 Packet delivery ratio DC 6% (BO = 6, SO = 2) 20 - nodes 113

List of Tables

3.1 Parameters of the cost-benefit estimation. 43

5.1 β values. 77

Bibliography

[1] Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans) (amendment of ieee std 802.15.4-2003), 2006.

[2] M.B. Abdullahi and Guojun 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)*, pages 297–299, May 2012.

[3] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Trans. on Infor. Theory*, 46:1204–1216, Jul 2000.

[4] Ian F. Akyildiz and Ismail H. Kasimoglu. Wireless sensor and actorâ networks: research challenges. *Ad Hoc Networks*, 2(4):351 – 367, 2004.

[5] A.E. Al-Fagih, F.M. Al-Turjman, and H.S. Hassanein. Ubiquitous robust data delivery for integrated rsns in iot. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 280–285, Dec 2012.

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

[7] M. Albano, S. Chessa, F. Nidito, and S. Pelagatti. Dealing with nonuniformity in data centric storage for wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(8):1398–1406, Aug 2011.

[8] Michele Albano and Stefano Chessa. Publish/subscribe in wireless sensor networks based on data centric storage. In *Proceedings of the 1st International Workshop on Context-Aware Middleware and Services: affiliated with the 4th International Conference on Communication System Software and Middleware (COMSWARE 2009)*, pages 37–42. ACM, June 2009.

[9] Cristina Alcaraz, Pablo Najera, Javier Lopez, and Rodrigo Roman. Wireless sensor networks and the internet of things: Do we need a complete integration?

[10] N.A. Ali and M. Abu-Elkheir. Data management for the internet of things: Green directions. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 386–390, Dec 2012.

[11] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. Multi-source data retrieval in iot via named data networking. In *Proceedings of the 1st International Conference on Information-centric Networking*, ICN '14, pages 67–76, New York, NY, USA, 2014. ACM.

[12] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Vincenzo Neri. Reliability and energy efficiency in multi-hop ieee 802.15.4/zigbee wireless sensor networks. In *IEEE Symposium on Computers and Communications (ISCC)*, pages 336–341, June 2010.

[13] B Ando, S Baglio, C Trigona, N Dumas, L Latorre, and P Nouet. Nonlinear mechanism in mems devices for energy harvesting applications. *Journal of Micromechanics and Microengineering*, 20(12):125020, 2010.

[14] Emmanuel Baccelli, Christian Mehlis, Oliver Hahm, Thomas C. Schmidt, and Matthias Wählisch. Information centric networking in the iot: Experiments with ndn in the wild. In *Proceedings of the 1st International Conference on Information-centric Networking*, ICN '14, pages 77–86, New York, NY, USA, 2014. ACM.

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

[16] Angelo Brayner, Aretusa Lopes, Diorgens Meira, Ricardo Vasconcelos, and Ronaldo Menezes. Toward adaptive query processing in wireless sensor networks. *Signal Processing*, 87(12):2911 – 2933, 2007.

[17] Hakan Cam, OzgurKoray Sahingoz, and AhmetCoskun Sonmez. Wireless sensor networks based on publish/subscribe messaging paradigms. In *Advances in Grid and Pervasive Computing*, pages 233–242. 2011.

[18] Zuhal Can and Murat Demirbas. A survey on in-network querying and tracking services for wireless sensor networks. *Ad Hoc Networks*, 11(1):596 – 610, 2013.

[19] Ning Cao, Qian Wang, Kui Ren, and Wenjing Lou. Distributed storage coding for flexible and efficient data dissemination and retrieval in wireless sensor networks. In *IEEE Intl Conf on Comm*, pages 1–5, May 2010.

[20] Cyril Cecchinel, Matthieu Jimenez, SÃ©bastien Mosser, and Michel 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*, SERVICES '14, pages 442–449, Washington, DC, USA, 2014. IEEE Computer Society.

[21] S. Chalasani and J.M. Conrad. A survey of energy harvesting sources for embedded systems. In *Southeastcon, 2008. IEEE*, pages 442–447, April 2008.

[22] 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)*, pages 71–74, Jan 2012.

[23] Claude Chaudet, Isabelle Demeure, Salma Ktari, Nicola Costagliola, and Samuel Tardieu. Publish/subscribe for wireless sensor networks. In *Proceedings of the 7th Asian Internet Engineering Conference*, AINTEC '11, pages 19–21, New York, NY, USA, 2011. ACM.

[24] S. Chessa and P. Maestrini. Robust distributed storage of residue encoded data. *Information Theory, IEEE Transactions on*, 58(12):7280–7294, Dec 2012.

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

[26] Rodolfo de Paz Alberola and Dirk Pesch. Duty cycle learning algorithm (dcla) for ieee 802.15.4 beacon-enabled wireless sensor networks. *Ad Hoc Networks*, 10(4):664–679, June 2012.

[27] Mohamed Diallo, Vasilis Sourlas, Paris Flegkas, Serge Fdida, and Leandros Tassiulas. A content-based publish/subscribe framework for large-scale content delivery. *Computer Networks*, 57(4):924 – 943, 2013.

[28] A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.

[29] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran. Decentralized erasure codes for distributed networked storage. *IEEE Transactions on Information Theory*, 52(6):2809–2816, June 2006.

[30] AlexandrosG. Dimakis and Kannan Ramchandran. Network coding for distributed storage in wireless networks. In Venkatesh Saligrama, editor, *Networked Sensing Information and Control*, pages 115–134. Springer US, 2008.

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

[32] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, jun 2003.

[33] A. Fujimura, S.Y. Oh, and M. Gerla. Network coding vs. erasure coding: Reliable multicast in ad hoc networks. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–7, Nov 2008.

[34] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.*, 29(7):1645–1660, September 2013.

[35] Haibo Hu, Jianliang Xu, and Xueyan Tang. Energy efficient data management. In NitaigourP. Mahalik, editor, *Sensor Networks and Configuration*, pages 259–279. Springer Berlin Heidelberg, 2007.

[36] He Huang, E.C. Ngai, and Jiangchuan Liu. A location-based publish/subscribe framework for wireless sensors and mobile phones. In *Wireless Comm. and Networking Conf. (WCNC), 2012 IEEE*, pages 2173–2178, April 2012.

[37] 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)*, pages 1509–1514, Sept 2012.

[38] Raj. Jain. *The Art of Computer Systems Performance Analysis : Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, New York, 1991.

[39] Lihong Jiang, Li Da Xu, Hongming Cai, Zuhai Jiang, Fenglin Bu, and Boyi Xu. An iot-oriented data storage framework in cloud computing platform. *Industrial Informatics, IEEE Transactions on*, 10(2):1443–1451, May 2014.

[40] Min-Sook Jin, Hosung Park, Euisin Lee, Soochang Park, and Sang-Ha Kim. A data dissemination model base on content-based publish/subscribe paradigm in large-scale wireless sensor networks. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6, April 2009.

[41] Abhinav Kamra, Vishal Misra, Jon Feldman, and Dan Rubenstein. Growth codes: Maximizing sensor network data persistence. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 255–266, New York, NY, USA, 2006. ACM.

[42] M. Khanafer, M. Guennoun, and H.T. Mouftah. Adaptive sleeping periods in ieee 802.15.4 for efficient energy savings: Markov-based theoretical analysis. In *IEEE International Conference on Communications (ICC)*, pages 1–6, June 2011.

[43] Zhenning Kong, S.A. Aly, and E. Soljanin. Decentralized coding algorithms for distributed storage in wireless sensor networks. *IEEE Journal on Selected Areas in Comm*, 28(2):261–267, Feb 2010.

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

[45] Bih-Hwang Lee and Huai-Kuei Wu. Study on a dynamic superframe adjustment algorithm for ieee 802.15.4 lr-wpan. In *IEEE Vehicular Technology Conference (VTC)*, pages 1–5, May 2010.

[46] Dongho Lee and Kwangsue Chung. Adaptive duty-cycle based congestion control for home automation networks. *IEEE Transactions on Consumer Electronics*, 56(1):42–47, February 2010.

[47] Jinbao Li, Shouling Ji, and Jinghua Zhu. Data caching based queries in multi-sink sensor networks. In *Mobile Ad-hoc and Sensor Networks, 2009. MSN '09. 5th International Conference on*, pages 9–16, Dec 2009.

[48] Wang Li-hong, Teng Hai-kun, and Yu Guang hua. Sensors access scheme design based on internet of things gateways. In *Intelligent Systems Design and Engineering Applications (ISDEA), 2014 Fifth International Conference on*, pages 901–904, June 2014.

[49] W.-H. Liao and C.-C. Chen. Data storage and range query mechanism for multi-dimensional attributes in wireless sensor networks. *Communications, IET*, 4(15):1799–1808, Oct 2010.

[50] Jae-Han Lim and Byung Tae Jang. Dynamic duty cycle adaptation to real-time data in ieee 802.15.4 based wsn. In *IEEE Consumer Communications and Networking Conference*, pages 353–357, January 2008.

[51] Yang Liu, Boon-Chong Seet, and Al-Anbuky A. Virtual brokers for large-scale publish/subscribe in wireless sensor networks. In *IEEE/IFIP Intl Conf on Embedded and Ubiquitous Comp*, pages 240–246, Dec 2010.

[52] Yang Liu, Boon-Chong 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*, pages 2553–2558, Sept 2012.

[53] Ying Liu, Beth Plale, et al. Survey of publish subscribe event systems. *Computer Science Dept, Indian University*, 16, 2003.

[54] Yu Liu, Jianwei Niu, Lianjun Yang, and Lei Shu. ebplatform: An iot-based system for ncd patients homecare in china. In *Global Communications Conference (GLOBECOM), 2014 IEEE*, pages 2448–2453, Dec 2014.

[55] Meng Ma, Ping Wang, and Chao-Hsien Chu. Data management for internet of things: Challenges, approaches and opportunities. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 1144–1151, Aug 2013.

[56] L. Mainetti, L. Patrono, and A. Vilei. Evolution of wireless sensor networks towards the internet of things: A survey. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pages 1–6, Sept 2011.

[57] Shaobo Mao, Man Hon Cheung, and V.W.S. Wong. Joint energy allocation for sensing and transmission in rechargeable wireless sensor networks. *Vehicular Technology, IEEE Transactions on*, 63(6):2862–2875, July 2014.

[58] A. Margara and G. Cugola. High-performance publish-subscribe matching using parallel hardware. *Parallel and Distributed Systems, IEEE Transactions on*, 25(1):126–135, Jan 2014.

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

[60] Takahiro MATSUDA, Taku NOGUCHI, and Tetsuya TAKINE. Survey of network coding and its applications. *IEICE Transactions on Communications*, E94.B(3):698–717, 2011.

[61] M.V. Moreno, J. Santa, M.A. Zamora, and A.F. Skarmeta. A holistic iot-based management platform for smart environments. In *Communications (ICC), 2014 IEEE International Conference on*, pages 3823–3828, June 2014.

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

[63] Choon-Sung Nam, Hee-Jin Jeong, and Dong-Ryeol Shin. Design of wireless sensor networks middleware using the publish/subscribe paradigm. In *Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on*, volume 1, pages 559–563, Oct 2008.

[64] M. Neugebauer, J. Plonnigs, and K. Kabitzsch. A new beacon order adaptation algorithm for ieee 802.15.4 networks. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, pages 302–311, January 2005.

[65] N. Nguyen, S. Krishnamurthy, Peng Xie, and D. Jones. Improving information availability in storage-centric sensor networks. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pages 82–90, Ocotober 2008.

[66] 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.

[67] Kyungseo Park and R. Elmasri. Query classification and storage evaluation in wireless sensor networks. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 35–35, 2006.

[68] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, Feb 1999.

[69] Cuong Pham and D.A. Tran. A search-efficient architecture for data-centric sensor networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6, Dec 2011.

[70] Shashank Priya. Advances in energy harvesting using low profile piezoelectric transducers. *Journal of Electroceramics*, 19(1):167–184, 2007.

[71] 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)*, pages 199–208, Sept 2013.

[72] 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)*, pages 199–208, Sept 2013.

[73] Haiying Shen. Content-based publish/subscribe systems. In Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon, editors, *Handbook of Peer-to-Peer Networking*, pages 1333–1366. Springer US, 2010.

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

[75] Yunlei Sun, XiuQuan Qiao, Bo Cheng, and Junliang Chen. A low-delay, lightweight publish/subscribe architecture for delay-sensitive iot services. In *20th IEEE Intl Conf on Web Services (ICWS)*, pages 179–186, June 2013.

[76] Lizzie Tang and Chris Guy. Radio frequency energy harvesting in wireless sensor networks. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, IWCMC '09, pages 644–648, New York, NY, USA, 2009. ACM.

[77] M.A. Tariq, B. Koldehofe, and K. Rothermel. Securing broker-less publish/subscribe systems using identity-based encryption. *IEEE Trans on Paral and Distributed Sys*, 25(2):518–528, Feb 2014.

[78] Duc A Tran and Linh H Truong. Enabling publish/subscribe services in sensor networks. *University of Massachusetts, Boston and IBM Zurich Research Laboratory, Switzerland*, 2010.

[79] Y.B.D. Trinugroho, M. Gerdes, M.M.M. Amjad, F. Reichert, and R. Fensli. A rest-based publish/subscribe platform to support things-to-services communications. In *Asia-Pacific Conf on Comm*, page 321, Aug 2013.

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

[81] D. Vukobratovic, C. Stefanovic, and V. Stankovic. Fireworks: A random linear coding scheme for distributed storage in wireless sensor networks. In *Information Theory Workshop (ITW), 2010 IEEE*, pages 1–5, Aug 2010.

[82] S. Vural, P. Navaratnam, Ning Wang, Chonggang Wang, Lijun Dong, and R. Tafazolli. In-network caching of internet-of-things data. In *Communications (ICC), 2014 IEEE International Conference on*, pages 3185–3190, June 2014.

[83] Dan Wang, Qian Zhang, and Jiangchuan Liu. Partial network coding: Theory and application for continuous sensor data collection. In *IEEE Inter. Workshop on Quality of Service (IWQoS)*, pages 93–101, June 2006.

[84] Jun Wang, Ling Feng, Wenwei Xue, and Zhanjiang Song. A survey on energy-efficient data management. *SIGMOD Rec.*, 40(2):17–23, September 2011.

[85] Ning Wang and Jiaru Lin. Network coding for distributed data storage and continuous collection in wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1–4, Oct 2008.

[86] Qian Wang, Kui Ren, Wenjing Lou, and Yanchao Zhang. Dependable and secure sensor data storage with dynamic integrity assurance. In *INFOCOM 2009, IEEE*, pages 954–962, April 2009.

[87] Wei 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*, pages 1466–1471, Sept 2012.

[88] Dan Xiao, Yang Zhang, and Junliang Chen. A real-time and reliable forwarding scheme of publish/subscribe system. In *Intl Conf on Measuring Technology and Mechatronics Automation*, pages 51–54, Jan 2014.

[89] Xianjun Yang, Xiaofeng Tao, E. Dutkiewicz, Xiaojing Huang, Y.J. Guo, and Qimei Cui. Energy-efficient distributed data storage for wireless sensor networks based on compressed sensing and network coding. *IEEE Transactions on Wireless Communications*, 12(10):5087–5099, October 2013.

[90] Yang Yang, Miao Jin, Yao Zhao, and Hongyi Wu. Cut graph based information storage and retrieval in 3d sensor networks with general topology. In *INFOCOM, 2013 Proceedings IEEE*, pages 465–469, April 2013.

[91] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *Internet of Things Journal, IEEE*, 1(1):22–32, Feb 2014.

[92] I.P. 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*, pages 1–5, June 2014.

[93] Rongfei Zeng, Yixin Jiang, Chuang Lin, Yanfei Fan, and Xuemin Shen. A distributed fault/intrusion-tolerant sensor data storage scheme based on network coding and homomorphic fingerprinting. *Parallel and Distributed Systems, IEEE Transactions on*, 23(10):1819–1830, Oct 2012.

[94] ZigBee Alliance. *ZigBee Specifications*, December 2006.