# Quantized Network Coding of Correlated Sources in Wireless Sensor Networks

*Mahdy Nabaee*

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

November 2014

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

# Abstract

In many sensor network applications, the sensor readings are inter-node correlated. In such cases, efficient gathering of sensor readings requires distributed compression. Distributed source coding provides practical solutions for compression of these correlated readings when the appropriate rates for the marginal encoding is known at the sensor nodes. In this thesis, we present a data-gathering technique for sensor networks that exploits correlation between sensor data at different locations in the network. Contrary to distributed source coding, our method does not rely on knowledge of the source correlation model in each node although this knowledge is required at the decoder node. Similar to network coding, our proposed method (which we call Quantized Network Coding) propagates mixtures of packets through the network. The main conceptual difference between our technique and other existing methods is that Quantized Network Coding operates on the field of real numbers and not on a finite field. In this thesis, we study our quantized network coding in both lossless and lossy networks.

In the study of lossless networks, we discuss the theoretical foundations for our data gathering technique. By exploiting principles borrowed from compressed sensing, we show that the proposed technique can achieve a good approximation of the sensor readings at the sink node with only a few packets received, and that this approximation gets progressively better as the number of received packets increases. Our first approach is to explain the theoretical foundations for sparse recovery from quantized network coded packets based on an analysis of the Restricted Isometry Property of the corresponding measurement matrices. Extensive simulations comparing the proposed Quantized Network Coding to classic network coding and packet forwarding scenarios demonstrate the delay/distortion advantage of quantized network coding. Furthermore, we discuss the advantages of quantized network coding in a Bayesian scenario where the prior of the sensor readings is available at the decoder node. For such Bayesian scenarios, we also discuss the adaptation of a message passing based decoding algorithm with the aid of simulations.

To study the practicality of quantized network coding in lossy networks, we adapt it into the IEEE 802.15.4 standard which characterizes low rate wireless communication for sensor networks. This is done by developing a comprehensive implementation of the PHY and MAC layers of the standard and then adjusting the MAC layer settings to match with our requirements. Our computer simulations using the developed implementation show

a significant decrease of the delay in many simulation scenarios. The results obtained using this implementation show more advantages for quantized network coding compared to classic routing based protocols especially for high packet drop rates.

# Résumé

Dans de nombreuses applications des réseaux de capteurs, les lectures des capteurs sont corrélées entre les nœuds. Dans de tels cas, la collecte efficace des lectures des capteurs nécessite une compression distribuée. Le codage de source distribué fournit des solutions pratiques pour la compression de ces lectures corrélées lorsque les taux appropriés sont connus pour le codage marginal dans les nœuds capteurs. Dans cette thèse, nous présentons une technique de collecte de données pour les réseaux de capteurs exploitant la corrélation entre les données des capteurs dans les emplacements différents du réseau. Contrairement au codage de source distribué, notre approche ne dépend pas de la connaissance du modèle de corrélation de la source dans chaque nœud, bien que cette connaissance soit nécessaire au niveau du nœud décodeur. Similaire au codage du réseau, l'approche que nous proposons (nous l'appelons le Codage Quantifié de Réseau) propage des mélanges de paquets à travers le réseau. La principale différence conceptuelle entre notre approche et d'autres méthodes existantes est que le Codage Quantifié de Réseau fonctionne dans le domaine des nombres réels et pas dans un corps fini. Dans cette thèse, nous étudions notre Codage Quantifié de Réseau dans les réseaux sans perte ainsi que dans les réseaux avec perte.

Dans l'étude des réseaux sans perte, nous discutons les bases théoriques de notre technique de collecte de données. Utilisant des principes empruntés de la perception comprimée (compressed sensing), nous montrons que la technique proposée peut atteindre une bonne approximation des lectures des capteurs dans le nœud collecteur en recevant peu de paquets, et que cette approximation devient progressivement meilleure quand le nombre de paquets reçus augmente. Notre première approche est d'expliquer les bases théoriques de la reconstruction à partir des paquets codés par le codage quantifié en appliquant une analyse de la Propriété de l'Isométrie Restreinte des matrices de mesures. Des simulations importantes comparant le Codage Quantifié de Réseau proposé avec les codages classiques de réseau et des scénarios d'expédition de paquets montrent l'avantage du codage quantifié de réseau au niveau du compromis délai/distorsion. En outre, nous discutons les avantages du codage quantifié de réseau dans un scénario bayésien où la distribution a priori des lectures des capteurs est disponible au nœud collecteur. Pour de tels scénarios bayésiens, nous discutons également, à l'aide de simulations, l'adaptation d'un algorithme basé sur le passage de messages.

Pour étudier la faisabilité du codage quantifié de réseau dans les réseaux avec pertes,

nous adaptons cette approche à la norme IEEE 802.15.4 qui caractérise la communication sans fil à faible débit pour les réseaux de capteurs. Cela se fait en développant une mise en œuvre complète des couches PHY et MAC de la norme, puis en ajustant les paramètres de la couche MAC pour qu'ils correspondent à nos exigences. Nos simulations informatiques utilisant la mise en œuvre développée montrent une baisse significative du délai dans de nombreux scénarios de simulation. Les résultats obtenus en utilisant cette mise en œuvre montrent plus d'avantages pour le codage quantifié de réseau par rapport aux protocoles classiques basés sur le routage, surtout pour les taux élevés de perte de paquets.

## Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Fabrice Labeau for his guidance and unwavering support during my studies. I thank him for offering excellent conditions to do graduate studies at McGill university and pursue my academic goals. His patient and positive attitude has always helped me tackle difficult situations during my studies.

I must thank Prof. Michael Rabbat and Prof. Ioannis Psaromiligkos for having productive discussions during the work of my doctoral program. I would like to thank Prof. Leszek Szczecinski and Prof. Benoit Champagne for their valuable comments during my doctoral committee meetings. My sincere thanks goes to Prof. Luc Devroye for sharing his knowledge on graph theory with me. I also would like to show my gratitude to Prof. Harry Leib because of his excellent offering of graduate courses which helped me strengthen my mathematical skills.

I owe a great debt of gratitude to Dr. Hugues Mercier and Dr. Mohsen Akbari for having productive discussions and teaching me good qualities for research. I must also thank Dr. Afshin Moein who helped me with French translations in this thesis. I would like to thank all of my friends in Montreal who accompanied me in good and difficult times during my studies.

I should finally thank Hydro-Quebec, the Natural Sciences and Engineering Research Council of Canada and McGill University which provided financial support for the completion of this thesis in the framework of the NSERC/Hydro-Quebec/McGill Industrial Research Chair in Interactive Information Infrastructure for the Power Grid.

I must express my gratitude to my brother, Mojtaba who has always been supportive of me in all stages of my life and helped me achieve my goals. Last but definitely not the least, my greatest gratitude goes to my parents who have made many sacrifices in their lives for me.

## Preface and Contributions

The research works presented in this dissertation are the result of my original work under the supervision of Prof. Fabrice Labeau at the department of Electrical and Computer Engineering, McGill University. Some of the contributions in this dissertation are published in the following scholarly articles. In all of these publications, the second author has initiated the discussion on the topic and provided the first author with the technical advice on the conducted research work. The first author was responsible to investigate the raised questions during the motivating discussions by reviewing the literature, conducting theoretical analysis and implementing computer simulations.

- M. Nabaee and F. Labeau, "Quantized Network Coding for Correlated Sources," in EURASIP Journal on Wireless Communications and Networking, vol. 2014, no. 40, March 2014.

- M. Nabaee and F. Labeau, "Bayesian Quantized Network Coding via Generalized Approximate Message Passing," in Proceeding of Wireless Telecommunications Symposium, Washington, DC, April 2014, pp. 1-7.

- M. Nabaee and F. Labeau, "Non-Adaptive Distributed Compression in Networks," in Proceeding of IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), Napa, CA, August 2013, pp. 239-244.

- M. Nabaee and F. Labeau, "Restricted Isometry Property in Quantized Network Coding of Sparse Messages," in Proceeding of IEEE Global Communications Conference, Anaheim, CA, December 2012, pp. 112-117.

- M. Nabaee and F. Labeau, "Quantized Network Coding for Sparse Messages," in Proceeding of IEEE Statistical Signal Processing workshop, Ann Arbor, MI, August 2012, pp. 828-831.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| CS | Compressed Sensing |
| NC | Network Coding |
| RLNC | Random (finite-field) Linear Network Coding |
| QNC | Quantized Network Coding |
| PF | Packet Forwarding |
| SNR | Signal to Noise Ratio |
| WPAN | Wireless Personal Area Network |
| RIP | Restricted Isometry Property |
| MAC | Medium Access Control |
| PHY | Physical |
| CSMA/CA | Carrier Sense Multiple Access with Collision Avoidance |
| CCA | Channel Clearance Assessment |
| GPS | Global Positioning System |
| i.i.d | Identically and Indenendently Distributed |
| JSM | Joint Sparsity Model |
| TX | Transmit |
| RX | Receive |
| AWGN | Additive White Gaussian Noise |
| IEEE | Institute of Electrical and Electronics Engineers |
| MMSE | Minimum Mean Squared Error |

# Chapter 1

# Introduction

Low implementation and maintenance costs of wireless sensor networks along with their independence from complex wired connections have made them an unavoidable sensing structure in a wide variety of applications [1]. Wireless sensor networks have proved their critical role in monitoring and control of power grids because of their large coverage and efficient data gathering and processing [2]. Sensor networks have also played an important role in environmental sciences by providing new data for the study of wildlife and providing critical warnings about hazardous events, *e.g.* floods [2]. As for health related applications, wireless sensor networks have been used in tele-monitoring of human physiological data, drug administration, tracking of patients and doctors, and fall detection [3].

Although the multi-node structure of sensor networks provides many communication advantages, it also introduces many challenges in terms of design. Many of these challenges are faced in the networking aspects of the communication, as in other wireless technologies [4]. However, in contrast to other technologies, sensor networks may not be able to cope with high computational power. In other words, the design of efficient communication algorithms which are compatible with the low computational power of sensor networks is not trivial and still is the subject of a lot of research.

One of the main functions of a sensor network is to collect sensed data in one (or more) data processing node(s). This task is usually referred to as *data gathering* in sensor network literature or *incast* in the information theory community. In this task's configuration, all of the sensed data in a network needs to be transmitted to a unique node. This node may then act as a *gateway* and forward these sensed data through a different backbone network.

The literature of data gathering mainly covers the schemes in which the sensed data is sent from one node to another node (usually one to which a one-to-one communication is possible), conventionally called *packet forwarding*. After a number of hops[1], the packet containing sensed data arrives at the gateway node. In a larger perspective, the way that packets move around the network is controlled via different protocols, including flooding and routing [5]. Although routing-based protocols [6] offer better efficiency (in terms of communication latency) than flooding, they require online adaptive route calculations which reduces flexibility. Moreover, routing-based packet forwarding faces practical difficulties because of its slow adaptation to network changes, resulting for instance from deploying new node(s) or from link failure(s).

A relatively new type of data gathering relies on the concept of *network coding* [7] where functions of received packets are forwarded in the next hop. Network coding has shown to have theoretical and practical advantages over routing-based packet forwarding, by providing a network diversity for the information flow [8, 9]. Since network coding is done in a distributed manner, it preserves the flexibility of the sensor network with respect to deployment changes. Specifically, this is achieved by the use of random network codes which are theoretically shown to be appropriate in many practical cases [10, 11]. However, the exact circumstances in which sensing and data gathering occur determine the appropriate protocol for network transmission. Some of the determining factors include the type of communication (unicast or multicast), packet drop rate, link failure statistics, and the statistics of sensed data.

When dealing with sensed data that presents inter-node correlation, efficient data gathering calls for compression of sensed data. Thanks to work carried out on distributed source coding [12, 13], it is possible to decrease the unnecessary flow of information in the network. In such a case, although there is no information theoretic difference between the use of packet forwarding or network coding [14], network coding has shown to be a better alternative to packet forwarding because of its advantages. Some recent work has focused on the design of practical codes for such purpose [15–17].

Distributed source coding requires the knowledge of appropriate marginal coding rates at each encoder node. However, in many scenarios, the assumption of this knowledge at the encoder side (*i.e.* in each node) might not be practical, especially when the characteristics

[1] A single transmission of the packet from one node to another node is also called *one hop* in the literature.

of sensed data change over time.

Motivated by this observation, we aim to develop a data gathering scheme which does not need the knowledge of inter-node correlation of sensed data but at the same time can take advantage of this correlation. Specifically, we investigate the feasibility of using compressed sensing [18, 19] concepts within the network coding framework. Compressed sensing theory offers an alternative compression strategy instead of conventional transform coding. However, unlike transform coding, compressed sensing does not need the knowledge of the correlation of data, at the encoder side. This is achieved by the use of random codes, which can be integrated into the random network coding structure. Motivated by the idea of using compressed sensing theory along with network coding, we develop a data gathering and transmission scheme, which can intrinsically perform distributed compression of inter-node correlated sensed data. The embedded distributed compression will result in a lower delay to achieve a decoding quality, compared to the other transmission methods with no need to the knowledge of correlation of data. It is also the first attempt to study the theoretical and practical applicability of this idea in a wireless sensor network.

Our data gathering scheme is inspired by network coding, but acts in the field of real numbers, allowing us to develop recovery algorithms at the gateway node, which provide approximate data recovery with low delay. The developed recovery mechanism is based on the ideas borrowed from the compressed sensing theory.

## 1.1 Objectives and Contributions

In this thesis, we combine the idea of using real field network coding with the concepts of compressed sensing to propose a *non-adaptive distributed compression* scheme, called *Quantized Network Coding* (QNC), for sparse and near-sparse sensed data[2]. We also discuss the theoretical feasibility of compressed sensing based network coding. In a lossless communication setting, we evaluate the compression performance of QNC compared to other data gathering methods. Our contributions with the assumption of a lossless sensor network focuses on studying the distributed compression capability of our proposed QNC scheme and can be summarized as follows:

- Introduction and Formulation of our novel Quantized Network Coding scheme of

[2]The term *non-adaptive* is used to reflect the fact that the encoding does not need to adapt itself to the characteristics of the sensed data in order to obtain high efficiency.

transmission and data gathering,

- Investigation of robust recovery of quantized network coded data using concepts borrowed from compressed sensing and specifically using restricted isometry property of random matrices,
- Asymptotic study of robust recovery of quantized network coded data using the results from Bayesian compressed sensing,
- Derivation of a bound on the recovery error of quantized network coded data using a restricted eigenvalue condition,
- Numerical evaluation of compression capability of quantized network coding in a lossless sensor network.

Our contributions for sensor networks with lossy communication may be listed as follows:

- Implementation of QNC for a practical wireless sensor network by adapting it to the IEEE 802.15.4 standard [20],
- Numerical evaluation of QNC scheme in a wireless sensor network with loss and interference.

## 1.2 Organization

This thesis is organized as follows. In Chapter 2, we provide a comprehensive review of the related literature. Specifically, we present a review of packet forwarding and network coding as two basic types of transmission methods used for data gathering. A review of the related studies in the information theory literature is also presented for packet forwarding and network coding. Further, the literature on the use of compressed sensing theory in sensor networks is reviewed in this chapter. Finally, we provide a brief review of the compressed sensing and sparse recovery literature and mention some of the related results, used in this thesis.

In Chapter 3, we introduce our proposed quantized network coding scheme of transmission and formulate it in a lossless network setting. We also discuss the choice of network coding coefficients for the proposed QNC scheme and suggest an appropriate design for

those coefficients. Using the concepts and results borrowed from the compressed sensing theory, we propose a recovery algorithm for the quantized network coded messages. Chapter 3 is concluded by our comprehensive simulation results, comparing the performance of QNC with a few different data gathering schemes, in terms of compression/delay efficiency.

In Chapter 4, we discuss the Bayesian framework for quantized network coding of messages. In the Bayesian scenario, we provide a theoretical bound on the decoding error of messages when a version of QNC, called one-step QNC, is used for transmission. This asymptotic result shows that the required number of received packets is of a smaller order than the number of sensed messages. We conclude Chapter 4 by discussing the use of a message passing based decoding algorithm for the Bayesian QNC scenario.

In Chapter 5, the adoption of QNC in a lossy network is studied. Specifically, we implement the QNC scenario over the PHY and MAC layers of IEEE 802.15.4 which describes the communication of nodes in a wireless sensor network. We provide a brief review of related parts of IEEE 802.15.4 standard in this chapter. Finally, in this chapter, we present our simulation results for QNC and routing based packet forwarding schemes of data gathering in lossy networks.

The concluding remarks are presented in Chapter 6.

## 1.3 Notation

Upper case letters represent random variables whereas their realizations are denoted by lower case letters. Throughout this thesis, the underlined letters represent vectors, which are column-wise unless specified. The $\ell_0$, $\ell_1$ and $\ell_2$ norms of the vectors are denoted by $||\cdot||_{\ell_0}$, $||\cdot||_{\ell_1}$ and $\|\cdot\|_{\ell_2}$, respectively. Further, the cardinality of sets is represented by $|\cdot|$. In the following, we provide an abstract description of data gathering scenario and the associated notation.

### Network

A sensor network is built up by its sensor nodes which have sensing, communication and computational capabilities. Each sensor node is denoted as $v \in \mathcal{V}$ where $\mathcal{V}$ is a finite sorted

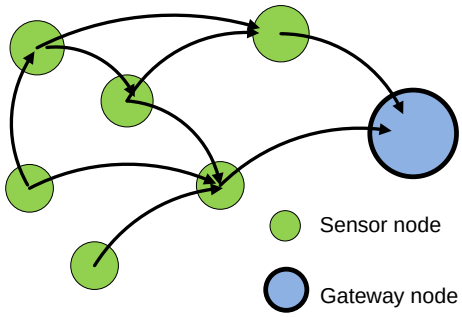



**Figure 1.1** Directed graph, representing a data gathering sensor network.

set of nodes [3] with $|\mathcal{V}| = n$. Sensor nodes are required to be able to maintain communication with other nodes in the same WSN. Usually, the ability to maintain communication with another node is determined by many factors of the deployment and media access control, including the power of transmitters, power of the noise and inter-channel interference. In an abstract model, we say that a node $v_1 \in \mathcal{V}$ can maintain communication to another node $v_2 \in \mathcal{V}$ if the received signal strength from the transmitter node to the receiver node is above a certain threshold. This is characterized in IEEE 802.15.4 standard by defining a receiver sensitivity parameter.[4] Such communication ability is represented by a directed edge $e \in \mathcal{E}$ where $tail(e) = v_1$ and $head(e) = v_2$. We represent the sensor network by the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, as shown in Fig. 1.1.

We define the sets of incoming and outgoing edges of node $v$, denoted by $In(v)$ and $Out(v)$, respectively, as follows:

$$In(v) = \{e : e \in \mathcal{E},\ head(e) = v\}, \tag{1.1}$$

$$Out(v) = \{e : e \in \mathcal{E},\ tail(e) = v\}. \tag{1.2}$$

The set of nodes $\mathcal{V}$ and set of edges $\mathcal{E}$ are both a subset of positive integer numbers.

### Source signals

Each sensor node $v$ makes periodic sensing measurements which are modeled by a random variable $X_v \in \mathcal{R}$. We assume that messages, $X_v$'s, take their values in a bounded interval

[3] Also called vertices in the literature

[4] A more precise measure of connectivity may use signal to noise ratio or the packet drop rate at the receiver node.

between $-q_{\max}$ and $+q_{\max}$; explicitly:

$$-q_{\max} \leq X_v \leq +q_{\max}. \tag{1.3}$$

This is a reasonable assumption as the sensing range of sensors is usually limited[5]. In this thesis, $X_v$ is called the *message* of sensor node $v$ and the sorted set of messages is formally noted as:

$$\underline{X} = [X_v : v \in \mathcal{V}]. \tag{1.4}$$

### Data Gathering

Having these information sources and the information network characterized, we study the transmission of $X_v$'s to a single gateway node. The gateway or decoder node, denoted by $v_0$, $v_0 \in \mathcal{V}$, has high computational resources and is usually in charge of forwarding the information to a next level network; *e.g.* a wired backbone network. Although the computational capability of other sensor nodes is not high we do not intend to model and study its effect in this thesis. The described incast of sources to the unique decoder node is referred to as *data gathering*.

In this thesis, we study data gathering scenario in wireless sensor networks. This is done by describing the conventional approaches based on packet forwarding and discussing their issues. Then, we will introduce and discuss our new data gathering method which addresses some of those issues.

[5] The choice of $q_{\max}$ can be made after a statistical study of the actual physical values of $X_v$'s, and can be chosen as some confidence region, in which most of the realizations of $X_v$'s lay.

# Chapter 2

# Literature Review

In this chapter, we review the literature related to the subjects discussed or used in this thesis. We begin the chapter by providing a background review on data gathering in wireless sensor networks. Specifically, in Section 2.1, packet forwarding and network coding are described as two types of transmission in networks. In Section 2.2, we provide an information theoretic comparison of these two types of transmission and discuss their differences in theory and practice. A review of compressed sensing and sparse recovery theory is presented in Section 2.3. The literature of using compressed sensing in wireless sensor networks is reviewed in Section 2.4.

## 2.1 Data Gathering Methods

Energy consumption and delivery delay are two important criteria in the design and adoption of different methods of transmission. In the following, we review data gathering schemes and separate them into three different categories.

### 2.1.1 Packet Forwarding

In many (and in fact in all of the practical) networks, packet forwarding is the basis of telecommunications from one node to another node in the network. When a packet is received at an intermediate node, it is stored in a queue to be served (transmitted to the next intermediate node) at its appropriate time. In general, if the content of the packet is not changed, it is said that *packet forwarding* (PF) is used for transmission in the network

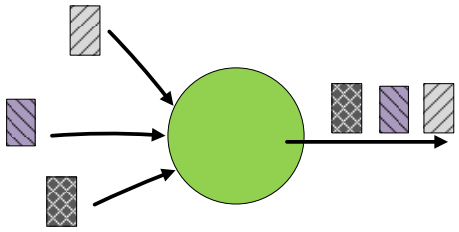


**Figure 2.1** Packet Forwarding at an intermediate node: Each incoming (received) packet is sent to the next outgoing node at its appropriate time. (The tiled squares represent the packets and the green circle is the intermediate node.)

(shown in Fig. 2.1). Although the way packet forwarding works is simple, it involves many practical difficulties both locally at the intermediate node and globally throughout the network.

One of the big challenges is to provide each message with a path to travel through the network to get to its destination, *e.g.*, the gateway node. Such paths, or the way packets are travelling in the network, is determined by a *routing* protocol [6]. Developing suitable routing protocols which satisfy different design requirements, including delay, energy consumption and reliability, has been the topic of a lot of research.

Flooding and gossiping are two basic routing protocols which relay packets without using their destination addresses. Although they can be implemented with no need for overhead transmission between the nodes, they have several drawbacks, including the overlap of coverage by different nodes, resulting in large energy consumption [21]. We review the application of randomized gossiping for the consensus problem in the following section. Iterative protocols in which the routes are in-frequently updated according to a rule are also proposed for routing of packets in the network.[1] Typically, the routing protocols try to find the *shortest path* or the path with the smallest point-to-point delay for forwarding of the packets. Hierarchy, geography, network flow and quality of service are the main parameters which are categorizing routing protocols proposed in the literature [5].[2]

[1] Usually, the rule is developed based on a criterion, *e.g.*, minimum consumed transmission power, minimum delay.

[2] As routing is not the focus of this thesis, we will not provide a comprehensive review of the literature on routing protocols. The reader may refer to [5, 6] for a full review of the available routing protocols in the literature.

### 2.1.2 Data Aggregation

Although packet forwarding is an essential part of multi-hop transmission in networks, it solely may not be efficient when it is dealing with redundant data. In [22], *data aggregation* is defined as the global process of routing data in the network and processing them at the intermediate nodes, with the objective of decreasing resource consumption and increasing the network lifetime. More specifically, a data aggregation protocol is built up from three main elements: an appropriate routing protocol, an effective aggregation function for the intermediate nodes, and a suitable representation of data at the nodes.

Usually, a specific category of routing protocols, called data-centric, are used in data aggregation. In contrast with classical routings where the shortest path is found, the nodes route packets based on their content. Some of the factors which determine the next hop for each packet are the position of suitable aggregation node, the data type, and the priority of information.

The aggregation functions used may be lossy or lossless and can be performed infrequently or periodically at each hop. Some simple aggregation functions include first order statistics; *e.g.* mean, min/max, and elimination of duplicates.

Usually, conventional transform coding based representation is a suitable representation of data for data aggregation. However, an appropriate design of data aggregation requires joint optimization of routing, aggregation and representation for each specific case in practice.

Generally, finding the joint optimal routing and compression structure is known to be NP-hard, referred to as Steiner tree [23]. Theoretical results show that for uncorrelated messages, data aggregation is not necessary [24]. In other words, forwarding packets along the shortest path is the best strategy. In contrast, when the messages are highly correlated, the best strategy is to perform compression as soon as possible [24]. In the intermediate cases, cluster-based solutions may be the optimal structures although no theoretical proofs are available to support it, as claimed [24]. Further, it is also a theoretical observation that deploying hierarchical networking topologies offers a close to optimal solution, in all of these cases, when there is no packet dropped [25]. In hierarchical networking, the nodes relay information according to a tree for which the gateway is the root node.

To overcome the vulnerability of hierarchical routing to packet drops, a few works have proposed that each node broadcasts the aggregated data to all the neighbouring nodes,

as opposed to sending it to a single next hop. These approaches try to create a multi-path structure for networking; Synopsis diffusion is one such multi-path protocol in which imaginary rings around the data sink compose a topology for forwarding of packets in the network [26].

We can summarize the drawbacks of existing data aggregation protocols as follows:

- hierarchical routing of packets which makes them vulnerable in lossy networks,
- the need for awareness of network topology or data correlation for efficient design of routing and aggregation functions,
- the need for feedback for adaptation to changes in the network deployment and data characteristics, resulting in extra overhead communication.

The distributed consensus problem deals with distributed estimation of a scalar or vector function from measurements of sensor network [27]. Advances in communication theory have drawn attention to the distributed consensus problem in networking and networked control theory. One of the widely studied examples of distributed consensus is averaging which is considered as the prototyping example of a consensus problem. The averaging consensus mainly focuses on finding the average (a single scalar value) of sensor readings usually coming from the same source, *e.g.* the temperature of the environment. Distributed averaging and gossip algorithms are two main algorithms used for the averaging consensus. In the distributed averaging algorithm, each node broadcasts its consensus to its neighbouring node at each iteration [28]. In [29], the authors have taken advantage of the broadcast nature of wireless sensor networks to propose a greedy neighbour selection for the gossiping algorithm.

In our data gathering scenario, the messages need to be estimated at one single node. This is in contrast with the consensus where the functions are estimated at all of the nodes. Considering the distributed consensus of a scalar value (*e.g.* the average value), one may run consensus $n$ times to estimate all of the messages, resulting in a time-consuming and inefficient process. If the messages are linearly dependent, one may run consensus for as many times as the number of degrees of freedom of messages. Furthermore, the knowledge of linear relation between the messages may also be needed at all the nodes to use distributed consensus for our data gathering.

### 2.1.3 Network Coding

Inspired by computer network applications,the problem of network information flow was initially introduced by Ahlswede *et al.* with an objective to increase the network throughput [7]. To demonstrate the advantage of network coding, we present the following simple example.

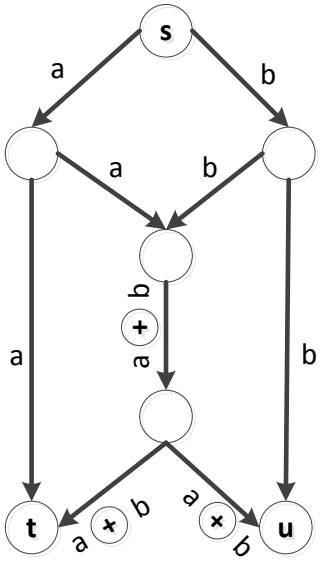



**Figure 2.2** Butterfly Network: Advantage of Network Coding over Routing based Packet Forwarding

In Fig. 2.2, a directed butterfly graph is shown which may represent a network with one source node, s. The source node has two different messages, a and b, which need to be delivered to both of t and u nodes. If simple packet forwarding was used, the messages a and b can be delivered to t and u via the side links, respectively. However, a can not be delivered to u at the same time as b being delivered to t because they have to share the middle link in time. In other words the throughput of the network would be 0.5 message per network use, assuming a unit capacity for the links and one bit representation for the messages. Now, assume that we transmit a + b over the middle link of the network. Since a is available at t, the received node t can decode b from the received a + b by performing an xor operation. Similarly, a can be decoded at node u, at the same time, resulting in a throughput of one message per network use for the network.

To formulate network coding, we describe some of the related notation used in this

thesis. We represent the content of a packet received at node $v \in \mathcal{V}$ over link $e$ by $Y_{e,rx}(t)$, where $t$ is an integer time index. The time index $t$ is used as an index for the steps of network coding, during which a block of $L$ bits are transmitted. In this thesis, we refer to $L$ as the *packet length*, representing the content sent over a link.[3] As a result of limited capacity of edges, the content of received packets, $Y_{e,rx}(t)$'s are from a finite alphabet. The content, sent over a link $e \in \mathcal{E}$, is represented by $Y_{e,tx}(t)$.

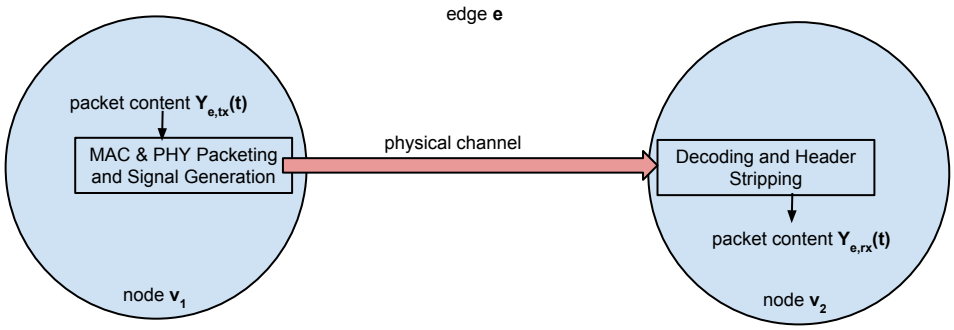



**Figure 2.3** Link Modelling

In general, network coding [7] at the nodes is defined as in below (shown in Fig. 2.4):

**Definition** In a network, denoted by the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the transmission is said to be performed using *memoryless network coding* if at all nodes, $v$, the content of output ports, at time $t+1$, $Y_{e,tx}(t+1), e \in Out(v)$, are functions of the incoming packets, $Y_{e,rx}(t), e \in In(v)$, and the node message, $X_v(t)$, that is:

$$Y_{e,tx}(t+1) = \mathbf{f}\Big(X_v, Y_{e',rx}(t) : e' \in In(v), t, e\Big) \ , \forall \ e \in Out(v) \ , \forall \ v \in \mathcal{V}. \tag{2.1}$$

In contrast with packet forwarding, where long queues of packets may need to be stored at the nodes, memoryless network coding only requires to store the very last received packets. It has also been shown that there is no need to use the previously received incoming packets, at each node, to improve the performance of network coding in terms of information rate, and memoryless network coding suffices [7, 11].

Network coding needs to forward packets between the nodes. It also processes the packets at intermediate nodes by performing network coding function on them.

[3] We do not discuss the effect and value of $L$ at this point, but it will be discussed in the next chapters.

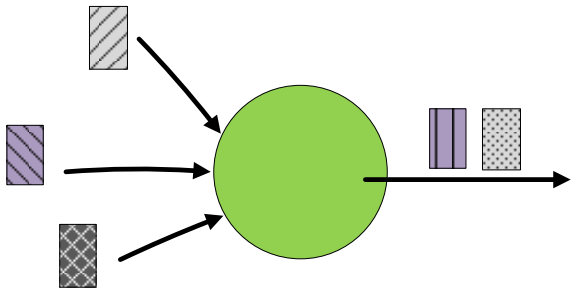


**Figure 2.4** Network Coding at an intermediate node: Incoming packets are combined together according to the network coding function and then are sent to the next hop at its appropriate time.

### Linear Network Coding

In general, the so called *network coding function* **f(.)**, may have any characteristic. However, theoretical works have shown that even simple operations can be efficient for network coding. Initially proposed in [10], linear network coding was shown to achieve the information theoretic limits of the network [11].

**Definition** If the network coding function **f(.)** is a linear function of messages and the received packets:

$$Y_{e,tx}(t+1) = \alpha_{e,v}(t)X_v + \sum_{e' \in In(v)} \beta_{e,e'}(t)Y_{e',rx}(t), \ \forall e \in Out(v), \tag{2.2}$$

the transmission is called *linear network coding* and can be characterized completely by using the coefficients of the linear functions, $\alpha_{e,v}(t)$'s and $\beta_{e,e'}(t)$'s, called *network coding coefficients* [10]. Further, all of the operations are in *finite field* and if the network coding coefficients are random, then the transmission is called *random* linear network coding (RLNC).

In practice, the representations are binary and therefore the operations can be done by using a digital implementation of binary operations.

Now, considering our data gathering scenario, if all of the messages are from a finite field **F**, then all of the operations for linear network coding have to be in the same field. Locally, at each node, the content of outgoing packets is a finite field linear combination of the contents of incoming packets to that node and the message of that node. Therefore

globally, the contents of received packets at the gateway node are linear combinations of the messages, assuming an initial rest condition. The mapping from the messages to the contents of collected received packets at the gateway node is represented by a matrix. If the matrix is invertible in the field, then the network coding is said to be *valid* [11]. The *initial rest condition* is also defined such that for the initial time index $t = 1$:

$$Y_{e,rx}(t) = 0, \ \forall e \in In(v), \ \forall v \in \mathcal{V}. \tag{2.3}$$

Ho *et al.* have shown that using random network coding coefficients, picked uniformly from the field $\mathbf{F}$, results in a valid network coding, with an overwhelming probability [11]. Explicitly, they proved the following theorem:

**Theorem 2.1.1** ***(Random Linear Network Coding)*** *[11] Considering an arbitrary network in which messages from a finite field are required to be transmitted to a gateway node. Further, assume that all of the network coding coefficients are picked randomly and uniformly from the finite field $\boldsymbol{F}$ with alphabet size of $2^{q_0}$ and $q_0 > 1$. If there is a valid solution where all the coefficients are deterministic fixed values, then the probability that the proposed random solution is valid is at least:*

$$(1 - 1/q_0)^{|\mathcal{E}|}, \tag{2.4}$$

*where $|\mathcal{E}|$ is the number of edges in the network.*

As this theorem suggests the probability of a valid network coding increases as the size of the field is increased.

For the cases where the sources are real-valued, it may be of advantage to consider network codes in the real field. In [30], *real network codes* have been studied for multi-resolution multicast and wireless multicast scenarios. Specifically, the vector of messages of a single source is transmitted to the destination node via a multi-hop network with multiple links between the intermediate nodes, as shown in Fig. 2.5 [30]. This is different from our data gathering scenario in which messages of different nodes are transmitted to the gateway node. Since the network coded packets sent over the links are from a quantized set of values, the quantization noises would have a distorting computational effect. As shown in [30], theoretical analysis of the condition number of the matrix, mapping the source's vector to the contents of the received packets at the receiver node, helps to understand the

effect of these quantizations. This result is then used to provide bounds on the achievable rate by using real network codes in the aforementioned multicast scenarios.

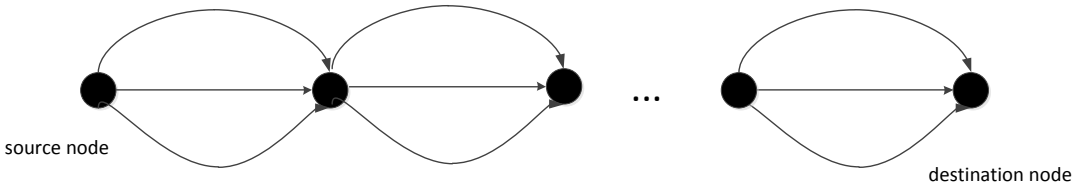



**Figure 2.5** The scenario studied in [30] where real network coding is used. The vector of message of a single source is transmitted to a destination node via multiple hops.

### Physical Layer Network Coding

In wireless networks, to avoid collision of multiple packets, they have to be scheduled in different time slots. In such cases, network coding in real field may be a solution to avoid scheduling of them [31]. The concept of *physical layer network coding* was initially proposed in [32] to exploit the network coding advantages that occur naturally over the electromagnetic waves[4]. This is basically motivated by the additivity of signals over the media, making it capable of performing a network coding operation over the air. Conventionally, interference signals are treated as an undesired factor and we have to schedule packets such that the interference is avoided. Essentially performing signal coding and forwarding (as opposed to packet forwarding), the following example shows the benefit of using analog network coding. In the following, we present an example in which physical layer network coding helps increasing the throughput [32].

As shown in Fig. 2.6, Alice and Bob need to transmit their messages to each other. Router is in the radio transmission range of both Alice and Bob. However, Alice and Bob are not in the radio transmission range of each other. Hence, they have to rely on Router to forward their messages. Specifically, assume that Router can send or receive at a single time. Traditionally, one of the sender nodes (say Alice) sends its message to Router and Router receives and stores it. Then, in the next time slot, the other sender (Bob) sends its message to Router and Router receives and buffers it. In the next two time slots, Router sends the message of Alice to Bob and then the message of Bob to Alice. Router can not

[4]Katti *et al.* have also called it *analog network coding* in [33] where the authors proposed a practical algorithm for its implementation.

send the message of both Alice and Bob at the same time, as they are interfering with each other. Now, using the concepts of network coding, one may do the following to decrease the transmission time. If at the first time slot, we let both Alice and Bob send their messages at the same time to Router, Router will receive a weighted sum of the signals, representing the messages of Alice and Bob. Then, in the second time slot, if Router transmits the same received signal, Alice and Bob can receive that weighted sum signal. Since Alice already knows the signal representing its own message, she can recover the signal, representing the message from Bob, by performing a simple subtraction.[5] Similarly, Bob can recover the signal, representing the message of Alice. Hence, the time slots required to transmit their messages to each other is decreased by half, increasing the network throughput.


**Figure 2.6** Wireless Networking via a router: Alice and Bob need to send their messages to each other. Physical layer network coding allows them to do this in smaller number of time slots than traditional packet forwarding.

## 2.2 Information Flow in Networks

In this section, we review the literature on the information theory aspect of transmission in the network. Specifically, we present the available bounds on the entropy of the sources which could be transmitted successfully (without loss) through the network capacity. Inspired by the notation in [7], we represent,

$$\underline{h} = [h_v : v \in \mathcal{V}], \tag{2.5}$$

where $h_v = \mathbf{H}(X_v)$, is the vector of entropies of the sources.[6]

Most of the works on network information flow analysis are focused on lossless networks because of their simplicity. There are still many problems for the lossy networks which are still not solved. In this section, we assume that the networks have lossless directed links but with a limited capacity, namely $C_e$ for the link $e \in \mathcal{E}$. We present the main results for

[5] This assumes that Alice knows the channel attenuations.

[6] $\mathbf{H}(.)$ denotes the conventional *entropy*.

two cases of independent and correlated sources in two separate parts, as follows.

### Independent Messages

For independent messages where the joint probability density function of messages is separable:

$$\mathbf{p}_{\underline{X}}(\underline{x}) = \prod_{v\in\mathcal{V}} \mathbf{p}_{X_v}(x_v), \tag{2.6}$$

the lower and upper bounds on the entropy vector are known. Specifically, for a lossless network and data gathering scenario, it is shown in [34] that:

$$\sum_{v\in\mathcal{S}} h_v \leq \sum_{\substack{e\in\ \mathcal{E} \\ tail(e)\in\mathcal{S} \\ head(e)\in\mathcal{S}^c}} C_e \ , \quad \forall\ \mathcal{S}\subseteq\mathcal{V}-\{v_0\}, \tag{2.7}$$

where $\mathcal{S}^c = \mathcal{V} - \mathcal{S}$, and $v_0$ is the index of the decoder node [7] . This is also referred to as the *min-max (cut-set) outer bound* in many different contexts [7, 35], and is illustrated in Fig. 2.7. This result is said to be the capacity rate region as it is an achievable rate region and also one may not achieve higher rates, as the converse proves. Moreover, this result may also be obtained by simplifying the results in [36, 37]. It is also worth-noting that the bounds can be achieved by using either of network coding and packet forwarding (via routing) [34].

For the case of lossy erasure networks, in which each link is an erasure channel and the side information (about the happening of an erasure) is provided at the destination node, the capacity is known [38]. By simplifying the reported capacity region for our data gathering scenario, one may obtain that:

$$\sum_{v\in\mathcal{S}} h_v \leq \sum_{\substack{v'\in\mathcal{S} \\ \exists e\in\mathcal{E}\ :\ tail(e)=v', \\ head(e)\in\mathcal{S}^c}} \Big(1 - \prod_{e'\in\ Out(v'),head(e')\in\mathcal{S}^c} \varepsilon_{e'}\Big) \ , \quad \forall\ \mathcal{S}\subseteq\mathcal{V}-\{v_0\}\ , \tag{2.8}$$

and $\varepsilon_e$ is the probability of occurrence of an erasure (for one bit) at edge $e$.

As presented in [38], the above rate region is obtained by using network coding and can not be obtained by using packet forwarding.This fact shows one of the advantages of

[7] Symbol "−" denotes the set subtraction operator.

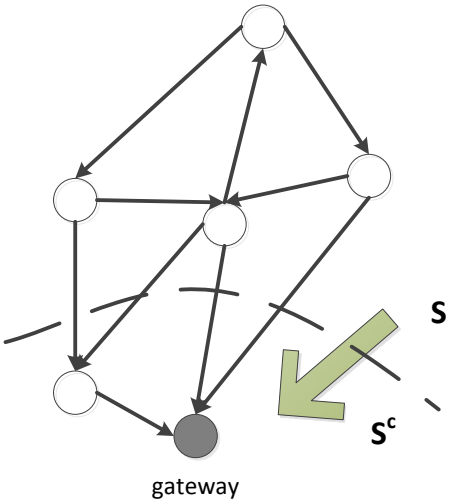



**Figure 2.7** Min Max Outer Bound is the capacity for data gathering scenario in lossless directed networks.

network coding over packet forwarding, in the case of lossy networks.

### Correlated Messages

It is also worth mentioning that the data gathering problem with correlated messages can be considered as a generalization of multi-terminal coding [39], when the messages are transmitted through multiple hops from the encoders to the decoder, instead of a single hop of encoder nodes to decoder node.

The problem of data gathering when the messages are correlated can not be formulated by using only the entropy tube, $\underline{h}$. In the literature, the conditional entropies are used to characterize the constraints implied by the network on the sources. By simplifying the results in [14, 34], we have that for a lossless network, the general joint probability distribution function of correlated sources has to be such that:

$$\mathbf{H}(\underline{X}_{\mathcal{S}}|\underline{X}_{\mathcal{S}^c}) \leq \sum_{\substack{e\in\ \mathcal{E} \\ tail(e)\in\mathcal{S} \\ head(e)\in\mathcal{S}^c}} C_e\ ,\quad \forall\ \mathcal{S}\subseteq\mathcal{V}-\{v_0\}, \tag{2.9}$$

where $\underline{X}_{\mathcal{S}} = \{X_v : v \in \mathcal{S}\}$ and $\underline{X}_{\mathcal{S}^c} = \{X_v : v \in \mathcal{S}^c\}$. The above bound can not be outperformed as the converse proof shows it. It is easy to show that the bound in (2.9) is simplified to the one in (2.7) when the messages are independent.

It is interesting to note that there is no need to perform network coding in this case, to achieve the boundaries of the entropy condition of (2.9). Moreover, to achieve the highest possible information rates, the distributed source coding of correlated messages (*i.e.;* Slepian-Wolf coding), is completely separated from the information flow through the network. However, this is a true statement only when enough prior information is available at the encoders to generate optimal Slepian-Wolf codebooks (*i.e.* appropriate rates for encoding at each source node).

The error rate advantages of network coding has motivated the work on its adoption in many incast and multicast scenarios. Specifically, in [11, 40], the advantage of using randomized linear network coding for multicast scenarios is studied and an upper bound for the transmission error probability is derived. The authors in [11, 40] study the case of two arbitrarily correlated sources.For such case, since both distributed source coding and network coding is performed on the sources, they required a joint source-network decoder to perform optimal (with respect to the resulting error probability) reconstruction. Their so called $\alpha$-decoder was essentially a maximum a-posteriori decoder [11, 40].

It is also interesting to see that performing distributed source coding [13], separate from network coding is shown to be sufficient [41].[8] In a few recent works, the authors have proposed practical code designs for performing network coding [15–17]. However, it is important to note that all of them have assumed a knowledge of correlation between the sources (for instance their Hamming distance) during the design of source encoders.

### Advantages of Network Coding over Packet Forwarding

In the following, we summarize the main advantages of network coding over routing based packet forwarding:

- **Lossy Networks:** In contrast to the case of lossless networks, network coding can be advantageous in lossy networks, in terms of the achievable rate regions. For instance, considering the case of a network with erasure channels (links), it was shown in [38] that only network coding can achieve the capacity rate region of the network. Intuitively, this is expected as the multi-path diversity, embedded in network coding, may decrease the probability of error and have an effect on the achievable rate regions.

[8] Joint source-network coding is still required to achieve decoding with minimum error rate.

- **Network Deployment:** Changes in the deployment of the network, including node elimination, force the routing-based packet forwarding to find new routes for the messages. As a result, routing-based packet forwarding cannot show a fast and efficient response to these changes. As an alternative for packet forwarding, in different scenarios, network coding can be very advantageous. The diversity in information flow is the main reason why it is not highly sensitive to the changes in the deployment.

## 2.3 Compressed Sensing and Sparse Recovery

The concept of compressed sensing or compressed sampling (CS) was proposed to integrate sensing (sampling) and compression (transform coding) tasks, when there is a computational resource constraint. However, many of its theoretical results have been used in other applications where near-sparse or exactly sparse signals are involved. A vector $\underline{s} \in \mathcal{R}^n$ is said to be *k-near-sparse* if there is a small positive scalar $\epsilon_k$ for which:

$$\frac{\left|\left|\underline{s} - \underline{s}_{(k)}\right|\right|_{\ell_1}}{||\underline{s}||_{\ell_1}} \leq \epsilon_k, \tag{2.10}$$

and $\underline{s}_{(k)}$ is (exactly) $k$-sparse:

$$\left|\left|\underline{s}_{(k)}\right|\right|_{\ell_0} = k, \tag{2.11}$$

In this thesis, $\epsilon_k$ is called *near-sparsity parameter*. Moreover, a vector $\underline{x}$ is said to be near-sparse in a transform domain, characterized by the matrix $\phi_{n\times n}$, if there is a near-sparse vector $\underline{s}$ such that:

$$\underline{s} = \phi^T \cdot \underline{x}. \tag{2.12}$$

An example of the sparsifying transform matrix, $\phi$, is the Karhunen Loeve transform of the vector. Note that the sparsity model used in this thesis is different from the joint sparse model (JSM) [42], in that our node source signals or *messages* are scalar random variables, with no temporal correlation in each node. This is a valid assumption for stationary sources as a local transform coding could be applied to the time samples and generate a set of samples with no or little time redundancy. In this thesis, we study the data gathering scenario for correlated messages which are $k$-near-sparse in the transform domain $\phi$ with the near-sparsity parameter $\epsilon_k$, as explicitly described above.

Now, consider a $k$-sparse vector, $\underline{s} \in \mathcal{R}^n$, with a random support, called $supp(\underline{s})$. If the

support of this vector, $supp(\underline{s})$, is known at the encoder, classic compression [43] tools offer an explicit optimal design, both in terms of compression ratio and allocated computational resources. However, in most practical cases, $supp(\underline{s})$ is not known, but its sparsity is, *i.e.* $k = |supp(\underline{s})|$. For such cases, compressed sensing has been proposed, in which a global non-adaptive sensing structure and a minimum mean squared error decoder are developed.

The compressed sensing [19] problem considers minimum mean squared error recovery of a $k$-sparse vector, $\underline{s}$, from an under-determined set of measurements, $\underline{z}$, of the form

$$\underline{z}_{m\times 1} = \Theta_{m\times n} \cdot \underline{s}_{n\times 1}, \tag{2.13}$$

where $m \ll n$. Although the problem is ill-conditioned, if $m \geq k$, and the location of non-zero components, *i.e.* supp($\underline{s}$), is known, the problem can be solved.

**Definition** An $m \times n$ matrix, $\Theta$, where $m < n$, is said to satisfy the *Restricted Isometry Property* (RIP) of order $k$, with the positive constant $\delta_k$, if:

$$1 - \delta_k \leq \frac{\|\Theta\ \underline{s}_k\|_{\ell_2}^2}{\|\underline{s}_k\|_{\ell_2}^2} \leq 1 + \delta_k,\ \forall\ \underline{s}_k \in \mathcal{R}^n\ , \tag{2.14}$$

where $\underline{s}_k$ is $k$-sparse.

In other words, $\Theta$ should preserve the $\ell_2$ norm of $k$-sparse vectors. A family of random matrices, including Gaussian, sub-Gaussian and binary matrices, has been shown to satisfy RIP with overwhelming probability [44]. In the following, two theorems are presented which are used in the next chapters of this thesis. These two theorems describe a recovery algorithm for compressed sensed messages and offer a recovery error bound for that recovery algorithm using the restricted isometry property.

**Theorem 2.3.1** ***(Noiseless Compressed Sensing)*** *[18, 45, 46] If an $m \times n$ matrix, $\Theta$, satisfies RIP of order $2k$, with a constant, $\delta_{2k} < \sqrt{2} - 1$, then there is a unique solution to the linear equation $\underline{z} = \Theta \cdot \underline{s}$, where $\underline{s}$ is $k$-sparse [18, 45]. The unique solution can be calculated, according to:*

$$\arg\min_{\underline{s}'} ||\underline{s}'||_{\ell_1},\quad s.t.\ \Theta\ \underline{s}' = \underline{z}\ , \tag{2.15}$$

*which can be solved by using linear programming [46].*

The type of recovery algorithm, specified in (2.15), is called $\ell_1$-min decoding. For the case of noisy measurements where,

$$\underline{z}_{m\times 1} = \Theta_{m\times n} \cdot \underline{s}_{n\times 1} + \underline{n}_{m\times 1}, \tag{2.16}$$

compressed sensing and decoding is still applicable, specified in the following theorem.

**Theorem 2.3.2** ***(Noisy Compressed Sensing)*** *[47, 48] Consider the noisy measurement equation of* (2.16) *where* $\Theta$ *satisfies RIP of order* $2k$ *with* $\delta_{2k} < \sqrt{2}-1$*, and* $\underline{s}$ *is an arbitrary* $k$*-sparse vector. Moreover, consider the recovery method,*

$$\hat{\underline{s}} = \arg\min_{\underline{s}'} ||\underline{s}'||_{\ell_1}, \quad s.t.\ \|\Theta\ \underline{s}' - \underline{z}\|_{\ell_2} \leq \epsilon_{rec}, \tag{2.17}$$

*where* $\epsilon_{rec}$ *is an upper bound on the* $\ell_2$*-norm of measurement noise, that is:* $\|\underline{n}\|_{\ell_2} \leq \epsilon_{rec}$*. For such scenario, we have:*

$$\|\underline{s} - \hat{\underline{s}}\|_{\ell_2} \leq c_1\ \epsilon_{rec}, \tag{2.18}$$

*where:*

$$c_1 = 4\frac{\sqrt{1+\delta_{2k}}}{1-(1+\sqrt{2})\delta_{2k}}. \tag{2.19}$$

## 2.4 Compressed Sensing in Sensor Networks

Recently, compressed sensing concepts have been used in data gathering for sensor networks. In this section, a review of different works in which compressed sensing is used in sensor networks is presented.

### Source Coding

In a single-hop scenario, compressive wireless sensing was proposed to reduce the delay in data gathering [49]. Specifically, the additive nature of the physical environment was used to provide linear combinations of sensed messages over the air. Such random linear combinations are then used to recover the correlated sensed messages. In [50], the authors have discussed the potentials of compressed sensing in a multi-hop network scenario. However, no practical scheme was reported to address the practical issues in multi-hop scenarios,

including encoding procedure. In [51], *single hop* transmission of correlated traffic data to a single base station was done using compressed sensing encoding and sparse recovery.

In [52], compressed sensing was used for data gathering of inter-node correlated messages in a multi-hop network. Specifically, the traditional routing based data gathering was modified to efficiently compress the messages at intermediate nodes. Although they offer a practical data aggregation method, their tree based data gathering has disadvantages in terms of not being robust to link errors. Resilience to link errors can be achieved by using network coding where multi-path transmission is embedded during the transmission.

### Network Coding and CS

Using compressed sensing and sparse recovery concepts for approximate decoding of network coded messages has recently drawn attention. Network coding is used for robust transmission in a multi-hop network scenario where the information flows via multiple paths. As it was discussed in the previous sections, conventional network coding requires at least the same number of received packets as the number of network coded messages, to recover the messages. However, using sparse recovery methods, we may be able to provide an approximate version of the messages even with fewer number of measurements. This fact has been studied and discussed in a few recent works.

In [53, 54], the authors have proposed a practical design for compression of inter-node correlated messages by using (finite field) random linear network coding of packets received at intermediate nodes. The random linear network coded packets are then approximately decoded using belief propagation in the finite field. Although their work is motivated from the Bayesian compressed sensing [55], theoretical feasibility of robust recovery has not been discussed in their work.[9]

In [56], Feizi *et al.* study the theoretical bounds on the achievable rates when a compressed sensing based $\ell_1$-min decoder is used for random linear network coded messages. Explicitly, they compare the two cases of using a maximum a-posteriori probability decoder (so called $\alpha$-decoder in [11])and using an $\ell_1$-min decoder for random linear network coded messages. Their results show that using an $\ell_1$-min decoder requires higher network capacity (links with higher capacities) than the case where maximum a-posteriori probability decoder is used. Further, in [57], Feizi *et al.* have extended random mixing solution and

[9]This may be because of the mathematical complexity of Bayesian compressed sensing and is the subject of study in recent works.

proposed joint source-channel-network coding for non-adaptive transmission of temporal and inter-node correlated messages over additive white Gaussian noise channels. A similar approach has been taken in [58] for joint source-network coding and decreasing the energy consumption of the network. Their work focused on formulation of network utility maximization problem.

In [59], the authors propose a practical design for network coding to take advantage of compressed sensing concepts. Specifically, they discuss the choice of network coding coefficients and the format of packets used in the network.

The main shortcoming in the aforementioned works is the lack of discussion on theoretical feasibility of sparse recovery with network graph constraints. Explicitly, the resulting global network coding equations, resulting from the random linear network coding (in either finite or real fields), are structured by the network deployment. Hence, conventional results in the compressed sensing can not be directly applied for such global network coding equation, making it difficult to study the theoretical feasibility of sparse recovery for random linear network coded messages.

## 2.5 Summary and Open Issues

In this chapter, we provided a review of different data gathering methods, including packet forwarding and data aggregation. Network coding was also discussed both in theory and practice as an alternative for routing based packet forwarding. As the theoretical basis to this thesis, we have presented a few important results in compressed sensing theory. Finally, we have presented a review of work in which compressed sensing is used in sensor networks.

Sparse recovery was previously used for decoding of random linear network coded sources both in real and finite fields. The works on real field have studied the single hop transmission in the context of analog (or physical layer) network coding [30, 49, 51]. Some other works have dealt with classical finite field linear network coding and used belief propagation for decoding of network coded messages [53, 54]. In [30], real network codes have been used for multi-hop transmission of temporal correlated messages of a single source node. Feizi *et al.* have linearly mixed values temporally and spatially to perform joint source-channel-network coding [56, 57].

Although sparse recovery has been used for approximate decoding of random linear

network coded sources, there are still questions to answer and issues to investigate. Random linear mixing of messages can be done over the physical medium when all the nodes are in a *single-hop proximity* [31]. For such case, compressed sensing can be directly applied and the decoding error is bounded using the noisy CS theory (Theorem 2.3.2). In the case where multi hops have to be employed for transmission, finite field random linear network coding may be applicable, for which the classical compressed sensing does not provide any results. This issue can be solved by incorporating quantization and real network codes to be able to use the classical compressed sensing theory. In this thesis, we call this approach *quantized network coding* and formulate it in the following chapter.

Although robust sparse recovery is studied for compressed sensing, it has not been studied for network coded messages. Specifically, as discussed in Section 2.3, robust recovery is guaranteed by using the RIP condition. The total measurement matrix is structured by the network coding coefficients and network deployment. Generally, it is not straightforward to apply CS concepts (such as RIP) to these global network coding equations. This has not been addressed in any of the previous works. In this thesis, we study the theoretical feasibility of robust sparse recovery when quantized network coding is used for transmission of inter-node correlated messages.

# Chapter 3

# Quantized Network Coding in Lossless Networks

Performing linear network coding in the real instead of finite field was previously discussed in [30,31]. The analog (physical layer [33]) network coding may be considered as a version of real network coding in which the media performs linear combinations on the electromagnetic signals. Although such scenario seems to have many advantages in an abstract study of lossless links, it can not be easily adopted in the current state of the art standards. Explicitly, the nonlinear characteristic of modulators and (MAC and PHY) packeting makes it unfeasible to have a linear end-to-end[1] for physical layer network coding.

Since our work is motivated by the concepts of compressed sensing, in which the results are valid in the infinite field of real number, we have to use a real field alternative for conventional finite field network coding [11]. In this thesis, we explicitly define and formulate a version of real network coding in which the inter-node communication link considerations are taken into account. Specifically, the finite capacity of the links restricts us to have a finite number of possible symbols to transmit between the nodes. This is addressed by performing quantization on the network coded contents before transmitting them over the inter-node links (edges). Our approach is called *Quantized Network Coding* (QNC) and is described in the following sections.

[1] over the communication links

## 3.1 Definition and Formulation

In this section, we assume that the communication links between the nodes are modeled as described in Section 2.1. Specifically, as shown in Fig. 2.3, the values at the sender and receiver sides of a link are from a discrete (finite) alphabet. If the link is lossy then the received value may be different from the sent value over the link. The communication of a value over the link may involve packeting and modulation. For simplicity, we do not consider the exact functionality of different communication layers. Explicitly, we represent the values at the sender and receiver sides of the link are represented by $Y_{e,tx}(t)$ and $Y_{e,rx}(t)$, as discussed in Section 2.1.

### 3.1.1 Principle

We define Quantized Network Coding (QNC) at node $v$ according to:

$$Y_{e,tx}(t) = \mathbf{Q}_e\Big(\sum_{e' \in In(v)} \beta_{e,e'}(t) \cdot Y_{e',rx}(t-1) + \alpha_{e,v}(t) \cdot X_v\Big), \tag{3.1}$$

for $t > 1$ where the initial rest condition (2.3) is also satisfied.[2] This means that, at time $t$, the message on any outgoing edge of a node is made up of a quantized linear combination of the messages received by the node at the previous time instant and the information $X_v$ measured by the node. The messages, $X_v$'s are assumed to be constant until the transmission is complete, which is why $X_v$'s are independent of $t$. The local network coding coefficients, $\beta_{e,e'}(t)$'s and $\alpha_{e,v}(t)$'s are real valued, and the determination of their value will be discussed in Section 3.2. The quantizer operator, $\mathbf{Q}_e(\centerdot)$, corresponding to outgoing edge $e$, is designed based on the values of link capacity $C_e$ and packet size $L$, and the distribution of its input (*i.e.* random linear combinations). A simple diagram of QNC at node $v$ is shown in Fig. 3.1. The linear combination in the argument of the quantizer in (3.2) is denoted by $Y_{e,\mathrm{RNC}}(t)$:

$$Y_{e,\mathrm{RNC}}(t) = \sum_{e' \in In(v)} \beta_{e,e'}(t) \cdot Y_{e',rx}(t-1) + \alpha_{e,v}(t) \cdot X_v. \tag{3.2}$$

[2]Essentially, after a batch is ready at $t = 1$, its transmission is started such that the first transmitted packets are available at their receiver nodes at $t = 2$ and so on.

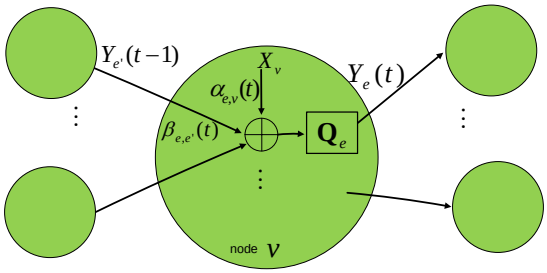



**Figure 3.1** A simple diagram of quantized network coding.

### 3.1.2 End-to-end equations

Denoting the quantization noise of $\mathbf{Q}_e(.)$ at time $t$ by $N_e(t)$,[3]

$$N_e(t) = Y_{e,\mathrm{RNC}}(t) - \mathbf{Q}_e\Big(Y_{e,\mathrm{RNC}}(t)\Big), \tag{3.3}$$

we can reformulate (3.2) as follows:

$$Y_{e,tx}(t) = \sum_{e' \in In(v)} \beta_{e,e'}(t) \cdot Y_{e',rx}(t-1) + \alpha_{e,v}(t) \cdot X_v + N_e(t). \tag{3.4}$$

In this chapter, we assume that the network is lossless,

$$Y_{e,rx}(t) = Y_{e,tx}(t), \ \forall e \in \mathcal{E}, \ \forall t. \tag{3.5}$$

We will study the lossy networks in Chapter 5.

We define the adjacency matrix, $[F(t)]_{|\mathcal{E}|\times|\mathcal{E}|}$, as well as matrix $[A(t)]_{|\mathcal{E}|\times n}$, as:

$$\{F(t)\}_{e,e'} = \begin{cases} \beta_{e,e'}(t) & , \ tail(e) = head(e') \\ 0 & , \ \text{otherwise} \end{cases}, \tag{3.6}$$

$$\{A(t)\}_{e,v} = \begin{cases} \alpha_{e,v}(t) & , \ tail(e) = v \\ 0 & , \ \text{otherwise} \end{cases}. \tag{3.7}$$

[3] We do not make any assumptions about its characteristics.

We also define the vectors of received edge contents, $\underline{Y}_{rx}(t)$, and noises, $\underline{N}(t)$, according to:

$$\underline{Y}_{rx}(t) = [Y_{e,rx}(t) : e \in \mathcal{E}], \tag{3.8}$$
$$\underline{N}(t) = [N_e(t) : e \in \mathcal{E}], \tag{3.9}$$
$$\tag{3.10}$$

As a result, (3.4) can be re-written in the following form:

$$\underline{Y}_{rx}(t) = F(t) \cdot \underline{Y}_{rx}(t-1) + A(t) \cdot \underline{X} + \underline{N}(t). \tag{3.11}$$

Depending on the network deployment, matrix $[B]_{|In(v_0)|\times|\mathcal{E}|}$ defines the relation between the received content of edges, $\underline{Y}_{rx}(t)$, and the received packets at the decoder node $v_0$. Explicitly, we define the vector of received packets at time $t$ at the decoder:

$$\underline{Z}(t) = [Y_{e,rx}(t) : e \in In(v_0)] = B \cdot \underline{Y}_{rx}(t), \tag{3.12}$$

where:

$$\{B\}_{i,e} = \begin{cases} 1 & , \ i \text{ corresponds to } e \in In(v_0) \\ 0 & , \ \text{otherwise} \end{cases}. \tag{3.13}$$

By considering (3.11) as the difference equation, characterizing a linear system with $\underline{X}$, $\underline{N}(t)$'s as its inputs, and $\underline{Z}(t)$ its output, and using the results for linear difference equations [60], one gets:

$$\underline{Z}(t) = \Psi(t) \cdot \underline{X} + \underline{N}_{\text{eff}}(t), \tag{3.14}$$

where the *measurement* matrix, $\Psi(t)$, and the *effective noise* vectors, $\underline{N}_{\text{eff}}(t)$, are calculated as follows:

$$\Psi(t) = B \cdot \sum_{t'=2}^{t} F_{\text{prod}}(t'+1;t)\ A(t'), \tag{3.15}$$
$$\underline{N}_{\text{eff}}(t) = B \cdot \sum_{t'=2}^{t} F_{\text{prod}}(t'+1;t)\ \underline{N}(t'), \tag{3.16}$$

In (3.15),(3.16), $F_{\text{prod}}(\centerdot;\centerdot)$ is defined as:

$$F_{\text{prod}}(t_1;t_2) = \begin{cases} F(t_2)\cdot F(t_2-1)\cdots F(t_1) & ,\ t_2 \geq t_1 \\ \mathbf{I}_{|\mathcal{E}|\times|\mathcal{E}|} & ,\ \text{otherwise} \end{cases} \tag{3.17}$$

and $\mathbf{I}$ denotes the identity matrix.

By storing $\underline{Z}(t)$'s, at the decoder, we build up the *total measurements vector*, $\underline{Z}_{\text{tot}}(t)$, as follows:

$$\underline{Z}_{\text{tot}}(t) = \begin{bmatrix} \underline{Z}(2) \\ \vdots \\ \underline{Z}(t) \end{bmatrix}_{m\times 1}, \tag{3.18}$$

where $m = (t-1)|In(v_0)|$. Therefore, the following can be established:

$$\underline{Z}_{\text{tot}}(t) = \Psi_{\text{tot}}(t)\cdot\underline{X} + \underline{N}_{\text{eff,tot}}(t), \tag{3.19}$$

where the $m \times n$ *total measurement matrix*, $\Psi_{\text{tot}}(t)$, and the *total effective noise* vector, $\underline{N}_{\text{eff,tot}}(t)$, are the concatenation result of measurement matrices, $\Psi(t)$'s, and the effective noise vectors, $\underline{N}_{\text{eff}}(t)$. Because of our assumption of a transmission starting from $t = 1$, measurements in $\underline{Z}(1)$ are not useful for decoding, and therefore:

$$\Psi_{\text{tot}}(t) = \begin{bmatrix} \Psi(2) \\ \vdots \\ \Psi(t) \end{bmatrix}, \tag{3.20}$$

$$\underline{N}_{\text{eff,tot}}(t) = \begin{bmatrix} \underline{N}_{\text{eff}}(2) \\ \vdots \\ \underline{N}_{\text{eff}}(t) \end{bmatrix}. \tag{3.21}$$

In conventional linear network coding, the total number of measurements, $m$ (see (3.18)), is at least equal to the number of data sources, $n$ (the number of nodes in the network here). Typically, the total measurement matrix should be of full column rank, and if there is no uncertainty involved because of measurement or quantization noise, we are able to uniquely find a solution. In this thesis, we are interested in investigating the feasibility of robust recovery of $\underline{X}$, when fewer number of measurements are received at the

decoder than the number of messages; *i.e.* $m < n$. Specifically, to achieve lower transmission delays than conventional linear network coding, we may need to sacrifice the decoding quality.

The characteristic equation (3.19) describing the QNC scenario can be treated as a compressed sensing measurement equation. This gives us an opportunity to apply the results in the literature of compressed sensing and sparse recovery [18, 61] to our QNC scenario with near-sparse messages. However, one needs to examine the required conditions which guarantee sparse recovery in the proposed QNC scenario. In this thesis (specifically in Chapters 3,4), we discuss theoretical and practical feasibility of robust recovery with a compressed sensing perspective.

## 3.2 Network Coding Coefficients

One of the main advantages of the compressed sensing approach is that it relies on a simple model of correlation for the sources; if sparse reconstruction can be applied successfully to recover $\underline{X}$ from equation (3.19) at a given time $t$, this is achieved without requiring the encoders (network nodes) to know much about the underlying signal correlation. This section discusses the design of the linear mixing coefficients $\alpha_{e,v}(t)$ and $\beta_{e,e'}(t)$ and the impact of this design on the ability to apply sparse reconstruction techniques at the sink node $v_0$ to approximately recover the $n$ source signals in $\underline{X}$ from $m$ measurements in $\underline{Z}(t)$ at a given time $t$, where $m \ll n$.

One of the properties that is widely used to characterize appropriate measurement matrices in the compressed sensing literature, is the *Restricted Isometry Property* (RIP) [48]. Roughly speaking, this property provides a measure of norm conservation under dimensionality reduction [62]. In compressed sensing, RIP of the measurement matrix between the sparse domain and the measurement domain allows to draw strong conclusions about the possibility to recover the original data from a small set of measurements [48]. In our case, this means that the RIP should hold for the measurement matrix $\Theta_{\text{tot}}(t) = \Psi_{\text{tot}}(t)\phi$. We have presented the formal definition of RIP in Section 2.3 where a review of compressed sensing was provided (equation (2.14)).

Random matrices with identically and independently distributed (i.i.d) zero mean Gaussian entries are known to be appropriate measurement matrices for compressed sensing. Explicitly, an $m \times n$ i.i.d Gaussian random matrix, denoted $G$, with entries of variance $\frac{1}{m}$,

satisfies RIP of order $k$ and constant $\delta_k$, with a probability exceeding $1 - e^{-\kappa_1 m}$, (called *overwhelming* probability) if $m > \kappa_2 k \log(\frac{n}{k})$, where $\kappa_1$ and $\kappa_2$ only depend on the value of $\delta_k$ (theorem 5.2 in [44]).

Using the results above, it can be understood that an an $m \times n$ i.i.d Gaussian random matrix, $G$, satisfies RIP of order $k$, with a high probability, when the order of the number of measurements, $m$, is $k \log(n/k)$, formally writing:

$$m = O(k \log(\frac{n}{k})), \tag{3.22}$$

which is smaller than the order of $n$, the size of the data [44].

### 3.2.1 QNC design for RIP

We now turn to the design of QNC coefficients in equation (3.2) so that the overall design satisfies RIP with high probability. We assemble here several results from the literature and additional simulations to motivate the proposed design.

In the following, we propose a design for local network coding coefficients, $\beta_{e,e'}(t)$'s and $\alpha_{e,v}(t)$'s, which results in an appropriate total measurement matrix, $\Psi_{\text{tot}}(t)$, in the compressed sensing framework.

**Theorem 3.2.1** *Consider a quantized network coding scenario with lossless links as described in Section 3.1, in which the network coding coefficients, $\alpha_{e,v}(t)$ and $\beta_{e,e'}(t)$, are such that:*

- $\alpha_{e,v}(t) = 0, \ \forall t > 2$,
- $\alpha_{e,v}(2)$*'s are independent zero mean Gaussian random variables,*
- $\beta_{e,e'}(t)$*'s are deterministic.*

*For such a scenario, the entries of the resulting $\Psi_{\text{tot}}(t)$ are zero mean Gaussian random variables. Further, the entries of different columns of $\Psi_{\text{tot}}(t)$ are mutually independent.* ■

**Proof** By choosing $\alpha_{e,v}(t) = 0, \ \forall \ t > 2$, we have:

$$\Psi(t) = B(t) \cdot F(t) \cdots F(3) \cdot A(2), \tag{3.23}$$

which implies that each entry of $\Psi(t)$'s and also $\Psi_{tot}(t)$ is a linear combination of entries of $A(2)$. Moreover, since entries of $A(2)$ are zero mean Gaussian random variables, then the entries of $\Psi(t)$'s and also $\Psi_{tot}(t)$ are zero mean Gaussian random variables. Since entries in different columns of $\Psi_{tot}(t)$, are linear combinations of two independent sets of random variables, *i.e.* entries of $A(2)$, then they are also independent. However, such conclusion can not be made for entries of the same column of $\Psi_{tot}(t)$. ■

Furthermore, we have also numerically concluded that a locally orthogonal set of $\beta_{e,e'}(t)$'s is a better choice than non-orthogonal sets.[4] This choice of coefficients is defined, for each node $v$ and for all $e, e' \in Out(v)$, as:

$$\begin{aligned} \sum_{e'' \in In(v)} \beta_{e,e''}(t) \cdot \beta_{e',e''}(t) &= 0, \ e \neq e', \\ \sum_{e' \in In(v)} \beta^2_{e,e'}(t) &= \frac{1}{|In(v)|^2}. \end{aligned} \tag{3.24}$$

In cases where the number of outgoing edges is greater than the number of incoming edges, *i.e.* $|Out(v)| > |In(v)|$, some of the outgoing edges are randomly selected for removal (not used for transmission) to ensure that the generated $\beta_{e,e'}(t)$'s are locally orthogonal. Furthermore, the second equation in (3.24) is a coefficient normalization which has no specific impact at this stage of the analysis, but which will be important in the study of bounds on sparse recovery performance in Section 3.3. Heuristically, such choice of orthogonal set makes each outgoing packets (of each node) contain innovation.

### 3.2.2 RIP Analysis and Tail Probability of $\ell_2$ norms

Satisfaction of RIP for random matrices is usually characterized by its probability (or its lower probability bounds) [44]. Moreover, to approach the probabilistic satisfaction of RIP, we first need to derive an expression for the tail probability of $\ell_2$ norms [19,44]. Specifically, a well behaved $\Psi_{tot}(t)$ (*i.e.* $\Psi_{tot}(t)$ with high probability of satisfying RIP) should be such that

$$\mathbf{P}\Big(\Big| \|\Psi_{tot}(t) \cdot \underline{x}\|^2_{\ell_2} - 1 \Big| \geq \epsilon\Big) \tag{3.25}$$

[4] This choice reduces the tail probabilities defined later on in equation (3.39), and, as such, increases the probability of the measurement matrix satisfying RIP.

is very small, for all $\underline{x}$, with $\|\underline{x}\|_{\ell_2} = 1$. In the following, we calculate this tail probability for our QNC scenario with the proposed network coding coefficients, and then present a theorem which explicitly describes the relation between the satisfaction of RIP and tail probability of (3.25). In the rest of this section, we assume that the conditions of theorem 3.2.1 hold.

Consider

$$\underline{z}' = \Psi_{tot}(t) \cdot \underline{x}, \tag{3.26}$$

where $\underline{x} \in \mathbb{R}^n$, and $\|\underline{x}\|_{\ell_2} = 1$. Since the conditions of theorem 3.2.1 are satisfied, Eq. 3.23 holds, and therefore:

$$\Psi_{tot}(t) = \begin{bmatrix} \Psi(2) \\ \vdots \\ \Psi(t) \end{bmatrix} = \Omega(t) \cdot A(2), \tag{3.27}$$

where

$$\Omega(t) = \begin{bmatrix} B(2) \\ B(3)F(3) \\ \vdots \\ B(t)F(t)\cdots F(3) \end{bmatrix}. \tag{3.28}$$

This implies:

$$\underline{z}' = \Omega(t)\ A(2) \cdot \underline{x}, \tag{3.29}$$

or equivalently:

$$\begin{aligned} z'_i &= \sum_{v=1}^{n} \{\Psi_{tot}(t)\}_{iv}\ x_v \\ &= \sum_{v=1}^{n} \sum_{e=1}^{|\mathcal{E}|} \{\Omega(t)\}_{ie}\ \{A(2)\}_{ev}\ x_v. \end{aligned} \tag{3.30}$$

By expanding $z_i'^2$, and using the fact that $\{A(2)\}_{ev}$ is non-zero only when $tail(e) = v$, we

have:

$$\begin{aligned} \|\underline{z}'\|_{\ell_2}^2 &= \sum_{i=1}^{m} z_i'^2 \qquad (3.31)\\ &= \sum_{e=1}^{|\mathcal{E}|}\sum_{e'=1}^{|\mathcal{E}|} \gamma_{e,e'}(\underline{x})\{A(2)\}_{e,tail(e)}\{A(2)\}_{e',tail(e')}, \end{aligned}$$

where:

$$\gamma_{e,e'}(\underline{x}) = \sum_{i=1}^{m}\{\Omega(t)\}_{ie}\ \{\Omega(t)\}_{ie'} \cdot x_{tail(e)}\ x_{tail(e')} \qquad (3.32)$$

Using eigenvalue decomposition, (3.31) simplifies to:

$$\|\underline{z}'\|_{\ell_2}^2 = \sum_{e=1}^{|\mathcal{E}|} \lambda_e(\underline{x}) \cdot \chi_e^2, \qquad (3.33)$$

where $\lambda_e(\underline{x})$'s are eigenvalues of the symmetric matrix

$$\Gamma(\underline{x}) = [\gamma_{e,e'}(\underline{x})]_{|\mathcal{E}|\times|\mathcal{E}|}, \qquad (3.34)$$

and $\chi_e^2$'s are independent Chi-Square random variables of first order. Moreover, for the characteristic function of $\|\underline{z}'\|_{\ell_2}^2$, we have:

$$\begin{aligned} \mathbf{E}[e^{j\omega\|\underline{z}'\|_{\ell_2}^2}] &= \mathbf{E}[e^{j\omega\sum_{e=1}^{|\mathcal{E}|}\lambda_e(\underline{x})\chi_e^2}] \qquad (3.35)\\ &= \prod_{e=1}^{|\mathcal{E}|}\mathbf{E}[e^{j\omega\lambda_e(\underline{x})\chi_e^2}] \qquad (3.36)\\ &= \prod_{e=1}^{|\mathcal{E}|}\frac{1}{\sqrt{1-j2\omega\lambda_e(\underline{x})}}, \qquad (3.37) \end{aligned}$$

where (3.36) is derived from independence of $\chi_e^2$'s. By using the inverse formula of characteristic function, Eqs. 3.38-3.39 can be obtained, where $\mathbf{p}_{\|\underline{z}'\|_{\ell_2}^2}(\centerdot)$ is the probability density

function of $\|\underline{z}'\|_{\ell_2}^2$, and (3.38) is resulted from the integral property of Fourier transform.

$$\begin{aligned}\mathbf{P}\Big(\Big|\|\underline{z}'\|_{\ell_2}^2-1\Big|>\epsilon\Big) &= 1+\int_{-\infty}^{1-\epsilon}\mathbf{p}_{\|\underline{z}'\|_{\ell_2}^2}(\nu)d\nu-\int_{-\infty}^{1+\epsilon}\mathbf{p}_{\|\underline{z}'\|_{\ell_2}^2}(\nu)d\nu\\ &= 1+\frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{\mathbf{E}[e^{j\omega\|\underline{z}'\|_{\ell_2}^2}]}{-j\omega}e^{-j\omega(1-\epsilon)}d\omega\\ &\quad-\frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{\mathbf{E}[e^{j\omega\|\underline{z}'\|_{\ell_2}^2}]}{-j\omega}e^{-j\omega(1+\epsilon)}d\omega\\ &= 1-\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{e^{-j\omega}\sin(\epsilon\omega)}{\omega\prod_{e=1}^{|\mathcal{E}|}\sqrt{1-j2\omega\lambda_e(\underline{x})}}d\omega, \qquad (3.38)\end{aligned}$$

The right hand side of (3.38) is the expression for the tail probability of $\ell_2$-norms, for a specific $\underline{x}$, resulting from our proposed network coding coefficients. Explicitly, we define the *tail probability* of a measurement matrix according to:

$$\begin{aligned}\mathbf{p}_{\text{tail}}(\Psi_{\text{tot}}(t),\varepsilon) &= \max_{\underline{x}'}\mathbf{P}\Big(\Big|\|\Psi_{\text{tot}}(t)\underline{x}'\|_{\ell_2}^2-1\Big|>\varepsilon\Big),\\ &\quad\text{subject to: } \|\underline{x}'\|_{\ell_2}=1 \qquad (3.39)\end{aligned}$$

We calculate the worst case tail probability, corresponding to an i.i.d. Gaussian matrix, and compare it with that of our $\Psi_{tot}(t)$. For an $m\times n$ i.i.d. Gaussian matrix, $G_{m\times n}$, the worst case tail probability, $\mathbf{p}_{tail}(G_{m\times n},\frac{\delta_k}{\sqrt{2}})$, can be calculated as:[5]

$$\mathbf{p}_{tail}(G_{m\times n},\frac{\delta_k}{\sqrt{2}})=1-\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{e^{-j\omega}\sin(\omega\frac{\delta_k}{\sqrt{2}})}{\omega\ (1-2j\frac{\omega}{m})^{m/2}}d\omega. \qquad (3.40)$$

Further, (3.41) directly implies:

$$\mathbf{p}_{tail}(\Psi_{\text{tot}}(t),\frac{\delta_k}{\sqrt{2}})=1-\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{e^{-j\omega}\sin(\omega\frac{\delta_k}{\sqrt{2}})}{\omega\prod_{e=1}^{|\mathcal{E}|}\sqrt{1-j2\omega\lambda_e(\underline{x})}}d\omega. \qquad (3.41)$$

In order to carry out the comparison, we generate random deployments of networks and calculating the worst case tail probability of (3.38) and (3.40) in each generated deployment. The resulting tail probabilities, and corresponding number of measurements are then

[5]This can be obtained similar to the reasoning procedure for (3.38).

averaged over different realizations of network deployments. In Fig. 3.2, $\mathbf{p}_{tail}(\centerdot,\centerdot)$ is drawn versus the number of measurements $m$ in logarithmic scale, for $\Psi_{tot}(t)$ (QNC) and $G_{m\times n}$ (Gaussian), and different values of RIP constant, $\delta_k$ ($\delta_k = 0.41421 \simeq \sqrt{2}-1$ is the largest RIP constant for which Theorem 2.3.1 can be applied). As it is shown in Figs. 3.2(a)-3.2(c), the number of edges is changed for our numerical evaluation of tail probabilities. Moreover, the network coding coefficients, $\alpha_{e,v}(t)$'s and $\beta_{e,e'}(t)$'s, are generated as described in Section 3.2.1.

Now, we establish the relation between the satisfaction of RIP and the tail probability $\mathbf{p}_{\text{tail}}(\Psi_{\text{tot}}(t),\varepsilon)$, by proving the following theorem.

**Theorem 3.2.2** *Consider $\Psi_{\text{tot}}(t)$ with the tail probability, as defined in (3.39), and an orthonormal transform matrix $\phi$. Then, $\Theta_{\text{tot}}(t) = \Psi_{\text{tot}}(t)\cdot\phi$ satisfies RIP of order $k$ and constant $\delta_k$, with a probability exceeding,*

$$1 - \begin{pmatrix} n \\ k \end{pmatrix} (\frac{42}{\delta_k})^k \ \boldsymbol{p}_{tail}(\Psi_{\text{tot}}(t), \varepsilon = \frac{\delta_k}{\sqrt{2}}). \tag{3.42}$$

**Proof** [6] To prove that RIP holds, we should show that inequality of (2.14) is satisfied, for all $k$-sparse vectors, $\underline{s}_k$. We only need to show it is satisfied, for vectors, with $\|\underline{s}_k\|_{\ell_2} = 1$, since $\|\Psi_{\text{tot}}(t)\phi\cdot\underline{s}_k\|_{\ell_2}$ is proportional with $\|\underline{s}_k\|_{\ell_2}$. Now, fix a set $T \subset \{1,2,\ldots,n\}$, with $|T| = k$, and let $\Gamma_T$ be the subspace of $k$-dimensional vectors, $\underline{s}_T$, spanned by columns of $\Psi_{\text{tot}}(t)$, with indices in $T$. According to lemma 7.5 in [19], we can choose a finite set of vectors, $\underline{w}_T \in \mathcal{W}_T$, where $\mathcal{W}_T \subset \Gamma_T$ and $\|\underline{w}\|_{\ell_2} \leq 1$, such that for all $\underline{s}_T \in \Gamma_T$, with $\|\underline{s}_T\|_{\ell_2} \leq 1$, we have:

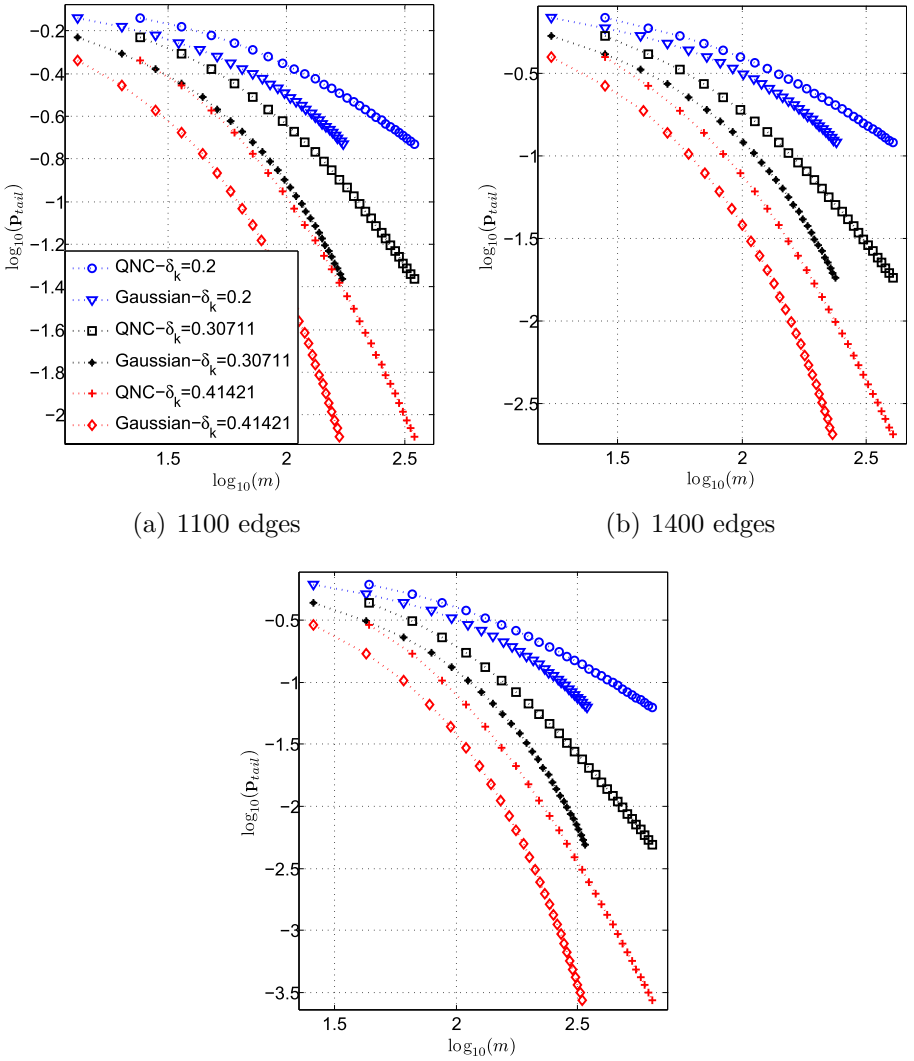
$$\|\underline{s}_T - \underline{w}_T\|_{\ell_2} \leq \frac{\delta_k}{14}, \tag{3.43}$$

conditioned on:

$$|\Gamma_T| \leq (\frac{42}{\delta_k})^k. \tag{3.44}$$

There are $\begin{pmatrix} n \\ k \end{pmatrix}$ different $T$'s, for which we repeat the above procedure and obtain: $\mathcal{W} = \bigcup_T \mathcal{W}_T$. By using the union bound and the fact that for every $\underline{x} = \phi\cdot\underline{w}$, where $\underline{w} \in \mathcal{W}$,

[6] Most of the proof is similar to the proof of theorem 7.3 in [19].




(a) 1100 edges

(b) 1400 edges

(c) 1800 edges

**Figure 3.2** Logarithmic tail probability versus logarithmic ratio of minimum required number of measurements in our QNC scenario and i.i.d. Gaussian measurement matrices, for $n = 100$ nodes, different RIP constants, and different number of edges. The nodes are deployed randomly and uniformly in a square two-dimensional region. The connectivity of the nodes is modeled by using an exponential power decay. Network coding coefficients are picked according to the design suggested in this section, *i.e.* Theorem 3.2.1 and (3.24). Such set of settings are the same as the one used in the simulations section (Section 3.4).

the RIP inequality (2.14) implies:

$$\mathbf{P}\Big(\Big|\|\Psi_{\text{tot}}(t)\cdot\underline{x}\|_{\ell_2}^2-1\Big|\geq\epsilon\Big)\leq\mathbf{p}_{tail}(\Psi_{\text{tot}}(t),\epsilon). \tag{3.45}$$

Therefore, for every $\underline{w}\in\mathcal{W}$, the inequality

$$(1-\frac{\delta_k}{\sqrt{2}})\|\underline{w}\|_{\ell_2}^2\leq\|\Psi_{\text{tot}}(t)\phi\cdot\underline{w}\|_{\ell_2}^2\leq(1+\frac{\delta_k}{\sqrt{2}})\|\underline{w}\|_{\ell_2}^2, \tag{3.46}$$

holds with a probability exceeding

$$1-\begin{pmatrix}n\\k\end{pmatrix}(\frac{42}{\delta_k})^k\ \mathbf{p}_{tail}(\Psi_{\text{tot}}(t),\frac{\delta_k}{\sqrt{2}}). \tag{3.47}$$

By simple algebra and using the properties of $\ell_2$ norm, the proof can be completed in a few steps. These steps are the same as the last part of the proof of theorem 7.3 in [19]. ■

In (3.38), we derived a detailed expression of the tail probability (3.39). Our ultimate goal would be to use this expression to directly conclude that the number of necessary measurements $m$ in the QNC scenario is of the same order as that of a well-known Gaussian measurement matrix, as defined above. However, the relationship between the network and QNC parameters on one hand and the measurement matrix $\Psi_{\text{tot}}(t)$ on the other hand is too complicated to easily draw conclusions (see equations (3.6), (3.7), (3.15)). We therefore resort to the following reasoning: we have first shown through simulations that the tail probabilities for the QNC and Gaussian measurements matrices are of the same order (Fig. 3.2); we then conclude to a similar behavior of QNC and Gaussian measurement matrices in terms of RIP satisfaction, and thus in terms of required number of measurements.

A different set of conditions are also used to discuss the feasibility of sparse recovery. One of such conditions which have recently drawn attention are based on the restricted eigenvalue condition of random matrices [63–65]. In general, restricted eigenvalue conditions are weaker than RIP to be held by a random matrix which makes them easier to apply to other cases. In Appendix A, we discuss restricted eigenvalue conditions. Further, motivated by a specific version of restricted eigenvalue condition, discussed in [65], we suggest a design for network coding coefficients. Then, we derive a mathematical bound for the $\ell_2$ norm of decoding error when such design is used for QNC scenario.

## 3.3 $\ell_1$-min Decoding

In this section, we discuss decoding of quantized network coded messages using the classical compressed sensing based decoding, *i.e.* $\ell_1$-min decoding. Specifically, considering the linear measurement equation (3.19) and the QNC design proposed in Section 3.2, we present an upper bound on the decoding error of $\ell_1$ min decoding.

It is well-known that recovery of exactly sparse vectors from an under-determined set of linear measurements can be done with no error, using linear programming [46,61]. However, when dealing with noisy measurements, $\ell_1$-min recovery does not necessarily offer a minimum mean squared error solution. There is still a lot of work being done to develop practical and near minimum mean squared error decoding algorithms for noisy cases.

Recovery of sparse vectors from quantized measurements have been recently studied in a number of works [66–68]. For instance, the authors in [67] consider the estimation problem of sparse vectors from measurements that are quantized and corrupted by Gaussian noise. The main aspect that differentiates our model from that in [67] is that in our QNC scenario the resulting effective total measurement noises are non-linear functions of quantization noises and link distortions at each link.

Along the lines of [18, 48], the compressed sensing based decoder for the QNC scenario solves the following convex optimization:

$$\begin{aligned} \underline{\hat{X}}(t) &= \phi \cdot \arg\min_{\underline{S}'} ||\underline{S}'||_{\ell_1}, \\ &\quad \text{subject to: } \|\underline{Z}_{\text{tot}}(t) - \Psi_{\text{tot}}(t)\ \phi\ \underline{S}'\|_{\ell_2} \leq \epsilon_{rec}(t) \end{aligned} \tag{3.48}$$

which can be solved by using linear programming [46]. In (3.48), $\epsilon_{\text{rec}}(t)$ is an upper bound on the $\ell_2$ norm of sum of measurement noise vectors.

The $\ell_1$-min decoding algorithm in (3.48) needs the knowledge of $\Psi_{\text{tot}}(t)$ and $\phi$ matrices. As the local network coding coefficients are generated in a pseudo-random way, the matrix $\Psi_{\text{tot}}(t)$ can be calculated by having the knowledge of the network structure (connectivity of the nodes). Further, the knowledge of the sparsifying matrix $\phi$ can be obtained at the decoder by calculating the Karhunen Loeve Transform on a number of previously received generations of messages. The decoder does not need to know the sparsity factor $k/n$ of messages as it finds the vector with the minimum $\ell_1$ norm which fits the measurement equation (right side in (3.48)).

The following theorems present our results on the recovery error using $\ell_1$-min decoding of (3.48).

**Theorem 3.3.1** *Consider the QNC scenario where the absolute value of messages is bounded by $q_{max}$ and the local network coding coefficients are such that:*

- $\alpha_{e,v}(t) = 0,\ \forall t > 2$,
- $\alpha_{e,v}(2)$*'s are independent zero mean Gaussian random variables with variance* $\sigma_0^2$,
- $\beta_{e,e'}(t)$*'s are deterministic and locally orthogonal according to (3.24).*

*In such scenario, overflowing of linear combinations $Y_{e.\mathrm{RNC}}(t)$ in (3.2) (over the limit $q_{max}$) within the nodes happens with a probability less than or equal to:*

$$2|\mathcal{E}|Q(\sigma_0^{-1}) \ , \tag{3.49}$$

*where $Q(.)$ is the tail probability of standard normal distributions (i.e. one-sided Q-function).*

**Proof** Using Cauchy-Schwartz inequality, for $t \geq 3$, we have:

$$\sum_{e' \in In(v)} |\beta_{e,e'}(t)| \ \leq \ \sqrt{(\sum_{e' \in In(v)} |\beta_{e,e'}(t)|^2)(\sum_{e' \in In(v)} 1)^2} \tag{3.50}$$

$$= \ \sqrt{(\frac{1}{|In(v)|^2})(|In(v)|)^2} = 1 \tag{3.51}$$

As a result, since $\alpha_{e,v}(t)$'s are zero for $t \geq 3$, it is straightforward to imply that over flow may not happen for $t \geq 3$.

For $t = 2$, since only the node message $X_v$ is available at each node, the values of $\beta_{e,e'}(2)$'s does not affect anything. Hence, only the value of $\alpha_{e,v}(2)$ can result in overflow and therefore $|\alpha_{e,v}(2)|$ should be less than or equal to one to prevent overflow. Moreover, because of the Gaussian distribution of $\alpha_{e,v}(2)$'s, each $\alpha_{e,v}(2)$ may have an absolute value more than one, with a probability of $2Q(\sigma_0^{-1})$. Therefore, using the union bound, the probability that there is at least one $\alpha_{e,v}(2)$ with $|\alpha_{e,v}(2)| > 1$ is upper bounded by $2|\mathcal{E}|Q(\sigma_0^{-1})$. ■

**Theorem 3.3.2** *Consider a QNC scenario where, for all $v \in \mathcal{V}$, the network coding coefficients satisfy the conditions in theorem 3.3.1, and for which, based on the discussion in*

*Section 3.2.2, the measurement matrix* $\Theta_{\text{tot}}(t) = \Psi_{\text{tot}}(t)\phi$ *satisfies RIP of order* $2k$ *with constant* $\delta_{2k} < \sqrt{2} - 1$*. The edge quantizers,* $\boldsymbol{Q}_e(.)$*'s, are assumed to be uniform with step size* $\Delta_e$ *on the range* $[-q_{\max}, +q_{\max}]$*. Further, we assume that the links are lossless with a limited capacity* $C_e$*. Then, with a probability exceeding*

$$1 - 2|\mathcal{E}|Q(\sigma_0^{-1}), \tag{3.52}$$

*for the* $\ell_1$*-min decoding of (3.48), we have:*

$$\left\|\underline{\hat{X}}(t) - \underline{X}\right\|_{\ell_2} \leq c_1\epsilon_{rec} + \frac{c_2}{\sqrt{2}}\epsilon_k, \tag{3.53}$$

*where* $\epsilon_{rec}^2(t)$ *is defined as*

$$\epsilon_{rec}^2(t) = \frac{1}{4}\sum_{t'=2}^{t}\sum_{e\in In(v_0)}\Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}|\{F_{\text{prod}}(t''+1;t')\}_{e,e'}|\ \Delta_{e'}\Big)^2, \tag{3.54}$$

*and the matrix product* $F_{\text{prod}}(.,.)$ *in (3.54) is defined in (3.17). The constants* $c_1$ *and* $c_2$ *are also defined as follows:*

$$c_1 = 4\frac{\sqrt{1+\delta_{2k}}}{1-(1+\sqrt{2})\delta_{2k}}, \tag{3.55}$$

$$c_2 = 2\frac{1-(1-\sqrt{2})\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}}. \tag{3.56}$$

**Proof** According to theorem 3.3.1, the conditions on the local network coding coefficients ensures that overflow does not happen with a probability exceeding $1-2|\mathcal{E}|Q(\sigma_0^{-1})$. Further, since the network is lossless, the only associated measurement noise is resulting from the quantization noise at the edges. For each uniform quantizer $\mathbf{Q}_e(.)$, $e \in \mathcal{E}$, we have:

$$-\frac{\Delta_e}{2} \leq N_e(t) \leq +\frac{\Delta_e}{2}. \tag{3.57}$$

This implies:

$$\left\|\underline{N}_{\text{eff,tot}}(t)\right\|_{\ell_2}^2 = \sum_{t'=2}^{t}\sum_{i'=1}^{|In(v_0)|}\{\underline{N}_{\text{eff}}(t)\}_{i'}^2$$

$$= \sum_{t'=2}^{t} \sum_{e\in In(v_0)} \{\sum_{t''=2}^{t'} F_{\text{prod}}(t''+1;t')\underline{N}(t'')\}_e^2 \tag{3.58}$$

$$= \sum_{t'=2}^{t} \sum_{e\in In(v_0)} \Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}\{F_{\text{prod}}(t''+1;t')\}_{e,e'}N_{e'}(t'')\Big)^2$$

$$\leq \sum_{t'=2}^{t} \sum_{e\in In(v_0)} \Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}|\{F_{\text{prod}}(t''+1;t')\}_{e,e'}|\ |N_{e'}(t'')|\Big)^2$$

$$\leq \frac{1}{4}\sum_{t'=2}^{t} \sum_{e\in In(v_0)} \Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}|\{F_{\text{prod}}(t''+1;t')\}_{e,e'}|\ \Delta_{e'}\Big)^2$$

$$= \epsilon_{\text{rec}}^2(t), \tag{3.59}$$

where (3.58) holds because of the one-to-one mapping structure of $B$ matrix. This provides an upper bound on the $\ell_2$ norm of measurement noise in our QNC scenario.

According to theorem 4.2 in [19], when the measurement matrix satisfies RIP of appropriate order and constant (as in the assumptions of theorem 3.3.2) and the measurement noise is bounded, $\ell_1$ min recovery can yield an estimate with bounded recovery error. Explicitly, the bound is as in (3.53), considering the near-sparsity model of the messages and the obtained bound on the measurement noise. ■

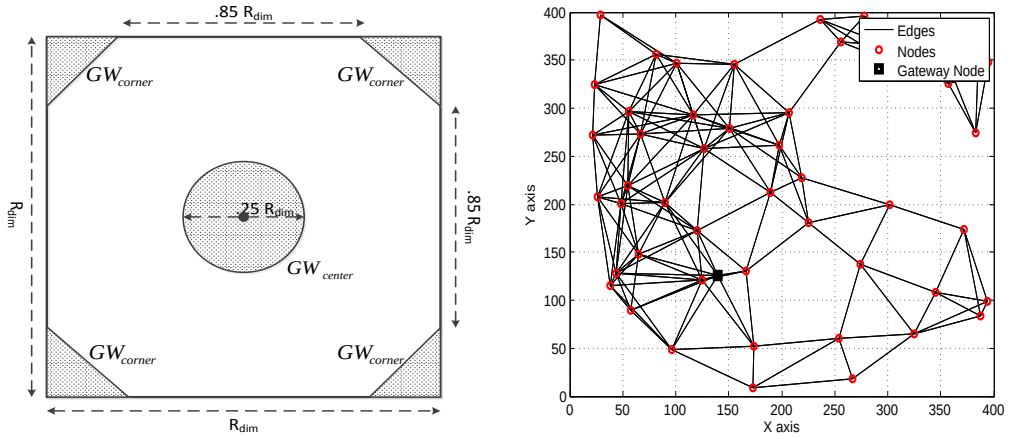
According to the preceding theorem, the upper bound, $c_1\epsilon_{\text{rec}}$, is decreased when the quantization steps, $\Delta_e$'s, are decreased. Since $\Delta_e = 2q_{\max}/2^{\lfloor LC_e \rfloor}$, a smaller upper bound on the $\ell_2$-norm of the recovery error can be obtained by increasing the block length, $L$. Although this can be done practically, it will simultaneously increase the point to point transmission delays in the network, which may not be desirable. This creates a trade-off between reconstruction quality and delay, which will be explored in detail in Section 3.4. Moreover, although a uniform quantizer may not be the best choice for some message distributions, it is still widely used in practice. It also allows us simplify the mathematical analysis to provide a theoretical bound on the resulting recovery error.

As discussed in theorem 3.3.2, the local network coding coefficients, proposed in (3.24), ensure that the normalization is respected and overflow does not happen, with high probability. An appropriate choice of $\sigma_0$ should also be picked for this purpose. For example, when the number of edges is in the order of 1000, selecting $\sigma_0 = 0.25$ would result in a low probability for overflow.

## 3.4 Simulation Results

In this section, we evaluate the performance of quantized network coding, by using different numerical simulations. The main motivation behind the proposed Quantized Network Coding technique is to allow for reconstruction of the correlated source signals at the sink node or decoder with a limited number of measurements. To this end, we will compare delay-distortion curves for different data gathering algorithms. Our performance analysis includes statistical evaluation of proposed QNC scenario versus packet forwarding and conventional finite field network coding schemes. The resulting analysis will provide a comprehensive comparison between these transmission methods for different network deployments and correlation of sources.

### 3.4.1 Network Deployment and Message Generation




(a) Gateway node selection

(b) A deployed network in $GW_{\text{center}}$ mode with $R_{\text{dim}} = 400$

**Figure 3.3** Network deployment and Gateway node selection.

To set up the simulations, we generate random deployments of networks with directed links, obtained from a transmission power loss model. Specifically, a certain number of nodes, $n$, are deployed in a two-dimensional $R_{\text{dim}} \times R_{\text{dim}}$ square region, according to a uniform distribution. One of the deployed nodes in the network is randomly picked to be the gateway node, $v_0$, in which the messages are decoded. In our simulations, we examine two different probability models to pick the gateway node. In the first model, denoted

by $GW_{\text{corner}}$, the gateway node is uniformly picked from the nodes within the region in the corners of the unit square, as shown in Fig. 3.3(a). In the second model, denoted by $GW_{\text{center}}$, the gateway node is uniformly picked from the nodes within the region in the center of unit square, as shown in Fig. 3.3(a). These two models correspond to the two extreme cases for the location of gateway node.

The symmetric connectivity (which is different from full duplex transmission over links) of two nodes is determined according to an exponential power decay model: if the average power of received signal from node $i$ at node $j$, denoted by $\bar{P}_{i\to j}$, is greater than a certain threshold (called receiver sensitivity), then there is an edge from $i$ to $j$ in the graph representing the network. The average power of received signal is also modeled by using the exponential decay model:

$$\bar{P}_{i\to j} = P_{i,\text{TX}} - 10\ c_{\text{att}}\ \log_{10}\Big(dist(i,j)\Big) - c_{\text{att},0}\ \ (\text{dBm})\ , \tag{3.60}$$

where $dist(i,j)$ is the Euclidean distance of $i$ and $j$ nodes in meter and $P_{i,\text{TX}}$ is the transmitted power from node $i$. The constants $c_{\text{att}}$ and $c_{\text{att},0}$ are called the attenuation exponent and the attenuation at one meter of the physical environment. Moreover, both $\bar{P}_{i\to j}$ and $P_{i,\text{TX}}$ are in dBm. The values of these constants and parameters is also presented in Table 3.1.

**Table 3.1** Values of the parameters involved in the power decay model. These values are taken from a physical channel modeling used for ZigBee based simulations, as discussed further in Section 5.1.

| Parameter | Value |
|---|---|
| Attenuation Exponent $c_{\text{att}}$ | 3.12 |
| Attenuation at One Meter $c_{\text{att},0}$ | 35.4 |
| Average TX Power $P_{i,\text{TX}}$ | 2.5 dBm |
| RX Sensitivity | $-95$ dBm |

We change the value of $R_{\text{dim}}$ to generate networks with different number of edges and different maximum hop distances, as described later in this section. Different settings for generating network deployments, the resulting average degree of nodes $|In(v)|$, and the resulting average hop distances of nodes from the gateway node are presented in Table 3.2.

In our simulation, each communication link (edges) can maintain a lossless communication of one bit per use, *i.e.,* $C_e = 1$, for all $e \in \mathcal{E}$. We also assume that there is no interference involved from transmission in other nodes which may have been achieved by

using a time multiplexing strategy. A sample network deployment is shown in Fig. 3.3(b), where the arrows represent the directed links between the nodes.

**Table 3.2** Average Number of hops in each deployment setting with $n = 100$ nodes.

| Deployment Scenarios | Simulation Settings | Average $\lvert In(v)\rvert$ | Average Hops |
|---|---|---|---|
| $\text{SIM}_1$ | $R_{\text{dim}} = 600, GW_{\text{corner}}$ | 5.3 | 9.7 |
| $\text{SIM}_2$ | $R_{\text{dim}} = 600, GW_{\text{center}}$ | 5.3 | 5.3 |
| $\text{SIM}_3$ | $R_{\text{dim}} = 350, GW_{\text{corner}}$ | 13.9 | 3.9 |
| $\text{SIM}_4$ | $R_{\text{dim}} = 350, GW_{\text{center}}$ | 13.9 | 2.3 |
| $\text{SIM}_5$ | $R_{\text{dim}} = 250, GW_{\text{corner}}$ | 24.8 | 2.7 |
| $\text{SIM}_6$ | $R_{\text{dim}} = 250, GW_{\text{center}}$ | 24.8 | 1.7 |

To generate a realization of messages, $\underline{x}$, we first generate a $k$-sparse random vector, $\underline{s}_k$, whose nonzero components are uniformly distributed between $-\frac{1}{2}$ and $+\frac{1}{2}$. Then a near-sparse vector, $\underline{s}$, is obtained such that elements of $(\underline{s} - \underline{s}_k)$ are drawn from independent zero mean uniform random variables and,

$$\frac{||\underline{s} - \underline{s}_k||_{\ell_1}}{||\underline{s}||_{\ell_1}} = \epsilon_k. \tag{3.61}$$

This is followed by generation of an orthonormal random matrix, $\phi$, and calculating random messages: $\underline{x} = \phi \cdot \underline{s}$. To ensure that $x_j$'s are bounded, they are normalized between $-q_{max}$ and $+q_{max}$ ($x_j$'s are multiplied by a constant value). The value of $q_{max}$, used for the simulations does not affect the simulation results, since we are using average SNR as a measure of decoding quality. We study the performance of different transmission scenarios by repeating our simulations for different values of sparsity factor, $\frac{k}{n}$, and near-sparsity parameter, $\epsilon_k$.

The average Signal to Noise Ratio (SNR) is used as the quality measure in our numerical comparisons. Explicitly, for the decoded messages in a scheme, $\hat{\underline{x}}$, the average SNR is defined as:

$$\text{SNR} = 20 \log_{10} \frac{\overline{||\underline{x}||_{\ell_2}}}{\overline{||\underline{x} - \hat{\underline{x}}||_{\ell_2}}}, \tag{3.62}$$

where $\overline{(\cdot)}$ stands for the average over different realizations of network deployments. For

each realization of network deployment, we only generate one realization of messages and therefore taking the average over different network deployments is enough to obtain the average SNR values.

The payback measure in our comparisons is the corresponding average delivery delay, to achieve the required quality of service (average SNR). Explicitly, delivery delay for a transmission which has terminated at $t$ is equal to $(t-1)L$. In the case of packet forwarding, we do not consider the learning period, required to find the routes from each sensor node to the decoder node.

**Table 3.3** The parameters of messages and the networks, used in our simulations.

| Parameter | Value(s) |
|---|---|
| Number of Nodes $n$ | 100 |
| Location Range of Nodes $R_{\text{dim}}$ | $250, 350, 600$ |
| Packet Length $L$ | $1, \ldots, 40$ |
| Sparsity Factor $k/n$ | $0.01, 0.05, 0.10$ |
| Near-sparsity Parameter $\epsilon_k$ | $0, 0.002, 0.02, 0.2$ |

The used simulation parameters are listed in Table 3.3. In the following, we describe the different simulated transmission scenarios:

### Quantized Network Coding (QNC)

For each generated random network deployment, we perform QNC with $\ell_1$-min decoding. Local network coding coefficients, $\alpha_{e,v}(t)$'s and $\beta_{e,e'}(t)$'s, are generated according to the conditions of theorem 3.3.1, where $\sigma_0 = 0.25$. Edge quantizers, $\mathbf{Q}_e(.)$'s, have uniform characteristic with a range of $[-q_{max}, +q_{max}]$ and $2^L$ intervals (since $C_e = 1,\ \forall e$). Random $\alpha_{e,v}(2)$'s and $\beta_{e,e'}(t)$'s can be generated in a pseudo-random way and therefore only the generator seed needs to be transmitted to the decoder in a packet header.
At the decoder, the received measurements up to $t$, $\underline{z}_{\text{tot}}(t)$, are used to recover the original messages. Specifically, for a realization of messages, $\underline{x}$, we define $\underline{\hat{x}}_{\text{QNC}}(t)$, to be the recovered messages, using $\ell_1$-min decoding, according to (3.48). The convex optimization, involved in (3.48) is solved by using the open source implementation of disciplined convex programming [69]. Moreover, the network deployment is assumed to be known at the

decoder in order to build up $\Psi_{\text{tot}}(t)$ matrices (the random generator seed is enough to regenerate local network coding coefficients). Although the exact sparsity of messages, $k$, does not need to be known for performing $\ell_1$-min decoding, the sparsifying transform, $\phi$, should be known. The block length, $L$, has to be known at the decoder to be able to calculate the level of the effective measurement noise, *i.e.* $\epsilon_{\text{rec}}(t)$'s.

### Quantization and Packet Forwarding (QandPF)

For each deployment, we also simulated a routing based packet forwarding and compared it with the results for QNC. To find the routes from each node to the gateway node, we find the shortest path from each node to the gateway node, using the Dijkstra algorithm [70]. Further, the real valued messages, $x_v$'s, are quantized at their corresponding source nodes, by using similar uniform quantizers, as used in QNC transmission. The system delivers all $x_v$'s to the decoder node over a certain period of time and keeps track of delivered messages over time, $t$, in the recovered vector of messages, $\underline{\hat{x}}_{\text{PF}}(t)$. Moreover, if a message, $x_v$, is not delivered by time index $t$, zero is used as its recovered value:

$$\{\underline{\hat{x}}_{\text{PF}}(t)\}_v = 0. \tag{3.63}$$

### Quantization and Packet Forwarding with CS Decoding (QandPFwithCS)

This scenario is exactly the same as QandPF scenario, except at the decoder side. Specifically, at the decoder node, if the messages of some nodes are still not delivered, the decoder tries to recover them from the other received (quantized) messages, using compressed sensing decoding. Explicitly, we define $\Psi_{\text{tot,PF}}(t)$ to be the mapping matrix from the messages to the received quantized messages, *i.e.*:

$$\{\Psi_{\text{tot.PF}}(t)\}_{i,v} = \begin{cases} 1 & , \ \mathbf{Q}(x_v) \text{ delivered by } t \text{ and corresponds to } i\text{'th received packet,} \\ 0 & , \ \mathbf{Q}(x_v) \text{ not delivered by } t \end{cases}, \tag{3.64}$$

and, $\underline{z}_{\text{tot,PF}}(t)$ to be the set of received (delivered via PF) quantized messages at the decoder. In such case, the following $\ell_1$ minimization is solved:

$$\begin{aligned} \underline{\hat{x}}_{\text{PFCS},0}(t) &= \phi \cdot \arg\min_{\underline{s}'} ||\underline{s}'||_{\ell_1}, \\ & \quad s.t.\ \left\|\underline{z}_{\text{tot,PF}}(t) - \Psi_{\text{tot,PF}}(t)\phi\underline{s}'\right\|_{\ell_2} \leq \epsilon_{\text{rec,PF}}(t), \end{aligned} \tag{3.65}$$

where $\epsilon_{\text{rec,PF}}(t)$ is the upper bound on the $\ell_2$ norm of quantized delivered messages.[7] Then, for each $v$ if its quantized messages $\mathbf{Q}(x_v)$ is still not delivered, we use $\{\underline{\hat{x}}_{\text{PFCS},0}(t)\}_v$, meaning:

$$\{\underline{\hat{x}}_{\text{PFCS}}\}_v = \begin{cases} \mathbf{Q}(x_v) & ,\ \mathbf{Q}(x_v) \text{ delivered by } t, \\ \{\underline{\hat{x}}_{\text{PFCS},0}(t)\}_v & ,\ \mathbf{Q}(x_v) \text{ not delivered by } t \end{cases}. \tag{3.66}$$

As it can be predicted, compressed sensing based decoding tries to find an approximate estimation for the un-delivered messages by using the redundancy of messages and improves the overall performance in terms of recovery error norm.

### Quantization and Network Coding (QandNC)

Conventional finite field network coding is also simulated for transmission of messages to the decoder node. In this scenario, similar to packet forwarding, the messages are first quantized at their source nodes, by using a uniform quantizer. The quantizers have a range between $-q_{max}$ and $+q_{max}$ and their step size depends on the transmission block length, $L$. Then, the quantized messages are transmitted to the decoder node by running a classical batch-based finite field network coding [10, 11]. The field size in network coding is determined by the value of $L$ and the network coding coefficients are picked randomly and uniformly from the field elements. At the decoder node, the received finite field packets are collected until $n$ of them are stored and the transmission is then stopped. If the finite field matrix, which maps the messages to the received packets at the decoder node, has full column rank, then the quantized messages can be reconstructed without any error. However, if the field size is not large enough and matrix inversion is not possible, then none of the messages can be decoded. In such case, we set the reconstructed (decoded) messages

[7] This depends on the characteristic of quantizers used at the source node to quantize each message before packet forwarding. Specifically, in our simulations where we used uniform quantizers with step size $\Delta_Q$, $\epsilon_{\text{rec,PF}}(t)$ is equal to the product of $\Delta_Q$ and the number of delivered quantized messages.

to be equal to their mean value (*i.e.* 0 in our simulations):

$$\{\hat{\underline{x}}_{\text{QandNC}}(t)\}_v = 0, \ \forall v. \tag{3.67}$$

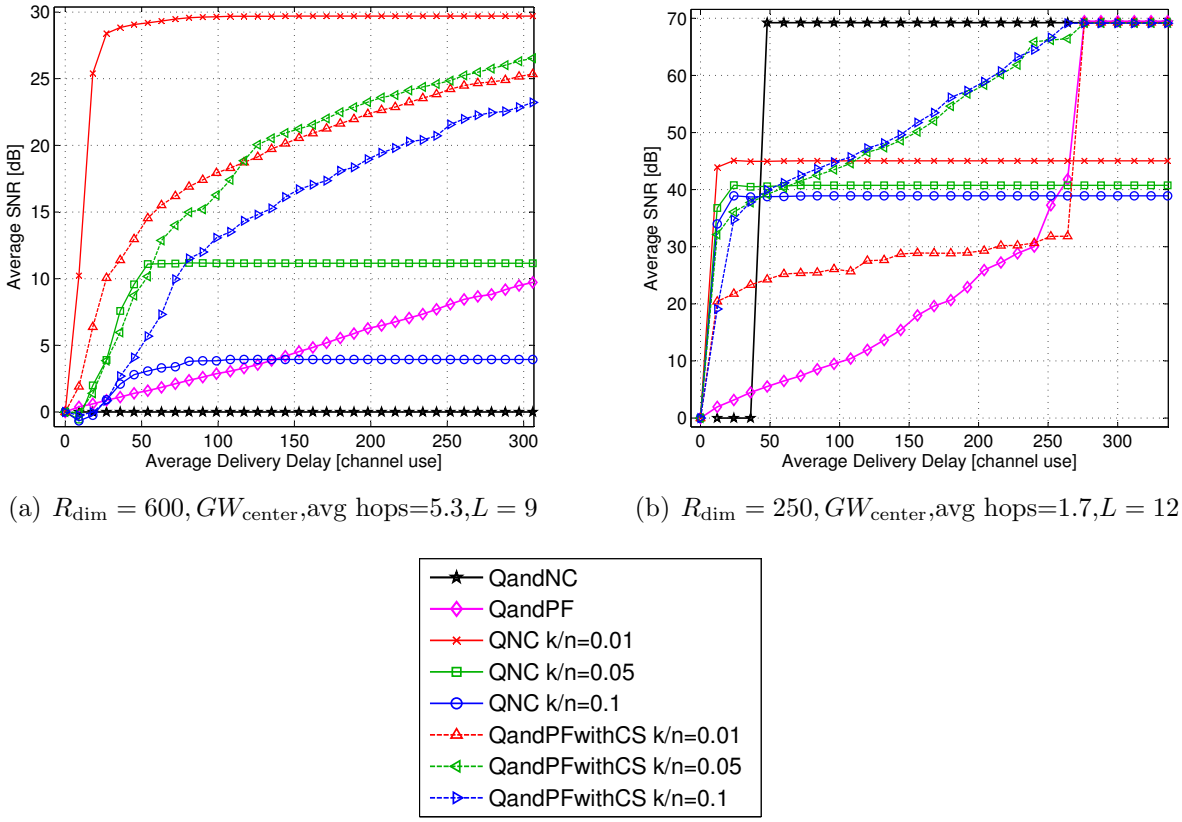
This is referred to as *all or nothing decoding* in the conventional network coding literature. Similar to QNC scenario, the network deployment is assumed to be known at the decode node and the mapping matrix (from messages to received packets) can be built up, by only receiving the seed of pseudo-random generators.

### 3.4.2 Analysis of Simulation Results

For a fixed block length, $L = 9$, the average SNR values versus the average delivery delay is depicted in Fig. 3.4. In Figs. 3.4(a),3.4(b), the horizontal axis represents the product $(t-1)L$, which is the delivery delay, corresponding to $L = 9$, for different values of $t \geq 1$. The vertical axis is the average SNR, calculated according to (3.62), for QNC, QandPF (with and without compressed sensing decoding), and QandNC scenarios.

As it is shown in Figs. 3.4(a),3.4(b), when using the same block length, QNC achieves significant improvement, compared to PF, for low values of delivery delay. These low delays correspond to the initial $t$'s in the transmission, at which a small number of packets are received at the decoder. As promised by theory of compressed sensing, fewer measurements enable message recovery, with an associated measurement noise. After enough packets are received at the decoder, QNC achieves its best performance (where the curve is almost flat). This best performance improves (*i.e.* average SNR increases) when the correlation of messages is higher (sparsity factor $k/n$ is lower).

The best performance for QandPF, QandPFwithCS and QandNC happens after a longer period of time than for QNC. As it can be seen, this is the best achievable quality (SNR value), which is limited only by quantization noises at the source nodes, for both QandPF and QandNC scenarios. As it is also expected, using compressed sensing decoding (as in QandPFwithCS scenario) provides a better estimation of the messages before all the packets are delivered. Furthermore, as opposed to QandPF which shows a progressive improvement in the quality, QandNC has an all or nothing characteristic, as mentioned earlier. It is also interesting to note that low density adjacency matrices in networks with small degree of nodes result in having (finite field) measurement matrices that are not of full rank in the QandNC scenario. Hence, as shown in Fig. 3.4(a), QandNC scheme fails




(a) $R_{\text{dim}} = 600, GW_{\text{center}}$,avg hops=5.3,$L = 9$

(b) $R_{\text{dim}} = 250, GW_{\text{center}}$,avg hops=1.7,$L = 12$

**Figure 3.4** Average SNR versus average delivery delay for QNC, PF, and QandNC scenarios, when $\epsilon_k = 0$.

to work properly.

The quantization noises and their propagation through the network does not allow QNC to achieve the same best performance as in PF and QandNC scenarios (where only source quantization noise is involved). However, as it is shown in the following, QNC outperforms QandPF (with and without compressed sensing decoding) and QandNC scenarios in a wide range of delay values, when an appropriate block length is chosen.

After simulating QNC, QandPF, QandPFwithCS, and QandNC scenarios for different block lengths and calculating the corresponding delay and recovery error norms, we find the best values of block length for each specific average SNR value. The resulting $L$-optimized curves for each of these scenario are shown in Fig. 3.5.

It can be seen in Figs. 3.7(a)-3.5(d), when the network does not have too many links (i.e., when the average hop distances is low), the proposed QNC scenario outperforms

both routing based packet forwarding (with and without compressed sensing decoding) and conventional QandNC scenarios. This is true for a wide range of average SNR values, varying up to around 35 $dB$, which is considered as high quality in many applications. Moreover, as it is expected, the average SNR of QNC scenario increases when the correlation of messages increases (i.e., when the sparsity factor, $k/n$, decreases).

As shown in Figs. 3.5(e),3.5(f), when dealing with networks with very high number of edges, which results in small average hop distances, the proposed QNC scenario can not outperform QandNC scenario, for very high SNR values (explicitly for average SNR values higher than 40 $dB$). This may be as a result of quantization noise propagation through the network during the QNC steps, which strengthen the effective measurement noise above the level that sparse recovery can compensate.

By comparing the figures, in which only the location of gateway node has changed, *i.e.* from $GW_{\text{center}}$ to $GW_{\text{corner}}$ (Fig. 3.7(a) to Fig. 3.7(b) and Fig. 3.5(c) to Fig. 3.5(d)), we can understand that QNC shows a more robust behavior than PF and QandNC schemes. In other words, QNC does not suffer from the complications (especially happening in packet forwarding) caused by asymmetric distribution of network flow. Using compressed sensing decoding for packet forwarding, as in QandPFwithCS scenario, improves the performance of packet forwarding in this situation, although it can not outperform QNC scenario.

We have also studied the effect of the near-sparsity parameter, $\epsilon_k$, on the performance of our QNC scheme. Those results are shown in Fig. 3.6, where the average SNR is depicted versus the average delivery delay, for different settings of network deployment and a fixed sparsity factor of $k/n = 0.01$. Increasing the near-sparsity parameter, $\epsilon_k$, means that the generated messages are getting further away from the sparsity model. As a result, the performance of QNC degrades when $\epsilon_k$ increases, which can be seen in Figs. 3.6(a)-3.6(f). A more sophisticated correlation model, which would incorporates in the decoding procedure other prior information about the messages than only sparsity may improve the performance of the QNC scenario.

In the routing based packet forwarding scenarios (with and without compressed sensing decoding), the intermediate (sensor) nodes have to go through route training and queuing of packets. One of the main advantages of QNC is that the intermediate nodes should only carry out simple linear combination and quantization, which reduces the required computational power of intermediate sensor nodes (they still have to perform sensing and physical layer transmission). On the other hand, at the decoder sides, QNC requires an $\ell_1$-min

decoder which is potentially more complex than the receiver required for packet forwarding. However, since the gateway node is usually capable of handling higher computational operations, this may not be an issue in practical cases.

### 3.4.3 $\ell_1$-Min Decoding using Estimated Transform Matrix ($\phi$)

As it was discussed in Section 3.3, the knowledge of sparsifying matrix $\phi$ which represents the transform domain in which the messages are sparse or near-sparse, should be available at the decoder node. This knowledge may be obtained from the previously decoded generations of messages at the decoder node. In this section, we assume that five generations of messages are transmitted using QandPF scenario. These received quantized generations of messages are then used to calculate the Karhunen Loeve Transform (KLT) of data. Since the KLT presents the domain with the best energy compaction property, it also represents the domain in which the messages are sparse.[8] Therefore, we use the calculated KLT as an estimate for the transform matrix $\phi$ at the decoder side.

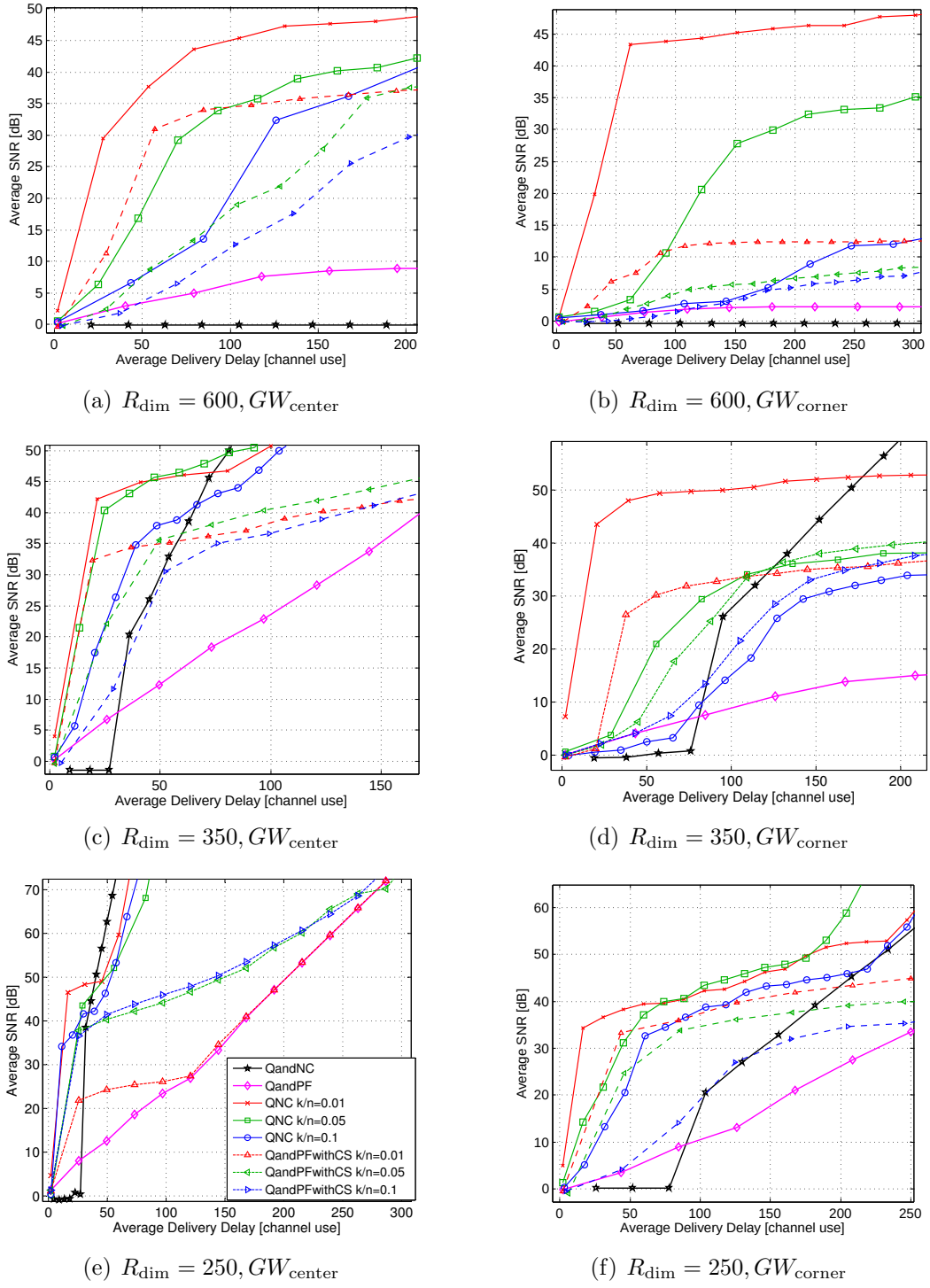
In Fig. 3.7, the decoding SNR is depicted versus the delivery delay for the case where $\phi$ is estimated at the decoder node by calculating the KLT of previously received generations of messages. As it is shown in this figure, when the sparsity factor of messages $k/n$ is small (meaning high correlation of messages), the performance of QNC scenario with estimated $\phi$ does not drop significantly. The decrease in the decoding SNR of QNC scenario with estimated $\phi$ is noticeable when the sparsity factor $k/n$ is increased ($k/n = 0.1$ for the curves in Fig. 3.7).

## 3.5 Summary and Conclusion

In this chapter, we introduced and formulated our quantized network coding for gathering of correlated data in a lossless scenario. Further, we have studied the theoretical conditions needed to guarantee robust sparse recovery by using an $\ell_1$-min decoding algorithm. Specifically, our analysis based on the RIP helped us propose an appropriate design for the local network coding coefficients which results in a total measurement matrix which has similar characteristic to an i.i.d Gaussian matrix, in terms of satisfying RIP. Finally, we presented our simulation results for different configuration of network deployment and

[8] This is done by calculating the sample covariance matrix for the received generations of messages and then eigenvalue decomposition of the resulting sample covariance matrix.

messages. Our simulation results show that the decoding SNR for QNC scenario is higher than other methods of data gathering, especially for low delay regions which is obtained by exploiting the correlation of messages at the decoder side.




(a) $R_{\text{dim}} = 600, GW_{\text{center}}$

(b) $R_{\text{dim}} = 600, GW_{\text{corner}}$

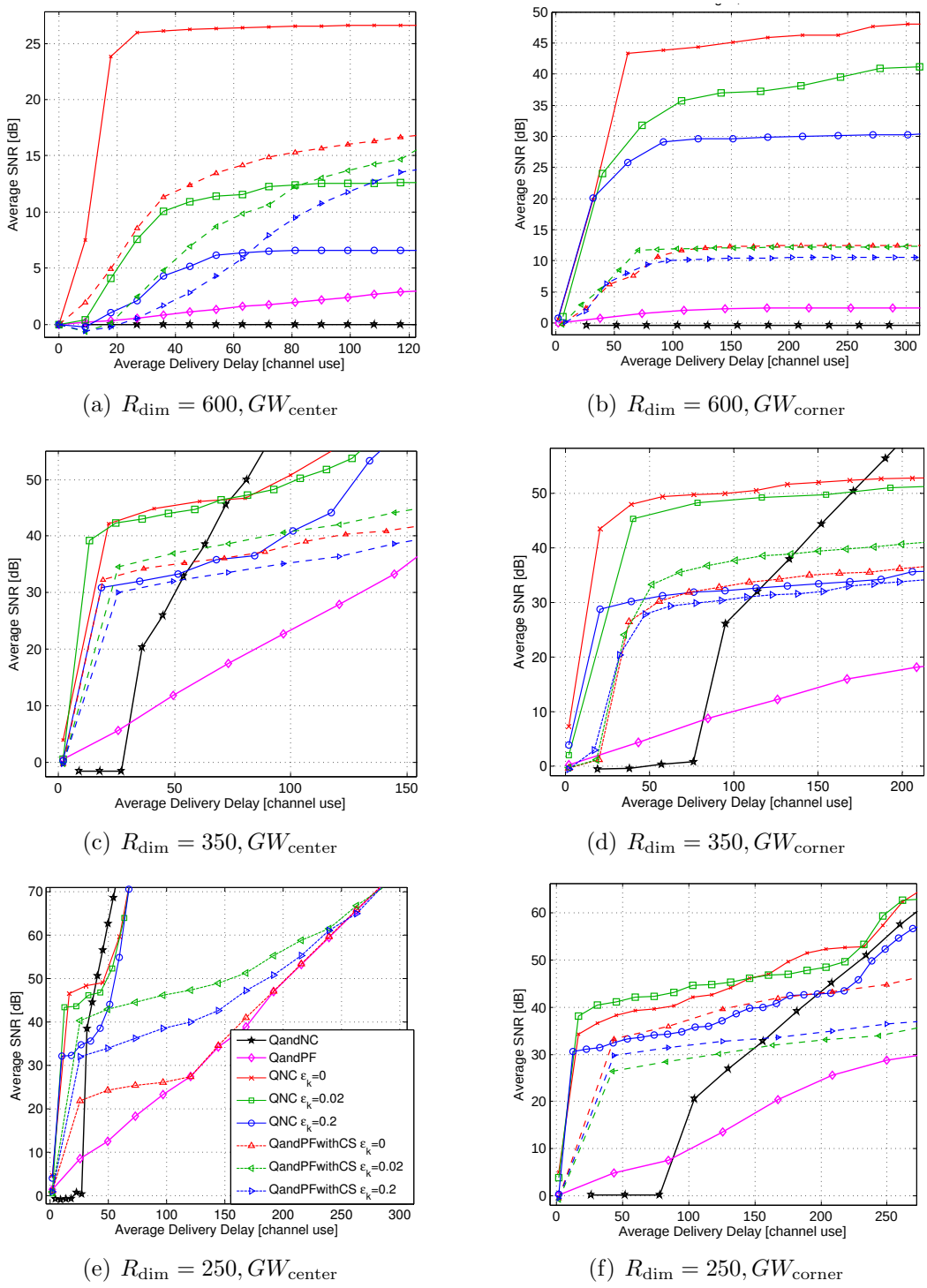
(c) $R_{\text{dim}} = 350, GW_{\text{center}}$

(d) $R_{\text{dim}} = 350, GW_{\text{corner}}$

(e) $R_{\text{dim}} = 250, GW_{\text{center}}$

(f) $R_{\text{dim}} = 250, GW_{\text{corner}}$

**Figure 3.5** Average SNR versus average delivery delay for QNC, PF, and QandNC in different deployment settings, for a fixed $\epsilon_k = 0$.




(a) $R_{\text{dim}} = 600, GW_{\text{center}}$

(b) $R_{\text{dim}} = 600, GW_{\text{corner}}$

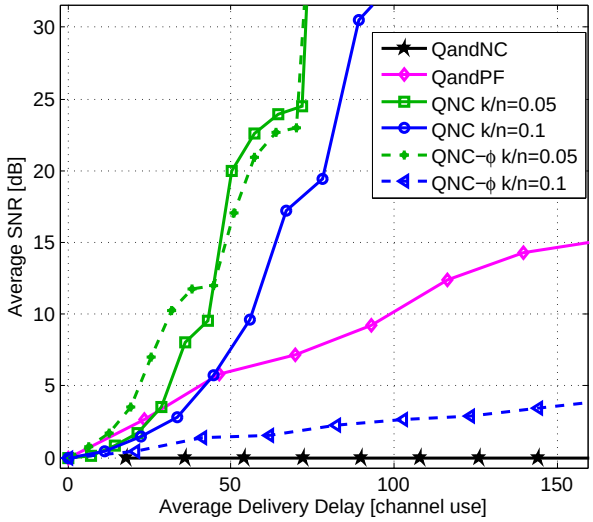
(c) $R_{\text{dim}} = 350, GW_{\text{center}}$

(d) $R_{\text{dim}} = 350, GW_{\text{corner}}$

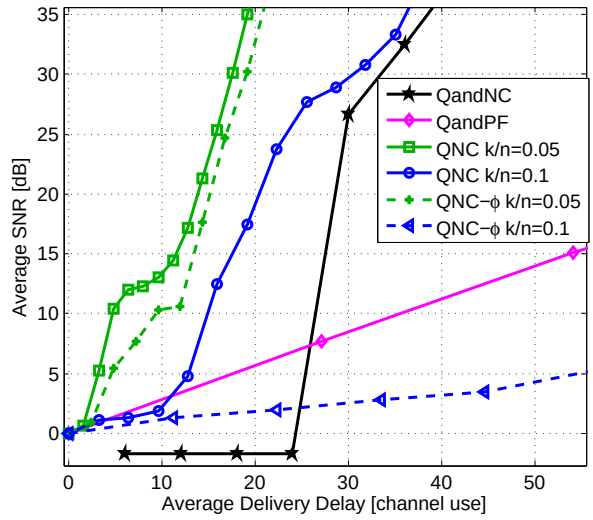
(e) $R_{\text{dim}} = 250, GW_{\text{center}}$

(f) $R_{\text{dim}} = 250, GW_{\text{corner}}$

**Figure 3.6** Average SNR versus average delivery delay for QNC, PF, and QandNC in different deployment settings, for $k/n = 0.01$.




(a) $R_{\text{dim}} = 600, GW_{\text{center}}, \epsilon_k = 0$




(b) $R_{\text{dim}} = 350, GW_{\text{center}}, \epsilon_k = 0$

**Figure 3.7** Average SNR versus average delivery delay when the sparse domain of the messages, determined by $\phi$, is estimated from 5 generations of previously received messages at the decoder node: QNC and QNC $- \phi$ denote QNC scenario using the original $\phi$ and the estimated $\phi$ for the decoding, respectively.

# Chapter 4

# Bayesian Quantized Network Coding

In this chapter, we study the performance of quantized network coding in a Bayesian scenario. To this end, we first describe the Bayesian scenario for quantized network coding. Further, we introduce a version of QNC called one-step QNC which is more easily adapted to the Bayesian framework than usual QNC. Then, we discuss a bound for the decoding error of messages in the described one-step QNC scenario. Further, for the Bayesian scenario with normal quantized network coded messages (described in Section 3.1), we discuss the use of a message passing-based decoder, as an alternative for the classical $\ell_1$-min decoder.

## 4.1 Bayesian Framework and One-Step QNC

Initially, we only assumed that the messages of the nodes are only sparse or near-sparse in some transform domain. However, in some cases, the prior information of the messages is also known. In such Bayesian scenarios, using $\ell_1$-min decoding does not take into account the prior of the messages. Bayesian compressed sensing [55] has recently been studied with a similar motivation. Motivated by the work in [55], we study the Bayesian QNC scenario and derive a performance bound for it in this section.

### 4.1.1 Network Load

Similar to Chapter 3, we assume that each link $e$ in the network is lossless with a capacity $C_e = C_0$ bits per use. We also assume that the edges are uniformly distributed between

pairs of nodes. Explicitly, for a pair of nodes, $v, v' \in \mathcal{V}$, $v \neq v'$, we have:

$$\mathbf{P}\Big(\exists e \in \mathcal{E} : tail(e) = v, head(e) = v'\Big) = \frac{|\mathcal{E}|}{n(n-1)}. \tag{4.1}$$

Moreover, we assume that the messages, take their values randomly and *uniformly*[1] between $-q_{\max}$ and $+q_{\max}$:

$$|X_v| \leq q_{\max}, \ \forall v \in \mathcal{V}. \tag{4.2}$$

In our data gathering scenario, we are interested to decode the messages at the decoder node with a maximum distortion of $D_0$:

$$\mathbf{E}[|X_v - \hat{X}_v|] \leq D_0, \ \forall v \in \mathcal{V}. \tag{4.3}$$

In (4.3), $\hat{X}_v$ is the decoded version of $X_v$, at the decoder node $v_0$. In this chapter, the product of the required number of packets to ensure the above distortion constraint and the packet length is called the *total network load*. This total network load can be used to reflect the required number of transmissions and can be used as a measure of efficiency for different transmission methods.

In the following, we derive the relation between the total network load and the distortion $D_0$ for two transmission scenarios: Quantization and Packet Forwarding, and, a so called One-Step QNC. Quantization and Packet Forwarding is based on routing based packet forwarding of the messages, as explicitly described in Section 3.4.1 (we called it QandPF). One-step QNC scenario is described in details in Section 4.1.2.

### 4.1.2 One-Step QNC

One-step QNC is a special case of QNC, in which we perform linear network coding only at the first time instance and then simply forward the quantized network coded packets to the decoder node. In the following, we explicitly describe our one-step QNC scenario and its formulation.

To initiate one-step QNC, each node transmits the quantized messages, $\mathbf{Q}(X_v)$'s, to its

[1] Assuming a near-sparseness as opposed to sparseness is a more realistic model. For the sake of simplicity, we only consider exactly sparse model.

neighboring nodes. As a result of initial rest condition, at $t = 2$, we have:

$$Y_{e,rx}(2) = \mathbf{Q}(X_{tail(e)}) \quad (4.4)$$
$$= X_{tail(e)} + N_e(2), \quad (4.5)$$

where $N_e(2)$ is the quantization noise. These quantized messages are used in each node $v$ to calculate a random linear combination, $P_v^{\text{one}}$:

$$P_v^{\text{one}} = \sum_{e' \in In(v)} \beta_{v,e'}^{\text{one}} Y_{e',rx}(2) + \alpha_v^{\text{one}} X_v, \ \forall v \in \mathcal{V}, \quad (4.6)$$

where the local network coding coefficients, $\beta_{v,e'}^{\text{one}}$ and $\alpha_v^{\text{one}}$, are uniformly and randomly selected[2] from $\{-c_{\text{norm}}\kappa, +c_{\text{norm}}\kappa\}$, with $\kappa = n/\sqrt{n + |\mathcal{E}|}$. The normalization condition

$$\sum_{e' \in In(v)} |\beta_{v,e'}^{\text{one}}| + |\alpha_v^{\text{one}}| \leq 1, \ \forall v \in \mathcal{V}. \quad (4.7)$$

ensures that overflow does not happen during the calculation of $P_v^{\text{one}}$ at the nodes. We choose an appropriate value for $c_{\text{norm}}$ such that (4.7) is satisfied at all nodes. Essentially, $c_{\text{norm}}$ is a scaling factor which would be neutralized at the decoder node, as discussed later.

Next, each node randomly and independently decides to forward its $P_v^{\text{one}}$ to the decoder node or not, according to a binary probability model. The chosen $P_v^{\text{one}}$ are sent to the decoder node by using a routing-based packet forwarding. When a packet is received at the decoder node, it is re-scaled by $1/c_{\text{norm}}$. Representing the re-scaled version of the $i$'th received packet at the decoder node by $\{\underline{Z}_{\text{tot}}(t)\}_i$, we have:

$$\{\underline{Z}_{\text{tot}}(t)\}_i = \frac{1}{c_{\text{norm}}} \Big( P_v^{\text{one}} + N_e(3) \Big), \ v \longrightarrow i, \quad (4.8)$$

where $v \longrightarrow i$ means that $P_v^{\text{one}}$ is forwarded to decoder and corresponds to the $i$'th received packet at the decoder. The outgoing edge $e$, $e \in Out(v)$, is also the one on which $\mathbf{Q}(P_v^{\text{one}})$ is sent out and therefore $N_e(3)$ is the corresponding quantization noise. The total number of received packets at the decoder node, by time instance $t$, is represented by $m$.[3] The

[2]This choice of $\beta_{v,e'}^{\text{one}}$ and $\alpha_v^{\text{one}}$ will allow us to derive an analytic conclusion on the performance of one-step QNC scenario in Section 4.1.3.

[3]Similar to usual QNC, $m$ is a function of $t$ and increases as $t$ increases.

parameters of probability model, used to decide about sending $P_v^{\text{one}}$'s, will determine the value of $m$.

The one-step QNC scenario can be formulated similar to our original formulation of measurement equation in (3.14), where the total measurement matrix and total measurement noise vector are such that:

$$\{\Psi_{\text{tot}}(t)\}_{i,v} = \begin{cases} \frac{1}{c_{\text{norm}}}\beta^{\text{one}}_{v',e'} & , \ v' \longrightarrow i, v \xrightarrow{e'} v', \\ \frac{1}{c_{\text{norm}}}\alpha^{\text{one}}_{v'} & , \ v' \longrightarrow i, v' = v, \\ 0 & , \ \text{otherwise} \end{cases} \tag{4.9}$$

$$\{\underline{N}_{\text{eff,tot}}(t)\}_i = \frac{1}{c_{\text{norm}}} N_e(3) + \frac{1}{c_{\text{norm}}} \sum_{e' \in In(v)} \beta^{\text{one}}_{v,e'} \ N_{e'}(2), \ v \longrightarrow i. \tag{4.10}$$

In (4.9), $v \xrightarrow{e} v'$ denotes that there is an edge $e$ from $v$ to $v'$. Further, as it was mentioned earlier $v \longrightarrow i$ means that $P_v^{\text{one}}$ is forwarded to the decoder node and corresponds to the $i$'th received packet at the decoder.

### 4.1.3 Asymptotic Analysis of One-Step QNC and Packet Forwarding

In this section, we derive an upper bound on the decoding error of QandPF and one-step QNC scenarios by analyzing their total network loads for a given allowable distortion $D_0$. Using the derived bounds, we discuss the asymptotic performance of QandPF and one-step QNC scenarios when the number of nodes $n$ is large.

#### Distortion Analysis for Quantization and Packet Forwarding

The limited capacity of the links requires us to quantize the messages before transmission over the links. This is what motivates us to use of phrase "Quantization and Packet Forwarding" (QandPF) for this method. Specifically, we quantize the messages and send the quantized version, $\mathbf{Q}(X_v)$, to the decoder node. Since the messages are uniformly distributed, a uniform quantizer minimizes the associated quantization noise power, for which:

$$\mathbf{E}[|X_v - \mathbf{Q}(X_v)|] = \frac{\Delta_Q}{4}. \tag{4.11}$$

Now, for a value of $D_0$, if we pick the packet length carrying the quantized values $L$ such that:

$$L = \lceil \frac{1}{C_0} \log_2(\frac{q_{\max}}{2D_0}) \rceil, \tag{4.12}$$

then the resulting quantization step $\Delta_Q$ is such that:

$$\Delta_Q = \frac{2q_{\max}}{LC_0} \leq D_0. \tag{4.13}$$

Hence, this will allow us to present the following corollary.

**Corollary 4.1.1** *For the described quantization and packet forwarding scenario with real-valued uniform messages, the distortion level of $D_0$,*

$$\boldsymbol{E}[|X_v - \hat{X}_v|] \leq D_0, \ \forall v \in \mathcal{V}, \tag{4.14}$$

*is achieved if and only if the adopted packet length, $L$, to transmit $(n-1)$ quantized messages is such that:*

$$L = \lceil \frac{1}{C_0} \log_2(\frac{q_{\max}}{2D_0}) \rceil, \tag{4.15}$$

*resulting a total network load of*

$$L \cdot (n-1) = (n-1) \lceil \frac{1}{C_0} \log_2(\frac{q_{\max}}{2D_0}) \rceil. \ \blacksquare \tag{4.16}$$

It can be understood from this corollary that the total network load is in the order of the number of nodes, $n$. In Section 4.1.3, we will see that by using our previously proposed one-step QNC, one requires a smaller load to ensure the same distortion level.

### Distortion Analysis for One-Step Quantized Network Coding

The precise distortion analysis for QNC scenario can not be done easily because of its reliance on sparse recovery algorithms, for which a comprehensive statistical distortion analysis is still not available. However, we are still able to obtain some performance bounds on the delay-quality performance, thanks to the work on Bayesian compressed sensing [55]. In the following, we present a series of theoretical results, which characterizes the delay-distortion performance of one-step QNC scenario for non-adaptive encoding of sparse messages.

**Theorem 4.1.2** *For the one-step QNC scenario, described in Section 4.1.2, where* $\underline{X}$ *is* $k$*-sparse in the transform domain* $\phi_{n\times n}$*, with*

$$q'_{\max} = \max_{\underline{X}} \left|\left|\phi^T \cdot \underline{X}\right|\right|_{\ell_\infty}, \tag{4.17}$$

*then for any* $\epsilon, \gamma > 0$*, using a number of packets,* $m$*, where*

$$m > 48(1+\gamma)\frac{(\kappa^2-1)\ kq'^2_{\max} + \Delta_Q^2/c^2_{\text{norm}}}{\epsilon^2}\log(n), \tag{4.18}$$

*one can decode* $\underline{Z}_{\text{tot}}(t)$ *into* $\underline{\hat{X}}$*, such that:*

$$\boldsymbol{P}(|X_v - \hat{X}_v| < \epsilon) \geq 1 - n^{-\gamma},\ \forall v \in \mathcal{V}. \tag{4.19}$$

**Proof** Assuming (without loss of generality) that $P_{v'}$ is forwarded to decoder and corresponds to the $i$'th received packet, for the independent $\{\Psi_{\text{tot}}(t)\}_{iv}$'s, we have:

$$\begin{aligned}
\mathbf{P}(\{\Psi_{\text{tot}}(t)\}_{iv} = 0) &= \mathbf{P}(v \neq v',\ \nexists e' : v \xrightarrow{e'} v') \\
&= \mathbf{P}(v \neq v')\ \mathbf{P}(\nexists e' : v \xrightarrow{e'} v' | v \neq v') \\
&= \Big(1 - \frac{1}{n}\Big)\Big(1 - \frac{(|\mathcal{E}|/n)!\binom{n-1}{|\mathcal{E}|/n}}{(|\mathcal{E}|/n)!\binom{n-2}{|\mathcal{E}|/n-1}}\Big) \\
&= \Big(1 - \frac{1}{n}\Big)\Big(1 - \frac{|\mathcal{E}|}{n(n-1)}\Big).
\end{aligned} \tag{4.20}$$

Since $+\kappa$ and $-\kappa$ are picked with the same probability for network coding coefficients, we have:

$$\begin{aligned}
\mathbf{P}(\{\Psi_{\text{tot}}(t)\}_{iv} = +\kappa) &= \mathbf{P}(\{\Psi_{\text{tot}}(t)\}_{iv} = -\kappa) \\
&= \frac{1}{2} - \frac{1}{2}\mathbf{P}(\{\Psi_{\text{tot}}(t)\}_{iv} = 0) \\
&= \frac{1}{2} - \frac{1}{2}\Big(1 - \frac{1}{n}\Big)\Big(1 - \frac{|\mathcal{E}|}{n(n-1)}\Big) \\
&= \frac{n + |\mathcal{E}|}{2n^2} = \frac{1}{\kappa^2}.
\end{aligned} \tag{4.21}$$

For $\forall i, v,\ 1 \leq i \leq m, 1 \leq v \leq n$, this implies:

$$\begin{aligned} \mathbf{E}[\{\Psi_{\text{tot}}(t)\}_{iv}] &= 0, \\ \mathbf{E}[\{\Psi_{\text{tot}}(t)\}_{iv}^2] &= 1, \\ \mathbf{E}[\{\Psi_{\text{tot}}(t)\}_{iv}^4] &= \kappa^2. \end{aligned} \tag{4.22}$$

Define positive integers $m_1$ and $m_2$, which we will determine, and set $m = m_1 m_2$. Partition the $m \times n$ matrix $\Psi_{\text{tot}}(t)$ into $m_2$ matrices $\{\Psi_1, \ldots, \Psi_{m_2}\}$, each of size $m_1 \times n$. Now, consider a realization of $\underline{X}$, called $\underline{x}$, for which we define $\{\underline{z}_1 = \frac{1}{\sqrt{m_1}}\Psi_1 \underline{x} + \underline{n}_1, \ldots, \underline{z}_{m_2} = \frac{1}{\sqrt{m_1}}\Psi_{m_2}\underline{x} + \underline{n}_{m_2}\}$. Moreover, $\{\underline{n}_l\}_i$'s are the corresponding elements of the total effective noise vector, for which (4.10) implies: $|\{n_l\}_i| \leq \Delta_Q$.

Furthermore, we define $\underline{u}_j$'s, where $1 \leq j \leq n$, to be the canonical vectors in $\mathbb{R}^n$. For them, we define $\{\underline{z}'_{1,j} = \frac{1}{\sqrt{m_1}}\Psi_1 \underline{u}_j, \ldots, \underline{z}'_{m_2,j} = \frac{1}{\sqrt{m_1}}\Psi_{m_2}\underline{u}_j\}$.

Now, we define random variables $r_{1,j}, \ldots, r_{m_2,j}$:

$$r_{l,j} = \underline{z}_l^T \cdot \underline{z}'_{l,j} = \sum_{i=1}^{m_1} \{\underline{z}_l\}_i \ \{\underline{z}'_{l,j}\}_i = \sum_{i=1}^{m_1} \frac{1}{m_1} w_{l,j,i},$$

where:

$$w_{l,j,i} = \Big(\sum_{v=1}^{n} \{\Psi_l\}_{i,v} x_v + \{\underline{n}_l\}_i\Big) \cdot \Big(\sum_{v=1}^{n} \{\Psi_l\}_{i,v} \{\underline{u}_j\}_v\Big).$$

Moreover, $r_{1,j}, \ldots, r_{m_2,j}$ are independent because of the independence of $\{\Psi_l\}_{i,v}$'s. Further, using 4.22, we have:

$$\begin{aligned} \mathbf{E}[w_{l,j,i}] &= \mathbf{E}[\sum_{v=1}^{n} \{\Psi_l\}_{i,v}^2 \ x_v \ \{\underline{u}_j\}_v] \\ &\quad + 2\mathbf{E}[\sum_{v=1}^{n} \sum_{v'=v+1}^{n} \{\Psi_l\}_{i,v} \{\Psi_l\}_{i,v'} x_v \{\underline{u}_j\}_{v'}] \\ &\quad + \{\underline{n}_l\}_i \ \mathbf{E}[(\sum_{v=1}^{n} \{\Psi_l\}_{i,v} \{\underline{u}_j\}_v)] \\ &= \sum_{j=1}^{n} x_v \ \{\underline{u}_j\}_v = \underline{x}^T \cdot \underline{u}_j, \end{aligned}$$

$$\begin{aligned}\mathbf{E}[w_{l,j,i}^2] &= \mathbf{E}[(\sum_{v=1}^{n}\{\Psi_l\}_{i,v}x_v)^2\cdot(\sum_{v=1}^{n}\{\Psi_l\}_{i,v}\{\underline{u}_j\}_v)^2] \\ &\quad +\{\underline{n}_l\}_i^2\ \mathbf{E}[(\sum_{v=1}^{n}\{\Psi_l\}_{i,v}\{\underline{u}_j\}_v)^2] \\ &\quad +2\{\underline{n}_l\}_i\ \mathbf{E}[(\sum_{v=1}^{n}\{\Psi_l\}_{i,v}x_v)(\sum_{v=1}^{n}\{\Psi_l\}_{i,v}\{\underline{u}_j\}_v)^2] \qquad (4.23)\end{aligned}$$

Using Lemma 1 in [71] and also the fact that $\mathbf{E}[\{\Psi_l\}_{i,v}^3]=0$, we have[4]:

$$\begin{aligned}\mathbf{E}[w_{l,j,i}^2] &= 2(\underline{x}^T\cdot\underline{u}_j)^2+\|\underline{x}\|_{\ell_2}^2\ \left\|\underline{u}_j\right\|_{\ell_2}^2 \\ &\quad +(\kappa^2-3)(\sum_{v=1}^{n}x_v^2\{\underline{u}_l\}_v^2)+\{\underline{n}_l\}_i^2\left\|\underline{u}_j\right\|_{\ell_2}^2,\end{aligned}$$

Recalling that $\underline{u}_j$'s are canonical vectors with unit $\ell_2$-norm, we have:

$$\mathbf{Var}[w_{l,j,i}] = \|\underline{x}\|_{\ell_2}^2+(\kappa^2-2)x_j^2+\{\underline{n}_l\}_i^2.$$

Now, for $r_{l,j}$, we have:

$$\mathbf{E}[r_{l,j}] = \sum_{i=1}^{m_1}\frac{1}{m_1}\mathbf{E}[w_{l,i}]=\underline{x}^T\cdot\underline{u}_j,$$

$$\begin{aligned}\mathbf{Var}[r_{l,j}] &= \frac{1}{m_1^2}\sum_{i=1}^{m_1}\mathbf{Var}[w_{l,j,i}] \\ &= \frac{1}{m_1}\Big(\|\underline{x}\|_{\ell_2}^2+(\kappa^2-2)x_j^2+\{\underline{n}_l\}_i^2\Big)\ .\end{aligned}$$

Thus, the Chebyshev inequality,

$$\mathbf{P}(|r_{l,j}-\underline{x}^T\underline{u}|\geq\epsilon) \leq \frac{\mathbf{Var}(r_{l,j})}{\epsilon^2},$$

[4]Lemma 1 in [71] provides a simplification of the first term in (4.23) for matrices which satisfy the moments conditions in (4.22).

and the fact that $|\{\underline{n}_l\}_i| \leq \Delta_Q/c_{\text{norm}}$, imply:

$$\begin{aligned}\mathbf{P}(|r_{l,j} - x_j| \geq \epsilon) &\leq \frac{1}{\epsilon^2 m_1}\Big(\|\underline{x}\|_{\ell_2}^2 + (\kappa^2 - 2)x_j^2 + \frac{\Delta_Q^2}{c_{\text{norm}}^2}\Big) \\ &\leq \frac{1}{\epsilon^2 m_1}\Big((\kappa^2 - 1)\max_{\underline{x}} \|\underline{x}\|_{\ell_2}^2 + \frac{\Delta_Q^2}{c_{\text{norm}}^2}\Big).\end{aligned}$$

By picking

$$m_1 > 4\frac{(\kappa^2 - 1)\max_{\underline{x}} \|\underline{x}\|_{\ell_2}^2 + \Delta_Q^2/c_{\text{norm}}^2}{\epsilon^2}, \tag{4.24}$$

we have:

$$\mathbf{P}(|r_{l,j} - x_j| > \epsilon) < \frac{1}{4}. \tag{4.25}$$

We define $\hat{x}_j$, the decoded value for $x_j$, to be the median of the independent random variables $r_{1,j}, \ldots, r_{m_2,j}$. Further, we define the indicator random variable $\xi_{l,j}$ to be equal to one if $|r_{l,j} - x_j| > \epsilon$ and zero otherwise. Now, if $\sum_{l=1}^{m_2} \xi_{l,j}$ is more than $\frac{m_2}{2}$, the median (decoded value) will not be inside the interval: $|\hat{x}_j - x_j| > \epsilon$. Moreover, we have: $\mathbf{E}[\sum_{l=1}^{m_2} \xi_{l,j}] < \frac{m_2}{4}$. By using the Chernoff inequality

$$\mathbf{P}(\sum_{l=1}^{m_2} \xi_{l,j} > (1+\delta)\mathbf{E}[\sum_{l=1}^{m_2} \xi_{l,j}]) \leq e^{-\delta^2 \mathbf{E}[\sum_{l=1}^{m_2} \xi_{l,j}]/3} \tag{4.26}$$

where $0 < \delta < 1$ [72], we obtain:

$$\mathbf{P}(\sum_{l=1}^{m_2} \xi_{l,j} > (1+\delta)\frac{m_2}{4}) < e^{-\delta^2 m_2/12}. \tag{4.27}$$

Taking the limit of (4.27) when $\delta \to 1^-$,

$$\mathbf{P}(\sum_{l=1}^{m_2} \xi_{l,j} > \frac{m_2}{2}) < e^{-m_2/12}. \tag{4.28}$$

Finally, by using the union bound, we have:

$$\mathbf{P}\Big(\exists j : |x_j - \hat{x}_j| > \epsilon\Big) \leq n \cdot e^{-m_2/12}. \tag{4.29}$$

Therefore, by choosing

$$m_2 > 12(1+\gamma)\log(n), \tag{4.30}$$

and combining with (4.24) to find $m = m_1 \cdot m_2$, we have proved the theorem. ■

**Theorem 4.1.3** *For the described one-step QNC scenario, if the total network load, $m \cdot L$, satisfies*

$$m \cdot L > 96 \frac{(\kappa^2 - 1)k q_{\max}'^2 + q_{\max}^2 / c_{\text{norm}}^2 2^{2-2C_0}}{(D_0 - 2q_{\max}/n)^2} \cdot \log(n), \tag{4.31}$$

*then we have:*

$$\boldsymbol{E}[|X_v - \hat{X}_v|] \leq D_0, \ \forall v \in \mathcal{V}. \tag{4.32}$$

**Proof** From theorem 4.1.2, we know that if for a choice of $\epsilon, \gamma > 0$ the condition of (4.18) is satisfied, then the expected value of $|X_v - \hat{X}_v|$ is smaller than $\epsilon$, with probability $1 - n^{-\gamma}$. We also know that $X_v$'s are bounded between $-q_{\max}$ and $+q_{\max}$. Therefore, if we limit (clip) the outcome of the decoder, $\hat{X}_v$, to be between $-q_{\max}$ and $+q_{\max}$, we can ensure that $|X_v - \hat{X}_v| \leq 2q_{\max}$, for the portion of the times ($n^{-\gamma}$) that theorem 4.1.2 fails to provide any guarantee. Hence:

$$\mathbf{E}[|X_v - \hat{X}_v|] \leq \epsilon(1 - n^{-\gamma}) + 2q_{\max} n^{-\gamma}. \tag{4.33}$$

And, by constraining:

$$\epsilon(1 - n^{-\gamma}) + 2q_{\max} n^{-\gamma} \leq D_0, \tag{4.34}$$

we can also ensure that $\mathbf{E}[|X_v - \hat{X}_v|] \leq D_0$ holds.

Now, by choosing $\gamma = 1$, we obtain:

$$\epsilon \leq \frac{nD_0 - 2q_{\max}}{n-1}. \tag{4.35}$$

And since

$$D_0 - \frac{2q_{\max}}{n} = \frac{nD_0 - 2q_{\max}}{n} < \frac{nD_0 - 2q_{\max}}{n-1}, \tag{4.36}$$

by choosing $\epsilon = D_0 - 2q_{\max}/n$ and $L = 1$, we conclude the proof. ■

Theorem 4.1.3 provides an easy statement about the required condition to ensure the distortion level in the one-step QNC scenario. Explicitly, when the number of network

nodes, $n$, is large, the total network load in one-step QNC scenario, $m \cdot L$, (Eq. 4.31) is in the order of $\log(n)$. As discussed in Section 4.1.3, the total network load in QPF scenario, $(n-1) \cdot L$, has order of $n$, for the same distortion level. This clarifies a significant reduction in the total network load in one-step QNC scenario, for large networks.

### 4.1.4 Summary

In this section, we discussed the possibility of non-adaptive distributed compression of correlated messages. This was done by discussing the asymptotic performance of one-step QNC scenario, which is a simplification of QNC scenario. Specifically, we derived a sufficient (not necessary) condition on the network load to satisfy a desired distortion level. We mathematically showed in (4.16) that for a given distortion level, conventional quantization and packet forwarding requires a total network load of order $n$. As shown in (4.31), this order is reduced to $\log(n)$ when one-step QNC is adopted for transmission, implying an embedded distributed compression. The resulting decrease in the total network load shows a potential decrease in the overall transmission power in the network.

## 4.2 Decoding using Generalized Approximate Message Passing

In Section 4.1, we introduced the Bayesian framework for quantized network coding of messages with known prior. We have also introduced one-step QNC in which linear mixing is done for only one step of QNC, allowing us to derive a mathematical performance bound for the Bayesian framework. Although studying one-step QNC has allowed us to provide a theoretical comparison of QandPF and QNC scenario, the fact that one-step QNC relies on routing makes it undesirable in practice compared to QNC.

In this section, we study the Bayesian framework by using numerical evaluation of normal QNC and QandPF scenarios. Specifically, we focus on the study of Minimum Mean Squared Error (MMSE) decoding for normal quantized network coded messages (as described in Section 3.1) where the prior of messages is known beyond its sparsity at the decoder node. In contrast with our studies in Section 4.1, in this section we assume that the prior of messages is modeled by a two state Gaussian mixture, explained below.[5]

[5]The two-state Gaussian mixture model provides a better fit to the real signals than the uniform distribution used for our analysis in Section 4.1.

We denote the elements of the near-sparse domain representing the messages $\underline{S}$ by $S_j$ where $1 \leq j \leq n$. Further, we consider independent binary random variables $T_j$'s to represent the state of $S_j$'s, with:

$$\mathbf{P}(T_j = 1) = \frac{k}{n}. \tag{4.37}$$

Each state, $T_j$, determines if $S_j$ is close-to-zero or not:

$$\begin{aligned} T_j = 1 \quad &\rightarrow \quad S_j \sim \mathcal{N}(0, \sigma_s^2), \\ T_j = 0 \quad &\rightarrow \quad S_j \sim \mathcal{N}(0, \sigma_0^2), \end{aligned} \tag{4.38}$$

where $\sigma_s^2 \gg \sigma_0^2$. Therefore, the prior of independently modeled $S_j$'s is as follows:

$$\mathbf{p}_{S_j}(s_j) = \frac{k}{n} g(s_j; \sigma_s^2) + (1 - \frac{k}{n}) g(s_j; \sigma_0^2) \tag{4.39}$$

where $g(\centerdot; \centerdot)$ is the probability density function of a zero mean Gaussian random variable:

$$g(s_j; \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} s_j^2}, \tag{4.40}$$

and,

$$\sigma^2_{\text{source}} = \mathbf{E}[X_v^2] = \mathbf{E}[S_j^2] = \frac{k}{n}\sigma_s^2 + (1 - \frac{k}{n})\sigma_0^2, \ \forall v, j. \tag{4.41}$$

As we will mention MMSE decoding is computationally unfeasible and we have to use alternatives for near MMSE decoding. Specifically, we propose to use Generalized Approximate Message Passing (GAMP) [73] in order to implement a near MMSE decoder for our Bayesian QNC scenario.

### 4.2.1 Minimum Mean Square Error Decoding

In a general case, the marginal quantization noises $n_e(t)$'s, are dependent of the messages. Therefore, finding the MMSE solution from the received quantized network coded packets is very complicated because of the step-wise non-linearity, contributed by the quantization noises. However, we make practical assumptions which makes us able to derive an appropriate mathematical model for the effective measurement noises and ease the analysis for MMSE decoding.

For ease of notation, we define $\{\Xi(t)\}_{e,v}$ to be the transform coefficient from $X_v$ to

$Y_{e,rx}(t)$, describing the transfer of information of messages through the network. More explicitly, by using the results in Section 3.1, the matrix $\Xi(t) = [\{\Xi(t)\}_{e,v}]_{|\mathcal{E}|\times n}$ can be calculated according to:

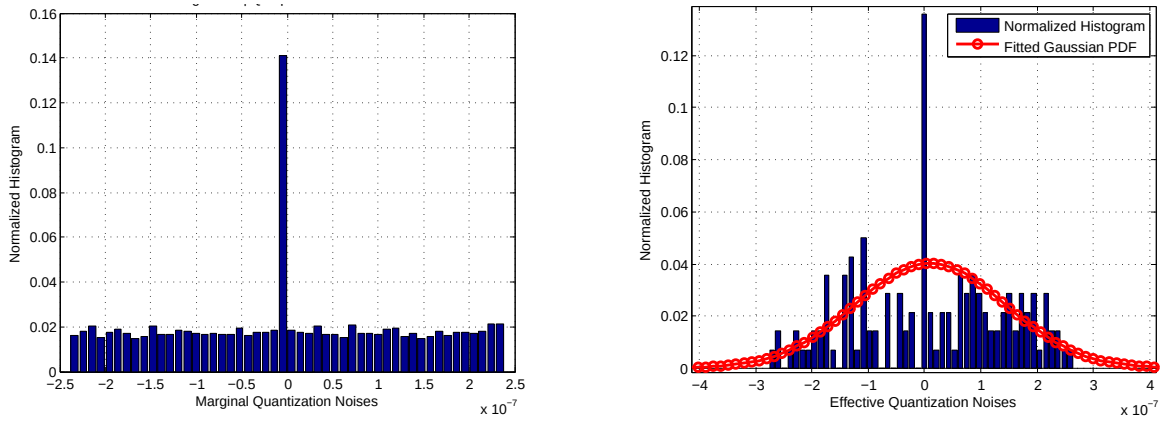
$$\Xi(t) = \sum_{t'=2}^{t} F_{\text{prod}}(t'+1;t)A(t'), \tag{4.42}$$

where $F_{\text{prod}}(\centerdot,\centerdot)$ is defined in (3.6). Therefore, the variance of $Y_{e,rx}(t)$'s can be formulated as:

$$\begin{aligned}\mathbf{E}[Y_{e,rx}^2(t)] &= \sum_{t'=2}^{t}\sum_{e'=1}^{|\mathcal{E}|}\{F_{\text{prod}}(t'+1;t)\}_{e,e'}^2\mathbf{E}[N_{e'}(t')^2] \\ &\quad + \sum_{v=1}^{n}\{\Xi(t)\}_{e,v}^2\mathbf{E}[X_v^2].\end{aligned} \tag{4.43}$$

When the quantization noises, $N_e(t)$'s, have small variance compared to that of the signal, the variance of $Y_{e,rx}(t)$'s can be approximated with the variance of noiseless propagated information, that is:

$$\mathbf{E}[Y_{e,rx}^2(t)] \simeq \sum_{v=1}^{n}\{\Xi(t)\}_{e,v}^2\mathbf{E}[X_v^2]. \tag{4.44}$$




(a) Histogram of Marginal Quantization Noises

(b) Histogram of Effective Quantization Noises

**Figure 4.1** Normalized Histogram of Marginal and Effective Quantization Noises: The quantizers have a uniform characteristic with the step size of $4.7e-7$. The network deployments and the network coding coefficients are generated as described in Section 3.4.

We study the case where the link quantizers have a *uniform* characteristic, similar to those used in Section 3.4. When the quantization step sizes are small, the marginal quantization noises, $N_e(t)$'s, have a probability density function close to uniform distribution. In Fig. 4.1(a), the histogram of these marginal quantization noises is shown. Moreover, $N_e(t)$'s are approximately independent and their variance, $\mathbf{E}[N_e^2(t)]$, is proportional to the variance of corresponding quantizer input. Hence:

$$\mathbf{E}[N_e^2(t)] \simeq c_{e,\mathrm{Q}}(t) \sum_{v=1}^{n} \Xi_{e,v}^2(t) \mathbf{E}[X_v^2] \tag{4.45}$$

$$= c_{e,\mathrm{Q}}(t) \sigma_{\mathrm{source}}^2 \sum_{v=1}^{n} \Xi_{e,v}^2(t) \tag{4.46}$$

where $c_{e,\mathrm{Q}}(t)$ is a positive scalar, depending on the quantizer design. Defining

$$\underline{N}_{\mathrm{tot}}(t) = \begin{bmatrix} \underline{N}(2) \\ \vdots \\ \underline{N}(t) \end{bmatrix}_{(t-1)|\mathcal{E}|\times 1}, \tag{4.47}$$

the effective total measurement noise, $\underline{N}_{\mathrm{eff,tot}}(t)$, can be formulated according to:

$$\underline{N}_{\mathrm{eff,tot}}(t) = \Psi_{\mathrm{N,tot}}(t) \cdot \underline{N}_{\mathrm{tot}}(t). \tag{4.48}$$

This implies:

$$\mathbf{E}[\underline{N}_{\mathrm{eff,tot}}(t) \underline{N}_{\mathrm{eff,tot}}^T(t)] = \Psi_{\mathrm{N,tot}}(t) \Lambda_{\mathrm{Q}}(t) \Psi_{\mathrm{N,tot}}^T(t) \tag{4.49}$$

where $\Lambda_{\mathrm{Q}}(t)$ is the covariance matrix of quantization noises:

$$\Lambda_{\mathrm{Q}}(t) = \mathbf{E}[\underline{N}_{\mathrm{tot}}(t) \cdot \underline{N}_{\mathrm{tot}}(t)^T]. \tag{4.50}$$

Considering that $\{\underline{N}_{\mathrm{eff,tot}}(t)\}_i$ is a sum of identical (close to uniform) and (approximately) independent random variables, based on the central limit theorem [74], one may say that the resulting effective quantization noises, $\{\underline{N}_{\mathrm{eff,tot}}(t)\}_i$'s, have a Gaussian distribution. This approximation is shown in Fig. 4.1(b), where the histogram of the effective quantization noises is depicted for a QNC scenario. The mean of this Gaussian distribution is the sum of the means of (approximately zero-mean) uniform quantization noises, and is

thus (approximately) zero. The mean of this Gaussian distribution is the weighted sum of the marginal uniform (with zero mean) distribution, resulting in a zero value. The variance of this Gaussian distribution can also be calculated from (4.50).

By definition, the MMSE estimate of $\underline{X}$ is its conditional expectation given the measurements, calculated according to:

$$\begin{aligned} \hat{\underline{x}}_{\text{MMSE}}(t) &= \mathbf{E}\Big[\underline{X}\Big|\underline{Z}_{\text{tot}}(t)=\underline{z}_{\text{tot}}(t)\Big] \\ &= \phi\ \mathbf{E}\Big[\underline{S}\Big|\underline{Z}_{\text{tot}}(t)=\underline{z}_{\text{tot}}(t)\Big] \\ &= \phi\int_{-\underline{\infty}}^{+\underline{\infty}} \underline{s}\ \mathbf{p}_{\underline{S}}\Big(\underline{s}\Big|\underline{Z}_{\text{tot}}(t)=\underline{z}_{\text{tot}}(t)\Big)\cdot d\underline{s}, \end{aligned} \tag{4.51}$$

where:

$$\mathbf{p}_{\underline{S}}\Big(\underline{s}\Big|\underline{Z}_{\text{tot}}(t)=\underline{z}_{\text{tot}}(t)\Big)=\frac{\mathbf{p}_{\underline{S}}(\underline{s})\ \mathbf{p}_{\underline{Z}_{\text{tot}}(t)}\Big(\underline{z}_{\text{tot}}(t)\Big|\underline{S}=\underline{s}\Big)}{\mathbf{p}_{\underline{Z}_{\text{tot}}(t)}\Big(\underline{z}_{\text{tot}}(t)\Big)},$$

and,

$$\mathbf{p}_{\underline{Z}_{\text{tot}}(t)}\Big(\underline{z}_{\text{tot}}(t)\Big)=\int_{-\underline{\infty}}^{+\underline{\infty}}\mathbf{p}_{\underline{Z}_{\text{tot}}(t)}\Big(\underline{z}_{\text{tot}}(t)\Big|\underline{S}=\underline{s}\Big)\mathbf{p}_{\underline{S}}(\underline{s})\cdot d\underline{s}.$$

In the Bayesian framework, we assume that the probability density function of messages in their sparse domain $\mathbf{p}_{\underline{S}}(\centerdot)$ is known. Now, having a prior of $\underline{S}$, the distribution of quantization noises (as described in this section), and the measurement equation of (3.14), one could calculate the posterior probability of $\underline{X}$ and its MMSE estimation, $\hat{\underline{X}}_{\text{MMSE}}(t)$. However, this requires a high computational complexity for the decoder, which makes it practically infeasible. To tackle this practicality issue, we propose to use a message passing-based iterative algorithm for decoding of messages. Such decoding algorithms have been used in a number of low density linear mixing problems, including lattice codes [75]. In the following, we describe the algorithm, used to decode messages in the considered Bayesian QNC scenario.

### 4.2.2 Generalized Approximate Message Passing

Message passing based decoding algorithms are widely used in scenarios where the linear mixing involves a low density structure, *e.g.* low density parity check codes. In such cases with low density structures, conventional message passing (also referred to as belief

propagation) is shown to asymptotically converge to the desirable MMSE solution. The low density structure also makes us able to have a low computational complexity. However, in cases with dense linear mixing, conventional message passing fails to work properly. Very recent works have developed new versions of the message passing algorithm, which are able to handle these situations [73, 76]. In this thesis, we use the so called Generalized Approximate Message Passing (GAMP) [73] which has better convergence rate than other message passing algorithm for high density linear mixing problems.

Consider the QNC measurement equation of (3.14) where the elements of the total effective noise, $\{\underline{N}_{\text{eff,tot}}(t)\}_i$'s, are dependent. By eigenvalue decomposition of their covariance matrix,

$$\mathbf{E}[\underline{N}_{\text{eff,tot}}(t)\underline{N}^T_{\text{eff,tot}}(t)] = U_N(t) \cdot \Lambda_N(t) \cdot U^T_N(t), \tag{4.52}$$

we define:

$$\underline{Z}'_{\text{tot}}(t) = \Lambda_N^{-\frac{1}{2}}(t)U^T_N(t) \cdot \underline{Z}_{\text{tot}}(t), \tag{4.53}$$

and

$$\underline{N}'_{\text{eff,tot}}(t) = \Lambda_N^{-\frac{1}{2}}(t)U^T_N(t) \cdot \underline{N}_{\text{eff,tot}}(t), \tag{4.54}$$

for which we have:

$$\underline{Z}'_{\text{tot}}(t) = \Theta'_{\text{tot}}(t) \cdot \underline{S} + \underline{N}'_{\text{eff,tot}}(t). \tag{4.55}$$

In 4.55, $\Theta'_{\text{tot}}(t)$ is as follows:

$$\Theta'_{\text{tot}}(t) = \Lambda_N^{-\frac{1}{2}}(t)U^T_N(t) \cdot \Psi_{\text{tot}}(t) \cdot \phi, \tag{4.56}$$

and $\{\underline{N}'_{\text{eff,tot}}(t)\}_i$'s are uncorrelated with unit variance. Hence, we can model our QNC scenario (described by (4.55)) with a factor graph representation, as shown in Fig. 4.2. In this graph, each constraint node, $j$, $1 \leq j \leq n$, (blue node) is connected to a subset of variable nodes (white nodes), $i$, $1 \leq i \leq m$, for which $\{\Theta'_{\text{tot}}(t)\}_{i,j} \neq 0$.

The GAMP algorithm developed in [76] can be simplified to the following steps for decoding of quantized network coded messages in the Bayesian framework:

1. The variable nodes have their prior information, set as their initial value to start the algorithm with. Specifically, the estimated values for the mean and variance of $S_j$ are

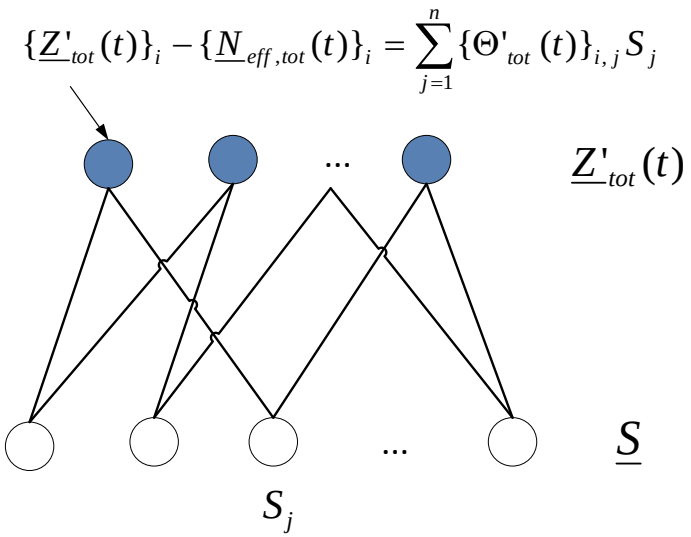



**Figure 4.2** QNC can be represented by using a factor graph from the sparse messages to the noisy measurements.

initialized by: $\hat{s}_j(\tau) = 0,\ \hat{\mu}_j^s(\tau) = \sigma^2_{\text{source}}$, where $\tau$ is the iteration step.

2. For each constraint node, $1 \leq i \leq m$, we calculate:

$$\begin{aligned}
\mu_i^p(\tau) &= \sum_{j'=1}^{n} |\{\Theta'_{\text{tot}}(t)\}_{i,j'}|^2 \mu_{j'}^x(\tau), \\
\hat{p}_i(\tau) &= \sum_{j'=1}^{n} \{\Theta'_{\text{tot}}(t)\}_{i,j'} \hat{s}_{j'}(\tau) - \mu_i^p(\tau)\hat{u}_i(\tau-1), \\
\hat{z}'_i(\tau) &= \sum_{j'=1}^{n} \{\Theta'_{\text{tot}}(t)\}_{i,j'} \hat{s}_{j'}(\tau),
\end{aligned}$$

with $\hat{u}_i(-1) = 0$.

3. For each $i$, we calculate:

$$\begin{aligned}
\hat{u}_i(\tau) &= (\{\underline{z}'_{\text{tot}}(t)\}_i - \hat{p}_i(\tau))/(1+\mu_i^p(\tau)), \\
\mu_i^u(\tau) &= 1/(1+\mu_i^p(\tau)).
\end{aligned}$$

4. For each variable node $1 \leq j \leq n$, we calculate:

$$\begin{aligned} \mu_j^r(\tau) &= \Big( \sum_{i=1}^{m} |\{\Theta'_{\text{tot}}(t)\}_{i,j}|^2 \mu_i^u(\tau) \Big)^{-1}, \\ \hat{r}_j(\tau) &= \hat{s}_j(\tau) + \mu_j^r(\tau) \sum_{i=1}^{m} \{\Theta'_{\text{tot}}(t)\}_{i,j} \hat{u}_i(\tau). \end{aligned}$$

5. For each $j$, we calculate:

$$\begin{aligned} \hat{s}_j(\tau+1) &= \sigma_{s,0}^2 \hat{r}_j(\tau)/(\mu_j^r(\tau) + \sigma_{s,0}^2), \\ \mu_j^s(\tau+1) &= \sigma_{s,0}^2 \mu_j^r(\tau)/(\mu_j^r(\tau) + \sigma_{s,0}^2). \end{aligned}$$

6. This procedure is repeated by the increment $\tau = \tau + 1$ and going back to step 2 until some convergence criterion, such as:

$$\|\underline{\hat{s}}(\tau) - \underline{\hat{s}}(\tau-1)\|_{\ell_2} \leq \epsilon_{\text{th}},$$

is met (where $\epsilon_{\text{th}}$ controls the precision of decoding). The final result $\underline{\hat{s}}(\tau)$ is used to calculate the GAMP estimation of messages: $\underline{\hat{x}}_{\text{GAMP}}(t) = \phi \cdot \underline{\hat{s}}(\tau)$.

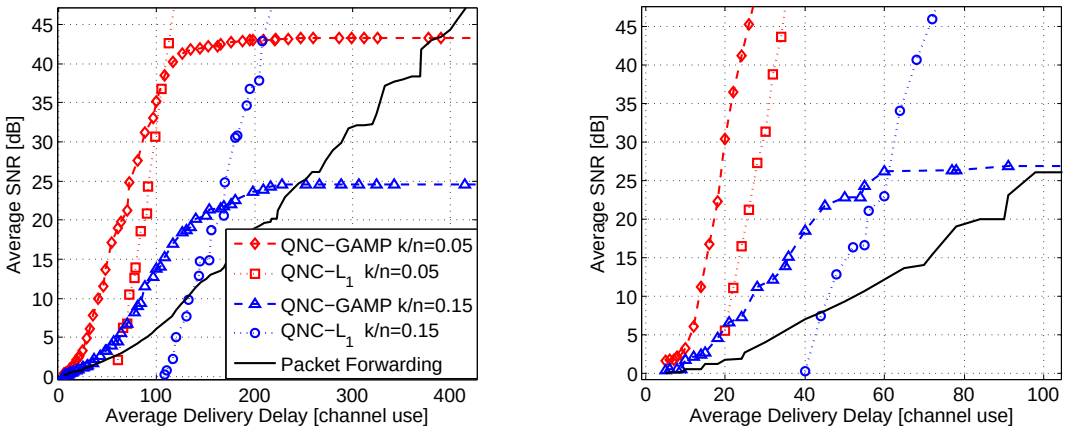
The major difference of GAMP algorithm from its previous message passing versions is that it reduces the vector values operations to a sequence of scalar operations. As a result, the computational complexity of the algorithm increases linearly with the size of problem as opposed to exponential increase in the case of standard belief propagation algorithm [76].

### 4.2.3 Simulation Results

In this section, we compare the performance of GAMP based decoding with that of $\ell_1$-min decoding. Specifically, we generate random deployments of networks with $n = 100$ nodes and $|\mathcal{E}| = 400, 1200$, as described in Section 3.4.1. For each deployment, we also randomly generate $\underline{S}$'s, with a mixture Gaussian distribution, as described in (4.39). The messages are derived from sparsity factors of $\frac{k}{n} = 0.05, 0.15$, and $\sigma_s^2 = 5,\ \sigma_0^2 = 0.05$. Furthermore, the sparsifying orthonormal matrix, $\phi$, is randomly generated, resulting in the generation of $\underline{X}$.

For each deployment, the network coding coefficients are generated similar to the simulations, described in Section 3.4. The resulting $\Psi_{\text{tot}}(t)$'s are dense Gaussian matrices, as Theorem 3.2.1. GAMP based decoding as well as $\ell_1$-min decoding are used to reconstruct the messages. The GAMP based decoder is as described in Section 4.2.2, and is implemented by using the implementation in [76]. The implementation of $\ell_1$-min decoding and routing-based packet forwarding is as described in Section 3.4.

For each SNR value, the best choice of block length $L$, is found for both QNC and packet forwarding scenarios and used to minimize the corresponding delivery delay. We present the results, by averaging them over different realizations of network deployments and messages. In Fig. 4.3, the resulting average SNR is depicted versus the average delivery delay.




(a) 400 edges (b) 1200 edges

**Figure 4.3** SNR versus delivery delay of QNC and QandPF scenarios: QNC-$L_1$ and QNC-GAMP refer to quantized network coding with $\ell_1$-min and GAMP-based decoding, respectively.

As shown in Fig. 4.3(a),4.3(b), using GAMP-based decoding helps us improve the decoding performance for the low SNR values, compared to $\ell_1$-min decoding. Moreover, when the sparsity factor $k/n$, of messages increases (meaning higher correlation between $X_v$'s), the gap between the QNC and packet forwarding curves increases. However, there is a drawback in using a GAMP-based decoder for high SNR values, especially when the sparsity factor of messages, $k/n$, is high (*i.e.* there is not a high correlation between $X_v$'s). Such cases can be explained to be a result of propagation of quantization noises, in the

network, which increases the noise power in the measurements. In fact, the used approximation in the modeling of noises in our QNC scenario may be a reason for the GAMP-based decoder to fail to work properly, in those regions.

## 4.3 Summary and Conclusion

In this chapter, we discussed the Bayesian framework for quantized network coding. Specifically, for a set of messages, characterized by a two-state Gaussian mixture model, we derived a bound on the decoding error of quantized network coding. This was done by describing a version of quantized network coding, called one-step quantized network coding. Further, for the Bayesian scenario, we discussed the use of Generalized Approximate Message Passing (GAMP) for decoding of quantized network coded messages. Our simulation results show that when the prior of the messages is known at the decoder side, using a GAMP-based decoder can result in higher decoding SNR than when a $\ell_1$-min decoder is used.

# Chapter 5

# Quantized Network Coding in Lossy Networks

In this chapter, we study quantized network coding in wireless sensor networks where the communication channels are lossy. Our study is done by using computer simulations in which all of the steps required to carry out the transmission are implemented. In the previous chapters, we used an abstract implementation of different transmission schemes. Although it lets us obtain an overall observation about the performance of the schemes, it does not consider many details in practical real implementations.

Similar to other wireless technologies, the communication of wireless sensor networks have also been standardized. Specifically, the way the content of the packets is converted into the physical signals at the transmitter nodes and then decoded at the receiver nodes needs to be characterized. In this chapter, we study the performance of quantized network coding and packet forwarding scenarios by putting them to the test of an industry standard for wireless sensor networks. As the IEEE 802.15.4 is the industry standard for low rate transmission in wireless personal area networks, we will use it for testing quantized network coding and packet forwarding scenarios.

## 5.1 Physical Channel Modeling

Before we discuss the IEEE 802.15.4 standard, we first describe the model used for simulation of the physical channel. In this section, we describe the way physical signals are propagating from their transmitter nodes in the air. Specifically, we use the model de-

scribed in [77] for simulating the propagation of the signals in the air. The model is based on the widely used exponential decay of the power through the medium. In [77], the authors have also obtained typical values of the associated parameters via real experimental measurements.

The physical model described below includes the effect of transmitter and receiver antennas. Further, we assume that the channel fading is slow. Therefore, the received signal at a receiver is a shifted-and-attenuated version of its original transmitted signal. The time-shift depends on the propagation delay of the physical media which is assumed to be negligible with respect to the delays associated with time-scheduling and queuing in the transmission. The attenuation of the signal depends on the decay model which characterizes the medium.

In [77], the path loss is modeled as follows:

$$P_{v_1 \to v_2} = P_{v_1,\mathrm{TX}} - 10 c_{\mathrm{att}} \log_{10} \Big( d(v_1, v_2) \Big) - c_{\mathrm{att},0} - C_{v_1 \to v_2,\mathrm{fading}} \quad (\mathrm{dBm}) \ , \tag{5.1}$$

where $d(v_1, v_2)$ is the physical distance between node $v_1$ and node $v_2$ in meters. In (5.1), $P_{v_1 \to v_2}$ is the power of received signal at node $v_2$ from node $v_1$ and $P_{v_1,\mathrm{TX}}$ is the power of transmitted signal from node $v_1$ in dBm. As described in Section 3.4, $c_{\mathrm{att}}$ and $c_{\mathrm{att},0}$ are the attenuation exponent and attenuation at one meter, respectively. Further, $C_{v_1 \to v_2,\mathrm{fading}}$ models the slow shadow fading, modeled by a zero mean Gaussian random variable with variance of $\sigma^2_{\mathrm{att}}$.

As it is described in the next section, the signals are modulated around their center frequencies. In our model, we assume that the signals of a center frequency are perfectly isolated from the signals of other frequencies. The signals with the same center frequency are added over the air. The additive noise that we use for our simulations is assumed to be Gaussian and white with the power $\sigma^2_{\mathrm{noise}}$. Usually the noise is the thermal noise of the devices and has a minimum power of $-115$ dBm, as investigated in [78]. At a specific center frequency, the receiver measures a signal which is the sum of attenuated versions of transmitted signals from different nodes and an additive noise. The list of parameters in the physical channel model used for our simulations is presented in Table 5.1.

**Table 5.1** Values of the parameters of the channel loss model obtained in a ground plain environment.

| Parameter | Value |
|---|---|
| Attenuation Exponent $c_{\text{att}}$ | 3.12 |
| Attenuation at One Meter $c_{\text{att},0}$ | 35.4 |
| Fading Parameter $\sigma_{\text{att}}$ | 1.83 |
| Physical Noise Power $\sigma^{dBm}_{\text{noise}} = 10\log_{10}(\sigma^2_{\text{noise}})$ | $-90 \ldots -75$ dBm |

## 5.2 Review of IEEE 802.15.4 Standard

The IEEE 802.15.4 standard describes the requirements for the MAC and PHY layers for low rate wireless private networks (LR-WPAN). Because of its simplicity in implementation and very low computational cost and power usage, IEEE 802.15.4 is intended to be suitable for sensor networks. One possible choice of network and application layers for wireless sensor networks is what is described by ZigBee Alliance.[1] In this chapter, we only consider the IEEE 802.15.4 recommendations for the implementation of PHY and MAC layers. In the following, we provide a brief review of IEEE 802.15.4 standard for wireless sensor networks.

### 5.2.1 General Architecture

In the standard, an LR-WPAN supports the deployment of two types of devices, called Full Function Device (FFD) and Reduced Function Device (RFD). An FFD may be a Personal Area Network (PAN) coordinator, or a coordinator, which requires a higher level of implemented features than RFD. The RFDs, according to the standard, are in charge of simple tasks in the network and should only implement a limited set of features of the IEEE 802.15.4 standard. Depending on the network topology, each PAN requires one or more FFDs, to act as its coordinator(s).

The standard supports two types of network topologies for the configuration of its nodes: the star topology and the peer-to-peer topology. In the star topology, only one of the devices has to be FFD and would act as the PAN coordinator. The rest of the devices will be attached to the PAN coordinator directly and can only be RFDs. In contrast to its simplicity, the star topology is not suitable when some of the sensor nodes are not in the communication range of the coordinator.

[1] ZigBee alliance is an organization which has been collaborating with the IEEE to specify a full protocol for low power, low cost and low data rate wireless communication.

The peer-to-peer topology has more flexibility because it does not need a centralized communication architecture. All the nodes can directly communicate with each other, conditioned on being in radio range. Other nodes may obtain end-to-end connectivity through multiple hops. As a result, there may be a need for more FFDs than the PAN coordinator. Although there is a PAN coordinator in this topology, its existence is not as important as it is in a star topology. As a combination of star and peer-to-peer topologies, a *cluster-tree* can also be formed by the devices in a sensor network.

### 5.2.2 Physical Layer Specifications

The PHY layer of 15.4 standard is in charge of the following tasks:

- activation and deactivation of the radio transceiver,
- energy detection within the current channel,
- Clear Channel Assessment (CCA),
- channel frequency selection,
- data transmission and reception over the channel.

Since different modulation schemes may be used for a specific frequency in a physical layer, the pair of frequency and modulation is denoted by special binary strings. To support the growing number of channels and their different modulation schemes, the combination of *channel page* and channel number is used [20].[2] In the standard, the 5 most significant bits of the channel bitmap with 32 bits specify the channel page.

The list of channel frequencies and their modulation scheme for channel page 00000 (which is used by the commercial users) is presented in Table 5.2. Among three different frequency ranges, listed in Table 5.2, only the 2400-2483.5 MHz range is reserved for wireless personal area networks internationally and is set to be used for most devices. The 868 MHz frequency and 902-928 MHz frequency range are allocated for use only in Europe and America, respectively. In the following, we will discuss the spreading and modulation schemes, used for transmission of information over the physical channel.

[2]This is because of the introduction of 868/915 MHz amplitude shift keying and O-QPSK specifications which results in the total number of channel assignments exceeding the channel numbering capability of 32 channel numbers, defined in the earlier edition of the standard.

**Table 5.2** IEEE 802.15.4 frequency bands in channel page 0000

| Frequency [MHz] | Channels | Chip Rate [kchip/sec] | Modulation | Bit Rate [kbps] |
|---|---|---|---|---|
| 868 - 868.6 | 1 | 300 | BPSK | 20 |
| 902 - 928 | 10 | 600 | BPSK | 40 |
| 2400 - 2483.5 | 16 | 2000 | O-QPSK | 250 |

The data rates which can be achieved for each frequency and modulation scheme are different from their corresponding chip rate. The use of chips is explained further in detail for O-QPSK modulation in $2500 - 2483.5$ frequency range.

### 2450 MHz Frequency Spreading and Modulation

The PHY layer converts the string of bits (coming from its higher layers) to the modulated signals. These bit strings that are served by the PHY are called PHY Protocol Data Unit (PPDU). The abstract architecture for the 2450 modulator, proposed in the 15.4 standard, is shown in Fig. 5.1. According to this design, every four consecutive bits of PPDU are mapped to a string of 32 *chip* bits. The chip bits are strings of pseudo-random bits, helping the PHY to prevent channel errors, acting as a channel coding scheme. These chip strings are specified in the IEEE 802.15.4 standard (Table 24 in [20]). Then, the resulting chip bits are then transmitted over the physical channel via O-QPSK modulation.


**Figure 5.1** Reference modulator diagram for 2450 MHz frequency rage .

The diagram of Offset Quadrature Phase Shift Keying (O-QPSK) modulator is presented in Fig. 5.2. As shown in this figure, the chip bits are divided into even-indexed and odd-indexed bits and go through different phases, called I and Q phases of modulation, respectively.

Denoting the continuous-time by[3] $t_{\text{PHY}}$, the even indexed bits are passed through I-

[3]We use the notation $t_{\text{PHY}}$ to denote continuous time; despite the fact that it is slightly cumbersome, it allows us to alleviate any confusion with the variable $t$ that has been used up to now for representing the discrete index of QNC step.

phase where each bit is modulated using a half sine pulse $p_{\text{pulse}}(t_{\text{PHY}})$, described as:

$$p_{\text{pulse}}(t_{\text{PHY}}) = \begin{cases} sin(\pi \frac{t_{\text{PHY}}}{2T_c}) & , \ 0 \leq t_{\text{PHY}} \leq 2T_c \\ 0 & , \ \text{otherwise} \end{cases} , \tag{5.2}$$

where $T_c$ is the inverse of chip rate. More precisely, each 1 is represented by a positive pulse $+p_{\text{pulse}}(t_{\text{PHY}})$ and each 0 is represented by a negative pulse $-p_{\text{pulse}}(t_{\text{PHY}})$ (called NRZ[4] encoding). The result is the envelope of I-phase signal after being multiplied by the carrier signal. The odd-indexed bits are passed through the Q-phase where they are initially delayed by half of $T_c$ and then modulated. The adopted delay in the Q-phase distinguishes the Offset QPSK from normal QPSK modulation, resulting in a relatively more constant envelope than normal QPSK.

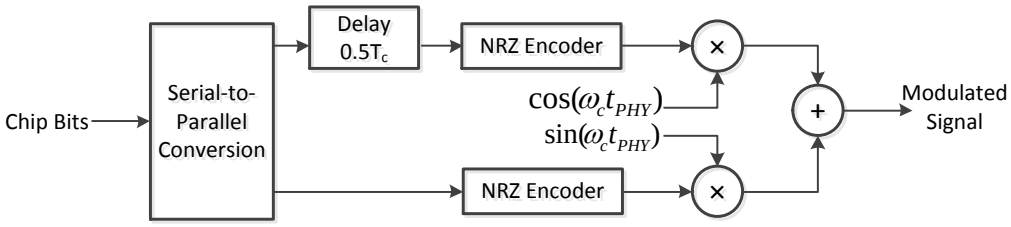



**Figure 5.2** O-QPSK Modulator Diagram: $\omega_c$ is the center frequency of the carrier and $T_c$ is the inverse of the chip rate.

### Receiver Design

According to the standard (Section 6.1.7 of [20]), the receiver has to be designed such that the average Packet Error Rate (PER) is below one percent. The units of packet used for this purpose are PHY Service Data Units (PSDU), containing the payload of PHY layer packets, each with a size of 20 octets (160 bits). Although these receiver sensitivity parameters are specified in the standard, there is no architecture specified for the receiver side of PHY layer in the standard.

Assuming perfect synchronization of the receiver, the optimal receiver design is such that it uses $2^4 = 16$ matched filters to calculate the correlation of received signal with the potential transmitted signals for each symbol. Such receiver design, shown in Fig. 5.3, is also used for implementation purposes in [79].

[4]Non Return to Zero

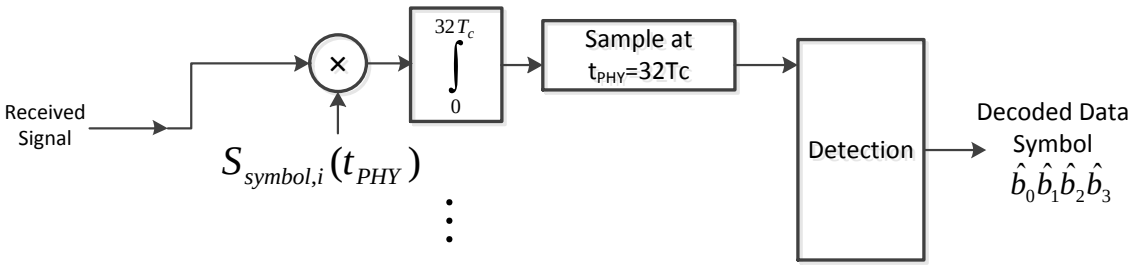



**Figure 5.3** PHY receiver design for 2450 MHz: $S_{symbol,i}$'s with $i = 0, \ldots, 15$ are the modulated signal resulting from different symbols.

### Measuring Channel State

As mentioned earlier, the PHY layer has direct access to the physical media and is able to provide control information about the physical channels to other layers of the standard. Specifically, PHY is in charge of the following tasks for its FFDs:

- *Energy Detection (ED):* PHY needs to estimate the received signal power within the bandwidth of IEEE 802.15.4 channels. This task does not need to identify or decode the received symbols.
- *Link Quality Identification (LQI):* This feature measures the quality of a received packet and can be implemented using ED or other methods. Although this information is not used for any part of IEEE 802.15.4 standard, it may be used in upper layers of the network.
- *Clear Channel Assessment (CCA):* This feature is essential for channel access mechanism, especially in the CSMA-CA[5] procedure.

It should also be noted that these tasks are essential to the channel frequency selection.

### 5.2.3 Functionality of MAC layer

The MAC layer is in charge of stuffing of the packets and in the IEEE 802.15.4 different types of MAC packets may be generated:

- Data frames which carry the information between the nodes,

[5]Carrier Sense Multiple Access with Collision Avoidance

- Beacon frames which carry a beacon and are used for synchronization and identification purposes,
- ACK frames which are used to acknowledge the receipt of a data frame to its transmitter,
- Control frames which are used to inform the different nodes about the networking parameters.

The MAC layer of the standard is in charge of bit stuffing of the packets and transmission and reception of MAC Protocol Data Unit (MPDU) across PHY. Furthermore, the MAC layer is in charge of the following tasks:

- beacon management,
- channel access,
- Guaranteed Time Slot (GTS) management,
- frame validation,
- acknowledged frame delivery,
- association and disassociation of users.

The operation of MAC in the standard may be done with or without the use of beacons. Beacons are periodically generated packets by the coordinator that are used for the synchronization and identification purposes. In the beacon-enabled mode, the beacons limit the range in which a round of uplink or downlink communication may be made for all the nodes, attached to a coordinator. This is explained in the following section where the superframe structure is presented. In the non beacon-enabled mode, all the nodes can simply send their data by using unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) protocol.

### Superframes

Low rate wireless personal area networks (LR-WPAN) are allowed to use a structure of the packets, called *superframe*. Each superframe is built up from multiple MAC frames.

Specifically, each superframe includes 16 equally sized time slots, bounded between two network beacon frames. The beacons are sent by a coordinator of the PAN and are used to synchronize the associated devices, to identify the PAN and to describe the structure of the superframe. Optionally, part of the slots between two beacons may be left un-used and the transceivers be in their inactive mode.

For the 16 slots in the active period of a superframe, some are dedicated for specific applications/users with low latency requirements, called Guaranteed Time Slots (GTS), in the *contention free* period (CFP). The rest of slots belong to the *contention access* period (CAP) where the users compete with each other for gaining channel access, via CSMA-CA mechanism.

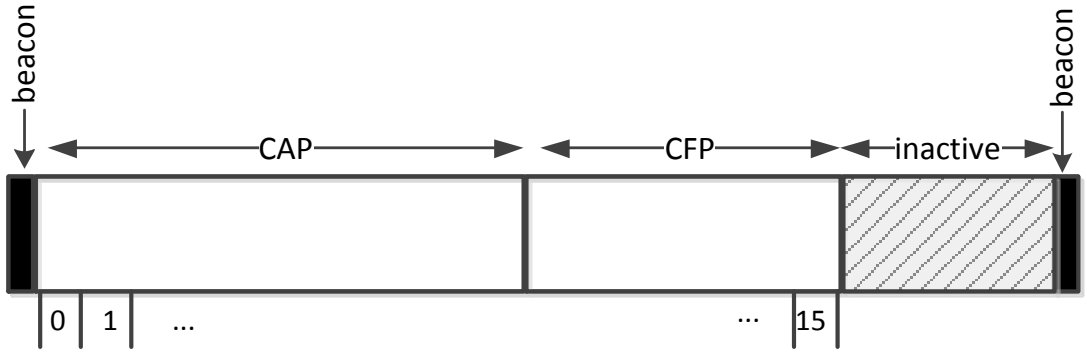



**Figure 5.4** The structure of a superframe.

### CSMA-CA Mechanism

The CSMA-CA mechanism is based on the concept of using random backoff periods. Basically, a user starts its transmission once it has detected that the channel is idle (not being used for transmission by other nodes) for a period of time, called Contention Window (CW). If the channel is not idle, the node would start waiting for a random backoff period and try again (checking the channel state and transmitting then). Moreover, the expected backoff period used by a node gets increased exponentially when the CCA reports a busy channel. The description of CSMA-CA mechanism with explicit use of parameters is presented in Section 7.5.1.4 in [20].

#### Maintenance of PAN

Among different PAN maintenance tasks, done by the MAC layer, we are interested in the association of the nodes and the maintenance of GTSs. Depending on the network topology, used for the PAN, there may be the need for one or more coordinators. In a star topology, the first node to start working and sending beacons can take charge as the PAN coordinator. Upon the detection of PAN coordinator beacon (and its identifier), the rest of the nodes will associate themselves with the PAN coordinator. If a node is not in the communication range of the adopted PAN coordinator and does not detect its beacon, then it may consider itself as a PAN coordinator and start sending beacons.

In the peer-to-peer topology, more than one of the nodes may be acting as the coordinator, although there is only one unique coordinator for the whole PAN. The very first node to start sending beacons and identifying itself as the coordinator is usually set as the PAN coordinator. In contrast with the star topology, the PAN coordinator in the peer-to-peer topology does not play a much more important role than other coordinators. After the first coordinator has started its beacon transmission and is discovered by its neighboring nodes, they may also start acting as a coordinator and start sending their beacons. This way, through the multiple hops of transmission, all the nodes will be able to have an end-to-end communication link. A node which is already associated a coordinator may start its beacon transmission (and act as a new coordinator) in a different frequency channel as the previously established beacon. This is done after an active channel scan has been performed over all the frequency channels.

### 5.2.4 Network Diversity

The formation of sensor networks using IEEE 802.14.5 standard may result in loosing the advantages of having network diversity. For example, consider a star topology in which two leaf nodes communicate via the PAN coordinator. In such case, if the link between one of the leaf nodes and the coordinator breaks, the communication is disturbed until a maintenance mechanism is adopted. However, the nodes have more than one possible communication link and if more frequency channels were adopted, the network diversity avoids situations where a link failure causes temporarily (or permanently) isolated nodes. In the following, we describe a scenario where IEEE 802.15.4 mechanisms are recruited to establish full diversity in a sensor network.

## 5.3 Adapting QNC within IEEE 802.15.4

To implement quantized network coding and even finite field network coding over the IEEE 802.15.4 standard, we need to make changes to the MAC layer of the standard. As it was discussed in Section 5.2.3, the MAC layer controls the access to the channel in order to prevent interference. In contrast with the routing-based packet forwarding in which the nodes only transmit to their corresponding coordinator, network coding may need to transmit to many of its neighbour nodes. Such requirement can fit into a peer-to-peer networking topology but needs accurate allocation of time and frequency slots, as discussed in the following section. Further, a node has to send its packets to a certain number of nodes at all the steps of QNC. This requirement can be carried out by a GTS based channel access mechanism.

### 5.3.1 Frequency and Time Allocation

As the principal equation of quantized network coding (3.2) shows, a node needs to transmit quantized linear combinations to all of its outgoing nodes. To do this, each node will transmit its QNC packets to its neighbouring nodes by time and frequency allocation over the available frequency range for the standard. Specifically, as we mentioned in Section 5.2.2, there are a total of 16 frequencies allocated for IEEE 802.15.4 standard. In a small network where the number of nodes is less than or equal to this maximum number of channels, each node can take over a unique frequency and use it for transmission of its packets. However, in many practical cases there are more nodes than 16 and we need to share the frequencies between different nodes in an efficient way.

Intuitively, we may propose to use each frequency in a scheduled period of of time at a single node for transmission. Basically, this would be a time-sharing to access a channel by different nodes to have *no interference* from other nodes. However, an intermediate approach in which the same frequency is used for multiple nodes at the same time may be applicable.

Essentially, the interference can be treated as a noise at the receiver nodes. If the power of the interference is too high then the receiver node of a signal may not be able to successfully decode it into a packet and a packet drop happens.[6] To this end, we formulate the problem of frequency and time allocation into the *graph coloring* problem.

[6]It should be noted that the probability of decoding a packet with an error is very low.

In the graph coloring problem, one needs to find the minimum number of colors for coloring the nodes such that no connected pair of nodes have the same color. There are several algorithms to find the solution to the graph coloring problem for a given graph of nodes and edges. In this thesis, we use a MATLAB implementation of a greedy algorithm, discussed in [80].

We define an interference graph, denoted by $\mathcal{G}_{\text{interference}} = (\mathcal{N}, \mathcal{E}_{\text{interference}})$, where $\mathcal{E}_{\text{interference}}$ is the set of pairs of connected nodes in $\mathcal{G}_{\text{interference}}$. The set of nodes is the same as in our original graph representing the sensor network. Two nodes $v_1$ and $v_2$ in the interference graph are connected if the average power of received signal at node $v_2$ from node $v_1$ is above a certain threshold $P_{\text{interference}}$. We assume that all of the nodes use the same transmission power:

$$P_{v,\text{TX}} = P_{\text{TX}}, \ \forall v \in \mathcal{V}. \tag{5.3}$$

Denoting the pairwise distance of $v_1$ and $v_2$ by $d(v_1, v_2)$ (in meters), the average received powers received at $v_2$ from $v_1$ (and received at $v_1$ from $v_2$) is modeled similar to the average power decay model used in (3.60). Hence, the average power received from $v_1$ at $v_2$, $\bar{P}_{v_1 \to v_2}$ is equal to the average power received from $v_2$ at $v_1$, $\bar{P}_{v_2 \to v_1}$ and are denoted by $\bar{P}_{v_1,v_2}$ such that:

$$\bar{P}_{v_1,v_2} = \bar{P}_{v_1 \to v_2} = \bar{P}_{v_1 \to v_2} = P_{\text{TX}} - 10 \ c_{\text{att}} \ \log_{10}\Big(d(v_1, v_2)\Big) - c_{\text{att},0} \ \ (\text{dBm}). \tag{5.4}$$

In (5.4), $P_{\text{TX}}$ is the transmission power used at all of the nodes in dBm. In the interference graph $\mathcal{G}_{\text{interference}}$, two nodes $v_1$ and $v_2$ are connected, if

$$\bar{P}_{v_1,v_2} \geq P_{\text{interference}}. \tag{5.5}$$

After running the graph coloring on $\mathcal{G}_{\text{interference}}$, each node $v$ is assigned a positive integer $R_{\text{time-freq}}(v)$ which characterizes the time and frequency sharing as explained below. Depending on the connectivities of the interference graph, the resulting indices $R_{\text{time-freq}}(v)$'s may be less than or equal to the maximum number of available frequencies; *i.e.* 16. In that case, each node is assigned a different frequency and all of the nodes can transmit at the same time. If the maximum of the calculated $R_{\text{time-freq}}(v)$ is more than the maximum number of frequencies then time sharing should also be adopted.

Denoting the time and frequency indices at which node $v$ can transmit by $R_{\text{time}}(v)$ and

$R_{\text{freq}}(v)$, we have:

$$R_{\text{time}}(v) = \left\lceil \frac{R_{\text{time-freq}}(v)}{16} \right\rceil, \tag{5.6}$$

$$R_{\text{freq}}(v) = \Big( (R_{\text{time-freq}}(v) - 1) \mod 16 \Big) + 1, \tag{5.7}$$

where $\lceil . \rceil$ represents rounding up. The maximum value of $R_{\text{time}}(v)$ over $v$ determines how many time sharing slots are required to complete inter-node transmissions for one step of QNC. The time sharing at a single frequency is illustrated in Fig. 5.5.

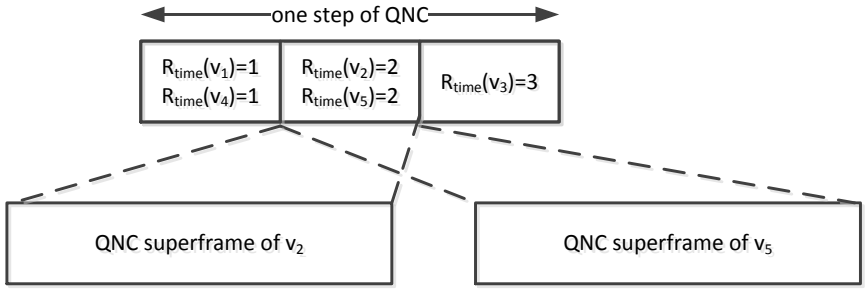



**Figure 5.5** Time sharing of a frequency for QNC scenario: Each node $v$ sends its QNC downlink superframe according to $R_{\text{time}}(v)$ at a specific QNC step $t$. As shown, there may be more than one node to be scheduled to use the frequency at a single time because they have little interference on each other's transmissions.

The choice of $P_{\text{interference}}$ introduces a trade-off to our scenario. A high value of $P_{\text{interference}}$ would result in an interference graph with few connections. Hence, the maximum value of $R_{\text{time}}(v)$ is small and the delay to transmit packets between the nodes is small. However, on the other hand, since the power of interfering signals is higher, the chances of packet drop is high.

The average power of received signals from the nodes that are considered to have a link to the receiver node (as discussed in Section 3.4.1) is a good measure to set the value of $P_{\text{interference}}$. In our simulations, we use a value of $P_{\text{interference}}$ which is 5 dB less than the average power of received signals from the connected nodes.

### 5.3.2 GTS based inter-node Transmissions

For each step of QNC with index $t$, all of the packets with the quantized network coded contents should be transmitted to their receiver nodes. A specific node $v$ acts as a transmitter according to the time and frequency allocation, described in Section 5.3.1. At the appropriate time slot, the transmitter sends its packets by building up a MAC superframe, described in Section 5.2.3.

At each time step of QNC, we transmit the packets of each node by using a superframe structure similar to the standard MAC superframe. In the standard, there are a maximum of 16 time slots in which a MAC packet can be transmitted. The standard requires us to allocate a minimum length for the contention access period to be used for transmission of MAC control frames.[7] Therefore, some of those 16 slots are either used by the control frames or left unused. Further, in some cases the number of packets that a node has to send is more than 16. Hence, the node has to send its packet by using multiple number of MAC superframes.The number of required MAC superframes (and beacons) is determined by the maximum number of outgoing edges (outgoing packets at a single $t$) in the network. The structure of superframe used for transmission of QNC packets from a node is illustrated in Fig. 5.6.

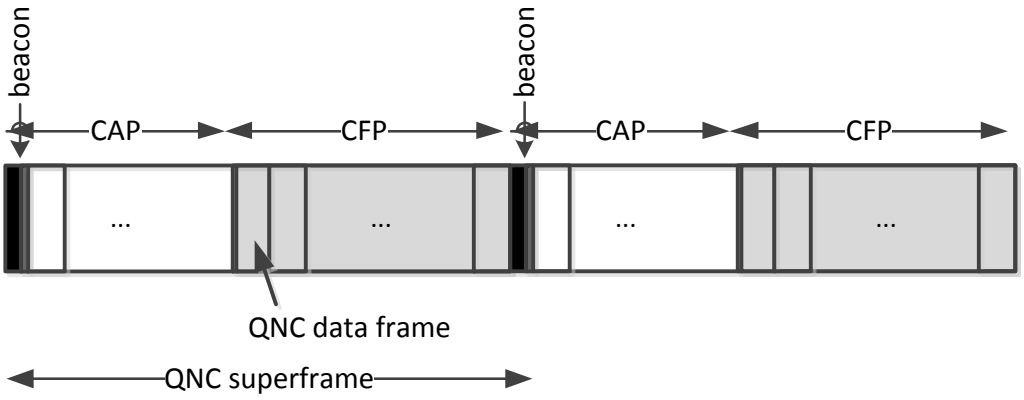



**Figure 5.6** Superframe used for downlink of quantized network coded packets from a node: The maximum number of outgoing edges is less than or equal to 32 in this case.

The *QNC data frame* is a MAC data frame where its payload is the binary string of

[7]The minimum length of CAP has to be 440 symbols which is equal to 110 Bytes when 2450 MHz frequency range is used.

corresponding quantized network coded value. Specifically, for a packet generated at node $v$ to be sent to the node $v' = head(e)$ with $e \in Out(v)$, $Y_{e,tx}(t)$ (calculated according to (3.2)) is converted to binary and the result is put as the payload of the corresponding MAC data frame. The size of QNC data frames is the same and is determined by the value of packet length parameter $L$ used in QNC scenario.

Since the links are lossy, some of the packets may be lost during the transmission. In the adapted MAC design for QNC scenario, we do not try to re-send those dropped packets. Further, the successful or unsuccessful transmission of packets is not acknowledged to its sender by an ACK frame. If a packet is not successfully retrieved at its designated receiver node, it is considered as being dropped. This is further explained in the next section where we discuss the decoding of the messages.

### 5.3.3 $\ell_1$-min Decoding with Packet Loss

As it was mentioned, the lost packets during the inter-node transmissions are not re-sent. Hence, the lost information would contribute to the total effective measurement noise. The correlation between the messages makes the decoder used in the QNC scenario able to decode the messages even in such cases. However, this requires a modification of the upper bound on the $\ell_2$ norm of total effective measurement noise vector. Specifically, if the bound used during the $\ell_1$-min decoding (3.54) is smaller than the real $\ell_2$ norm of the error vector, the decoder may not be able to find any solution. In this section, we modify the $\ell_1$-min decoding algorithm (3.48) to take into account the packet drops in the lossy networks.

If a packet transmitted over the link $e$ at QNC step $t$ is dropped, there will be an error in $Y_{e,rx}(t)$ with respect to $Y_{e,tx}(t)$. However, since the quantized network coded contents are bounded between $-q_{\max}$ and $+q_{\max}$, the error resulting from the packet drop is also bounded. Specifically, if we pick

$$Y_{e,rx}(t) = 0 \tag{5.8}$$

when the corresponding packet is lost, then we have:

$$|Y_{e,rx}(t) - Y_{e,tx}(t)| \leq q_{\max}. \tag{5.9}$$

We assume that the packet losses at different links and at different QNC steps are independent and identically distributed. Further, we model the packet loss by using a binary

random variable where the probability of packet loss is denoted by $p_{\text{loss}}$.

The event that the packet transmitted over link $e$ at QNC step $t'$ is lost, is denoted by $(e,t') \in \mathcal{L}_t$ where $\mathcal{L}_t$ is the set of lost packets up to $t$. Explicitly, this set $\mathcal{L}_t$ is such that:

$$\mathcal{L}_t \subseteq \{(e,t') | e \in \mathcal{E}, 2 \leq t' \leq t\}. \tag{5.10}$$

By using the results for the binominal distribution, for any positive integer $\lambda_{\text{loss}}$ we have:

$$P(|\mathcal{L}_t| < \lambda_{\text{loss}}) = \sum_{\lambda=0}^{\lambda_{\text{loss}}} \begin{pmatrix} (t-1)|\mathcal{E}| \\ \lambda \end{pmatrix} p_{\text{loss}}^{\lambda} (1-p_{\text{loss}})^{(t-1)|\mathcal{E}|-\lambda}, \tag{5.11}$$

where the right hand side is the cumulative probability mass function of binominal PDF.

For a given $\mathcal{L}_t$, we can derive an upper bound on the norm of total effective measurement noise. This can be done similar to the steps taken to derive (3.54). Specifically, for a given $\mathcal{L}_t$, the $\ell_2$ norm of total effective measurement noise can be upper bounded by $\epsilon_{\text{rec,loss}}(\mathcal{L}_t)$ where:

$$\epsilon^2_{\text{rec,loss}}(\mathcal{L}_t) = \frac{1}{4} \sum_{t'=2}^{t} \sum_{e \in In(v_0)} \Big( \sum_{t''=2}^{t'} \sum_{e'=1}^{|\mathcal{E}|} |\{F_{\text{prod}}(t''+1;t')\}_{e,e'}| \; \zeta(e',t'') \Big)^2, \tag{5.12}$$

and

$$\zeta(e',t'') = \begin{cases} \Delta_Q & , \; (e',t'') \in \mathcal{L}_t \\ q_{\max} & , \; (e',t'') \notin \mathcal{L}_t \end{cases}. \tag{5.13}$$

This provides an upper bound for the total measurement noise if the decoder knows which packets are lost during the transmission.

If the decoder only knows that the maximum number of packets lost during the transmission is bounded by $\lambda_{\text{loss}}$; *i.e.* $|\mathcal{L}_t| \leq \lambda_{\text{loss}}$, then the upper bound used for $\ell_1$-min decoding has to hold for different possibilities of $\mathcal{L}_t$. This can be ensured by maximizing the right hand side term in (5.12), resulting in the following upper bound on the $\ell_2$ norm of total measurement error:

$$\epsilon^2_{\text{rec,loss}}(t) = \max_{\substack{\mathcal{L}_t \\ |\mathcal{L}_t| \leq \lambda_{\text{loss}}}} \frac{1}{4} \sum_{t'=2}^{t} \sum_{e \in In(v_0)} \Big( \sum_{t''=2}^{t'} \sum_{e'=1}^{|\mathcal{E}|} |\{F_{\text{prod}}(t''+1;t')\}_{e,e'}| \; \zeta(e',t'') \Big)^2, \tag{5.14}$$

where $\zeta(e', t'')$ is as defined in (5.13) for each set $\mathcal{L}_{t'}$. Now, by using the probability in (5.11) we can present the following.

**Corollary 5.3.1** *Consider a QNC scenario with lossy links where the packet losses are independent with a probability $p_{\text{loss}}$. In such a scenario and for a given positive integer $\lambda_{\text{loss}}$, we have:*

$$\|\underline{Z}_{\text{tot}}(t) - \Psi_{\text{tot}}(t)\underline{X}\|_{\ell_2} \leq \epsilon_{\text{rec,loss}}(t), \tag{5.15}$$

*with a probability of*

$$\sum_{\lambda=0}^{\lambda_{\text{loss}}} \begin{pmatrix} (t-1)|\mathcal{E}| \\ \lambda \end{pmatrix} p_{\text{loss}}^{\lambda}(1-p_{\text{loss}})^{(t-1)|\mathcal{E}|-\lambda}, \tag{5.16}$$

*where $\epsilon_{\text{rec,loss}}(t)$ is defined in* (5.14). ■

The choice of $\lambda_{\text{loss}}$ introduces a trade-off for the design of $\ell_1$-min decoder. A small value of $\lambda_{\text{loss}}$ results in a small $\epsilon_{\text{rec,loss}}(t)$. However, the number of dropped packets in reality may be much higher than $\lambda_{\text{loss}}$. This means that the bound in (5.14) may be violated in real situations and $\ell_1$-min decoder may not be able to find any solution. On the contrary, if we pick a large value for $\lambda_{\text{loss}}$, the resulting bound in (5.14) may be too large, resulting in a large decoding error (see Theorem 3.3.2).

## 5.4 Simulation Results

In this section, we discuss the simulation results obtained by implementing QNC and Quantization and Packet Forwarding (QandPF) scenarios (described in Section 3.4) in networks with noisy links. We implement these two scenarios over the MAC and PHY layers of the IEEE 802.15.4 standard, as described in the following. We evaluate the adapted QNC scenario within the IEEE 802.15.4 by comparing its resulting decoding SNR, transmission energy, and delay with those of QandPF scenario. The messages and the network deployments are generated according to the model described in Section 3.4.

For our lossy simulations, we pick the decoder node such that it is in the center of the region in which network is deployed.This is denoted by $GW_{\text{center}}$ in Section 3.4. We perform the simulations for two values of $R_{\text{dim}} = 250$ and $R_{\text{dim}} = 400$ which results in networks with different number of links. The physical medium used for our simulations is based on the

model described in Section 5.1. The messages are generated such that they are bounded between $-10$ and $+10$ which means $q_{\max} = 10$. The list of the values of the parameters used in the generation of messages and network deployments is presented in Table 5.3.

**Table 5.3** Parameters and their values used for our simulations of networks with lossy links.

| Parameter | Value(s) |
|---|---|
| Sparsity Factor $k/n$ | $0.05, 0.10, 0.15$ |
| Near-Sparsity Parameter $\epsilon_k$ | 0 |
| Number of Nodes $n$ | 50 |
| Deployment Range $R_{\text{dim}}$ | $250, 400$ meters |
| Average TX Power $P_{i,\text{TX}}$ | 2.5 dBm |
| RX Sensitivity | $-95$ dBm |
| Noise Power $\sigma^{dBm}_{\text{noise}}$ | $-85, -82, -79$ dBm |
| Packet Length $L$ | $8, 16, 24, 32, 40$ bits |
| Range of Messages $q_{\max}$ | 10 |

### 5.4.1 Implementation of QandPF

To implement QandPF scenario, we use the MAC and PHY layers of the IEEE 802.15.4 standard without any changes. In the standard, it is described how the nodes initiate the networking and exchange MAC control packets to set up the hierarchy in the network. The maintenance of the PAN was briefly described in Section 5.2.3.

We assume that before the start of data gathering, the decoder node has received enough packets to know the connectivities in the network. This will allow the decoder to calculate the routes from each node to the decoder node. We also assume that the route information has been sent to the nodes before the start of the data gathering. In our simulations, we do not consider the delay associated with the initial maintenance and calculation of the routes in the PAN.

After the PAN is initiated, the nodes have to be grouped in hierarchical clusters of nodes. Specifically, in each cluster, there is a coordinator node which receives the packets from the rest of the nodes. The coordinator relays the received packets as well as its own packet to the next coordinator in the hierarchy. This is carried out in the whole network until all of the packets have arrived at the decoder node.

Before a real-valued message $X_v$ is packed into a MAC data frame, it is quantized by using a *uniform* quantizer. The step size of the quantizer is determined by the specified value of the packet length $L$. The simulations for lossless networks in the previous chapters were done for all integer values of $L$ from 1 to 40 bits. On the contrary in this section, we can only perform simulations for values of $L$ that are integer multiples of 8. This restriction comes from the fact that the size of payload of MAC packet in the IEEE 802.15.4 standard is always an integer number of bytes (8 bit). In our simulations, we use $L = 8, 16, 24, 32, 40$ bits.

When a node is assigned to a cluster with a specific coordinator and a specific frequency channel, it competes with other nodes to gain access to the channel via the CSMA/CA mechanism. Specifically, we implement a beacon-enabled version in which beacon frames are periodically sent from the coordinator. It should also be noted that the coordinator of a cluster is not determined based on the time at which the nodes start propagation of beacons (as recommended in the IEEE 802.15.4 standard). In our simulation, the coordinator is the node to which the remaining nodes of the cluster forward their packets.

In our simulations, there is no limitation on the queue in which the received packets are stored. We only make sure that there are not multiple copies of the same packet in the queue. This may happen if a packet is successfully received at the receiver node but its acknowledgement is not received at the sender node. The list of different parameters used for performing CSMA/CA procedure for channel access is presented in Table 5.4. We repeat the simulations for many different deployments and collect different resulting

**Table 5.4** Parameters used in CSMA/CA implementation for QandPF.

| Parameter | Value(s) |
|---|---|
| Backoff Exponent | $3 - 10$ |
| Backoff Unit Time | 0.165 millisecond |
| Max Number of Backoffs | 10 |
| Contention Window Length | 2 |

parameters for our evaluations. These results are presented in Section 5.4.3 in comparison with the QNC scenario.

### 5.4.2 Implementation of QNC

Similar to QandPF scenario, we do not consider the delays associated with the initial set up and maintenance of the PAN in our evaluations. In the initial configuration step, the decoder is assumed to receive enough information to know the complete structure of the network. This will allow the decoder to generate the network coding coefficients by using the same seed used for generation of these pseudo-random coefficients at the other nodes.

As described in Section 5.3, each node needs to transmit its QNC packets at each step of QNC. This requires careful time and frequency allocation for the nodes, as described in Section 5.3. Once the decoder node knows the complete deployment structure (*i.e.* connectivity of the nodes), it can allocate the time and frequency according to the algorithm described in Section 5.3. To determine the interfering nodes, $P_{\text{interference}}$ is set to be 5 dB less than the average power of received signals from connected nodes. The frequency and time allocation information is then sent back to the nodes. The beacon-to-beacon time difference is determined based on the packet length $L$ used for the QNC scenario. It should also be noted that since the MAC and PHY headers are around 19 bytes length, the beacon-to-beacon delay has no linear relation with the packet length.

In the packet forwarding scenario a packet is re-transmitted if its acknowledgement is not received. In QNC scenario, we do not use any ACK packets and if a packet is lost during the transmission, it is not re-transmitted. As it was discussed in Section 5.3, the decoder has the ability to decode the messages even when a few packets are missed during the transmission. In our simulations, the value of $\lambda_{\text{loss}}$ (the maximum number of packet losses) is set according to the average packet loss rate obtained from our initial simulations (shown in Fig. 5.7). Specifically, for a network with $|\mathcal{E}|$ edges at QNC step $t$ and average packet drop rate $\bar{p}_{\text{loss}}$ (which depends on the noise power $\sigma^2_{\text{noise}}$), we choose: $\lambda_{\text{loss}} = \bar{p}_{\text{loss}}(t-1)|\mathcal{E}|$. However, our simulation results show that choosing such value for $\lambda_{\text{loss}}$ leads the $\ell_1$-min decoder to be able to find a solution.

### 5.4.3 Analyzing the Results

To analyze the simulation results, we use different parameters resulting from QandPF and QNC scenarios. We use decoding SNR as the main measure of performance in the simulations, as defined in Section 3.4. We also measure the average total transmission energy in both scenarios. For a modulated signal generated by the PHY layer $s(t_{\text{PHY}})$

which is time-limited between $t_{\mathrm{PHY},1}$ and $t_{\mathrm{PHY},2}$, its energy is defined as:

$$E_{\mathrm{PHY}} = \int_{t_{\mathrm{PHY},1}}^{t_{\mathrm{PHY},2}} s^2(t_{\mathrm{PHY}}) \cdot dt_{\mathrm{PHY}}. \tag{5.17}$$

It determines how much energy the transmitter node has to consume to propagate the signal in the physical medium. We calculate the sum of all energies in the network for each network deployment in both QNC and QandPF scenarios. The average of these energies over different network deployments is reported in the following and it used for our evaluations.

We also calculate the total number of packet drops in the network in both cases of QNC and QandPF scenarios. The packet drop is usually caused by the physical layer noise, modeled for the channel. However, in QandPF scenario, it may also be due to the collisions resulting from the competition of nodes to access the channel. The ratio of the dropped packets to the transmitted packets is called *packet drop rate* in the following. In Fig. 5.7, the packet drop rate is depicted versus the power of the noise $\sigma^{dBm}_{\mathrm{noise}} = 10\log_{10}(\sigma^2_{\mathrm{noise}})$, described in Section 5.1. For the results in this figure, the packet drops are only caused by the channel noise (no interference or collision was involved in the calculations for this figure). It was shown by our simulations that having $\sigma^{dBm}_{\mathrm{noise}} = -90$ dBm results in no packet drops. As shown in Fig. 5.7, the packet drop rate changes from 0 (no packet drop) at noise power of $-90$ dBm to 0.5 (many packet drops) at noise power of $-75$ dBm. We run our simulations for different values of noise power as presented in Table 5.3.

In Figs. 5.8-5.13, we present our simulation results. For each deployment scenario and noise power, we show the following curves for QNC and QandPF scenarios:

- Average total transmission energy versus the elapsed time from the beginning of transmission,
- Average packet drop rate over different packet lengths versus the elapsed time,
- Average decoding SNR (as described in Section 3.4) versus the elapsed time,
- Average decoding SNR versus the sparsity factor $k/n$,
- Average decoding SNR versus the average total transmission energy required to achieve the corresponding decoding SNR.

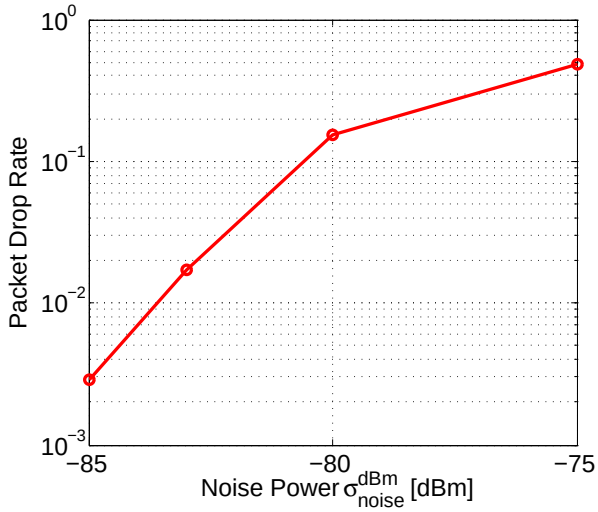



**Figure 5.7** Frequency of packet drops versus the power of the noise in dBm.

In quantized network coding, we transmit over all of the outgoing links. As a result, the number of (inter-node) transmissions in QNC scenario is much more than that in QandPF scenario. Since the total transmission energy is almost linearly related to the number of (inter-node) transmissions, the consumed total energy in QNC scenario is more than that in QandPF scenario. This fact is illustrated in the Figs. 5.8-5.13(a).

The packet drop rates shown in Figs. 5.8-5.13(b) are calculated for each case of QNC and QandPF scenarios. In contrast with the packet drop rate, illustrated in Fig. 5.7, the packet drops in QandPF scenario may also be due to the collisions happened during the CSMA/CA mechanism. As it is discussed in Section 5.3, the packet drops in QNC scenario happen only because of the physical channel error and the interference from other nodes.[8]. Hence, as it is shown in Figs. 5.8-5.13(b) that the resulting packet drop rate in QandPF scenario is more than the packet drop rate in QNC scenario.

Our resulting curves in Figs. 5.8-5.13(c) show the average decoding SNR versus the delay in milliseconds. The resulting decoding SNR for QNC scenario is better than that for QandPF scenario, in almost all of the cases. Similar to the results in Section 3.4, the smaller values of sparsity factor (meaning higher level of correlation between messages) results in a higher decoding SNR in QNC scenario. However, in QandPF scenario where

[8]Our design for the allocation of frequency and time slots prevents interference with respect to a threshold value $P_{\text{interference}}$, defined in Section 5.3

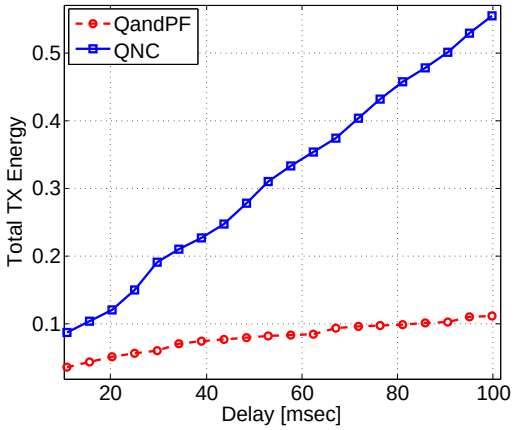
the correlation of messages is not considered during the data gathering, the changes in the sparsity factor does not result in any changes on the resulting decoding SNR. This is also illustrated in Figs. 5.8-5.13(d). The results in Figs. 5.8-5.13(c) and Figs. 5.8-5.13(d) show the advantage of using QNC over QandPF in terms of the delay, required to achieve a certain decoding SNR (level of quality).

The curves in Figs. 5.8-5.13(e) show the total transmission energy required to achieve a value of decoding SNR. It is clear that QandPF outperforms QNC in terms of the total transmission energy required to achieve a value of decoding SNR. However, for some low values of decoding SNR, the QNC scheme outperforms QandPF scheme. Such cases correspond to low delay values for which correlation of messages allow the $\ell_1$-min decoder to obtain an approximate of messages from very few number of received packets.

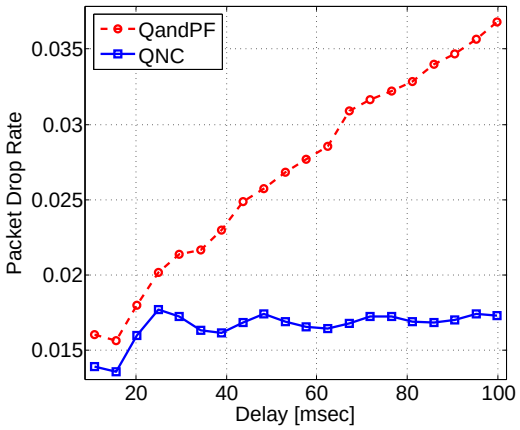
In Fig. 5.14, we depict the resulting decoding SNR for different noise powers in dBm. As it is expected, the resulting decoding SNR decreases as the noise power increases.

## 5.5 Summary and Conclusion

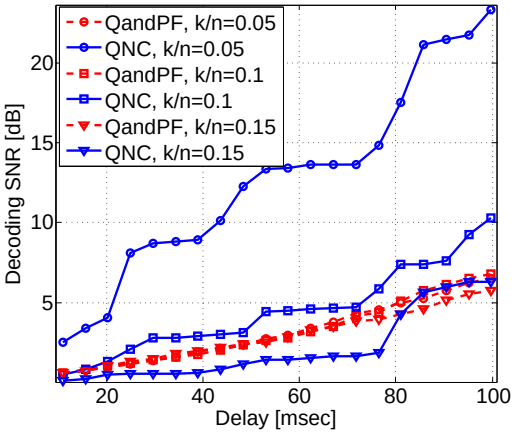
In this chapter, we studied the feasibility of using quantized network coding in practice. This was done by adapting it into the IEEE 802.15.4 standard which characterizes the wireless communication in sensor networks. Further, we have proposed a time and frequency allocation scheme to efficiently deal with the delay and interference during the steps of network coding. Finally, we used computer simulation to evaluate the overall performance of quantized network coding and packet forwarding by using a realistic model for the physical channel. Our simulation results show that quantized network coding results in higher decoding SNR, compared to the conventional packet forwarding. As the assumptions in our computer simulations are very close to the practical cases, we can claim a significant improvement of decoding SNR over quantization and packet forwarding, even for implementation in practice.



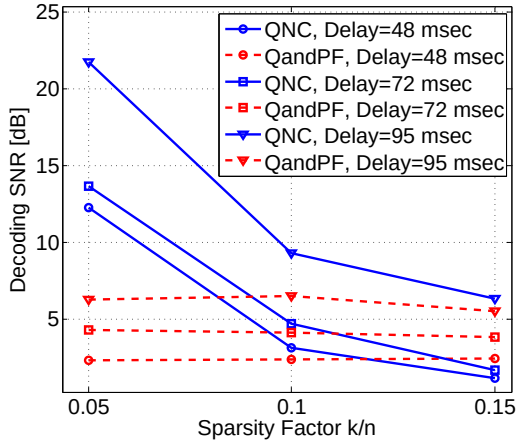

(a) Total TX Energy versus Delay



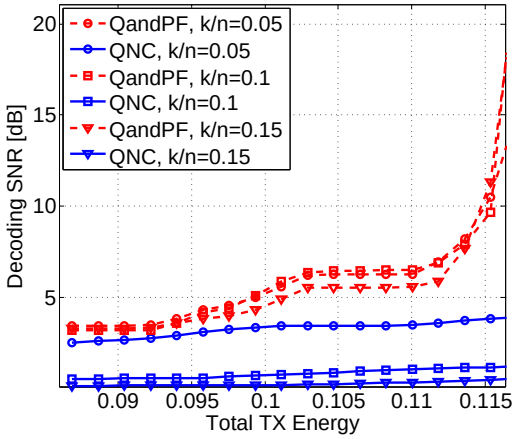

(b) Packet Drops versus Delay




(c) Decoding SNR versus Delay

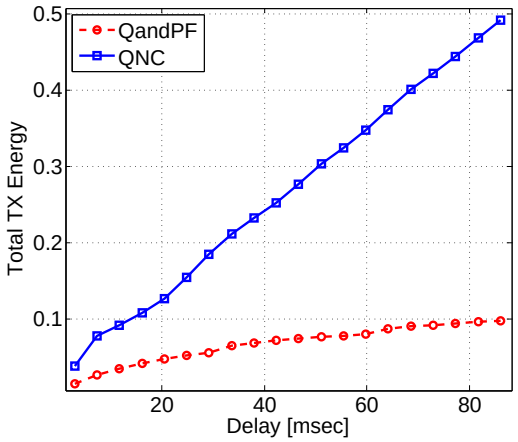



(d) Decoding SNR versus Sparsity Factor

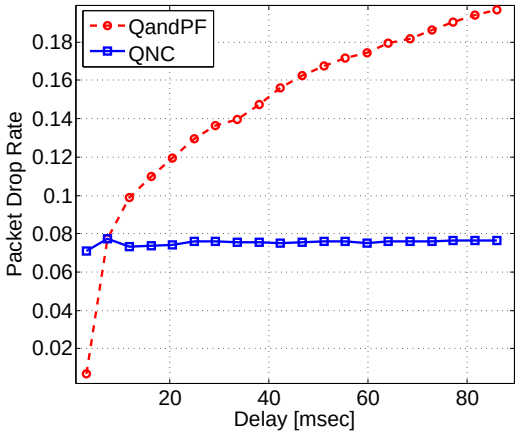



(e) Decoding SNR versus Total TX Energy

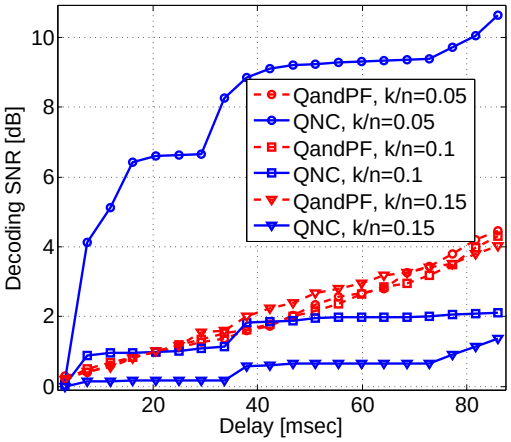
**Figure 5.8** Comparison of QNC and QandPF scenarios using IEEE 802.15.4 standard for the noise level of $-85$ dBm and $R_{\text{dim}} = 400$.



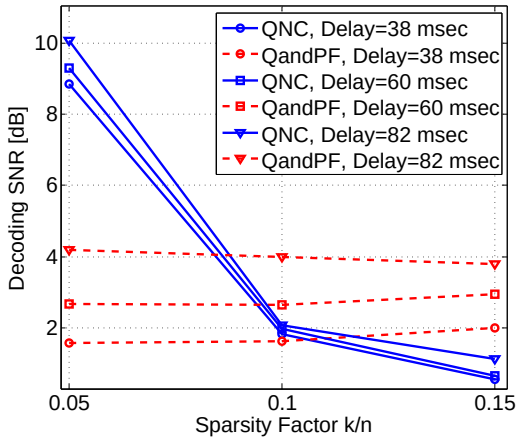

(a) Total TX Energy versus Delay



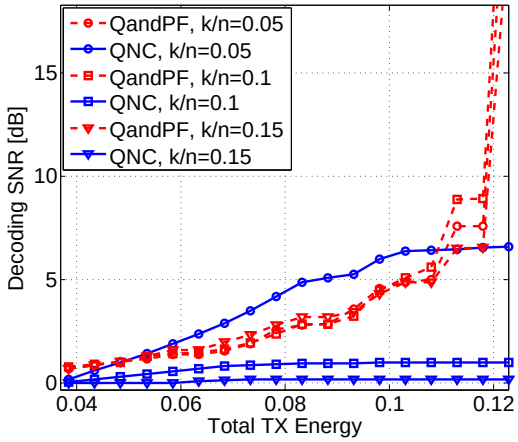

(b) Packet Drops versus Delay




(c) Decoding SNR versus Delay

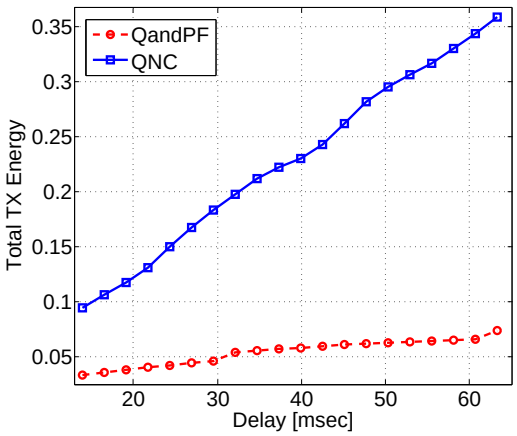



(d) Decoding SNR versus Sparsity Factor

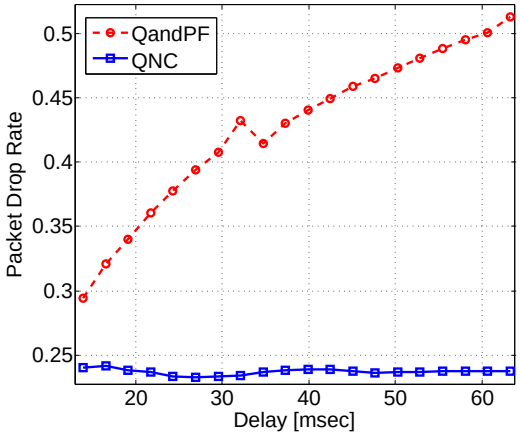



(e) Decoding SNR versus Total TX Energy

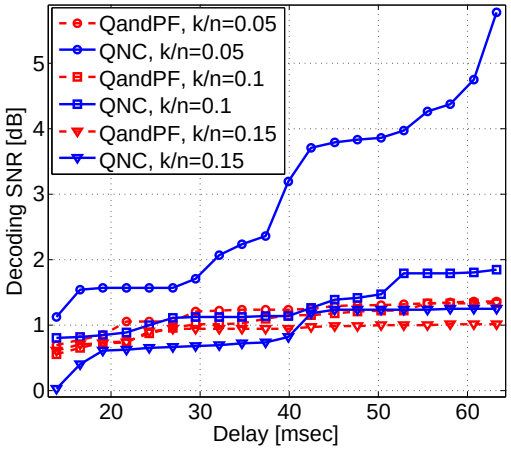
**Figure 5.9** Comparison of QNC and QandPF scenarios using IEEE 802.15.4 standard for the noise level of $-82$ dBm and $R_{\text{dim}} = 400$.



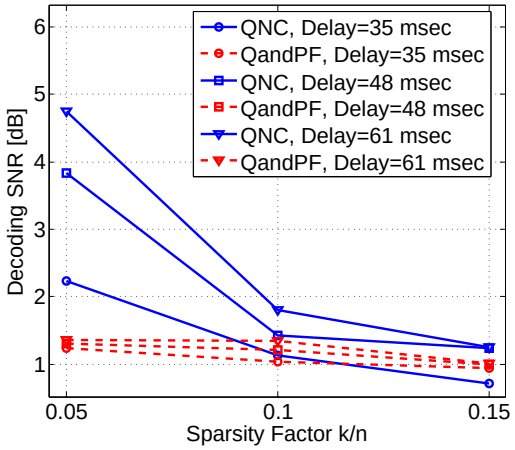

(a) Total TX Energy versus Delay



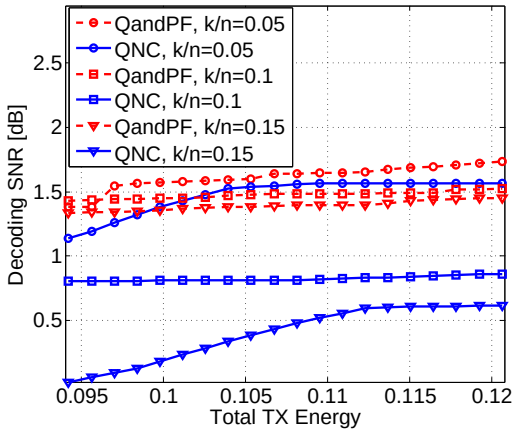

(b) Packet Drops versus Delay




(c) Decoding SNR versus Delay

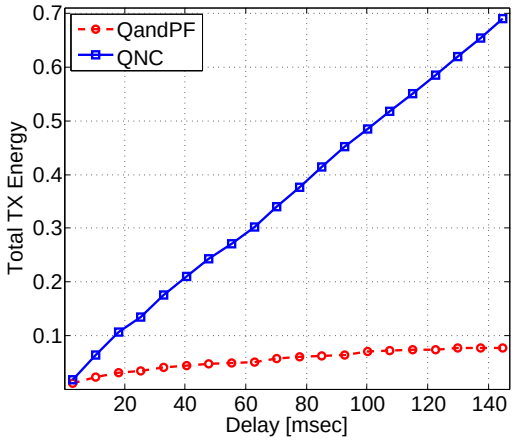



(d) Decoding SNR versus Sparsity Factor

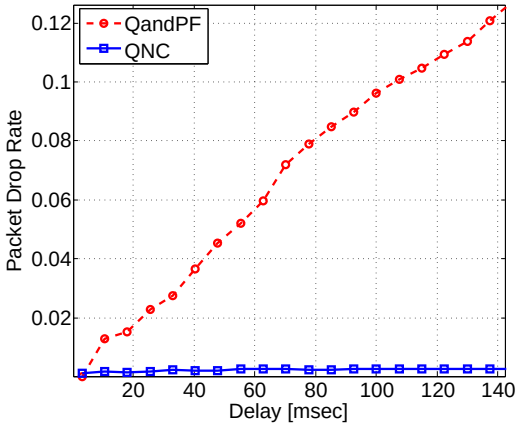



(e) Decoding SNR versus Total TX Energy

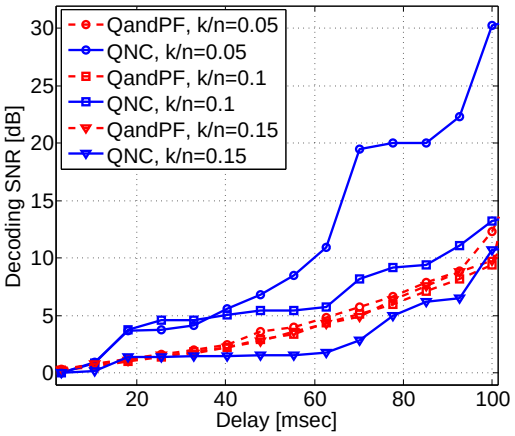
**Figure 5.10** Comparison of QNC and QandPF scenarios using IEEE 802.15.4 standard for the noise level of $-79$ dBm and $R_{\text{dim}} = 400$.



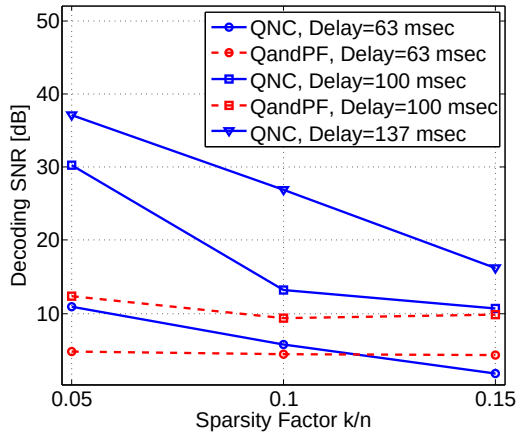

(a) Total TX Energy versus Delay



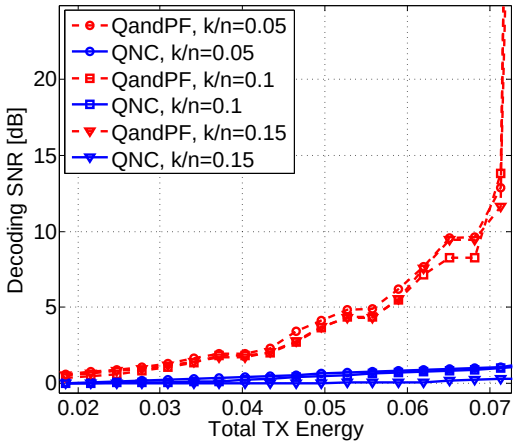

(b) Packet Drops versus Delay




(c) Decoding SNR versus Delay

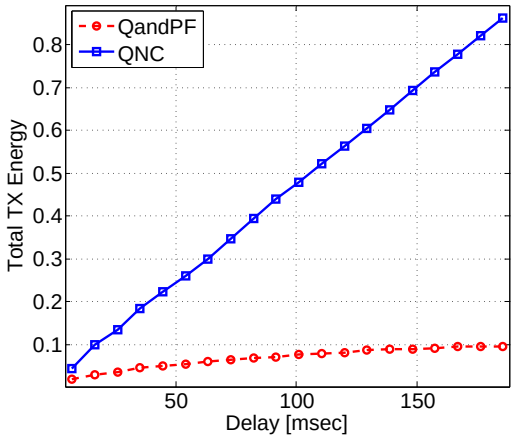



(d) Decoding SNR versus Sparsity Factor

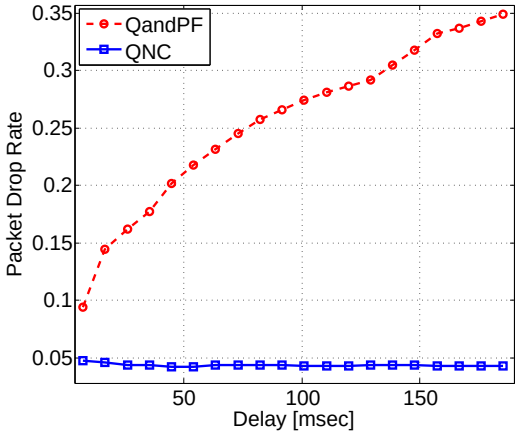



(e) Decoding SNR versus Total TX Energy

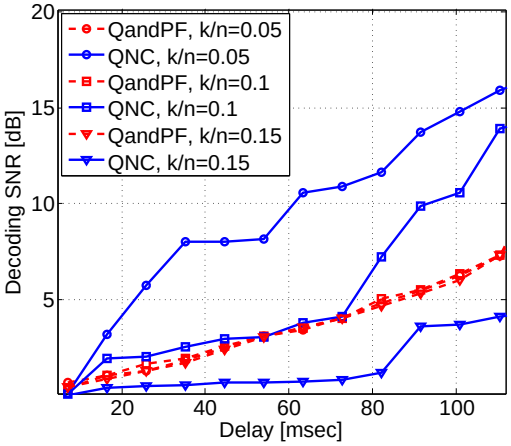
**Figure 5.11** Comparison of QNC and QandPF scenarios using IEEE 802.15.4 standard for the noise level of $-85$ dBm and $R_{\text{dim}} = 250$.



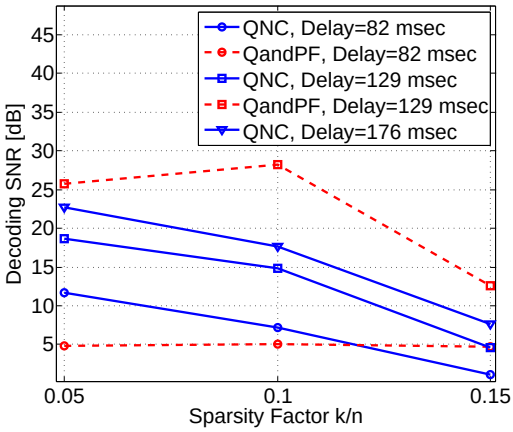

(a) Total TX Energy versus Delay



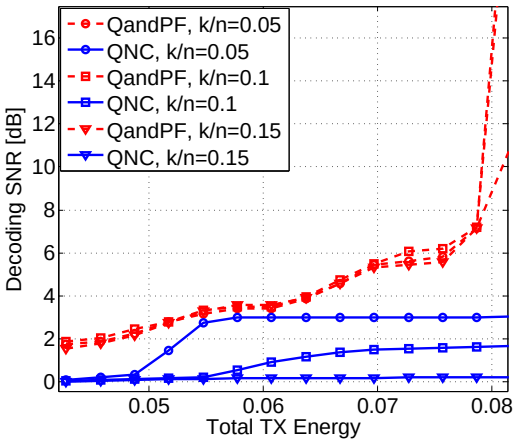

(b) Packet Drops versus Delay




(c) Decoding SNR versus Delay

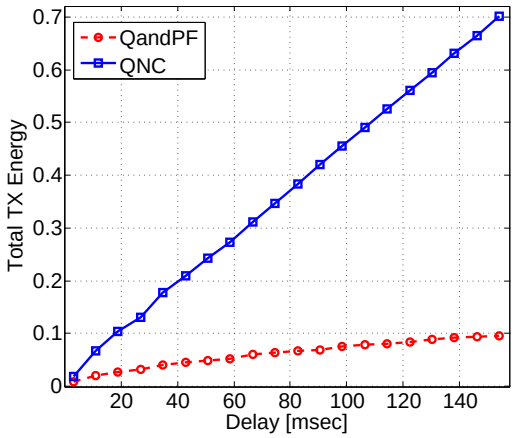



(d) Decoding SNR versus Sparsity Factor

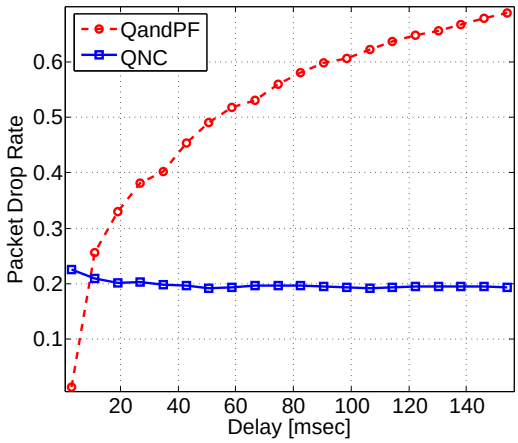



(e) Decoding SNR versus Total TX Energy

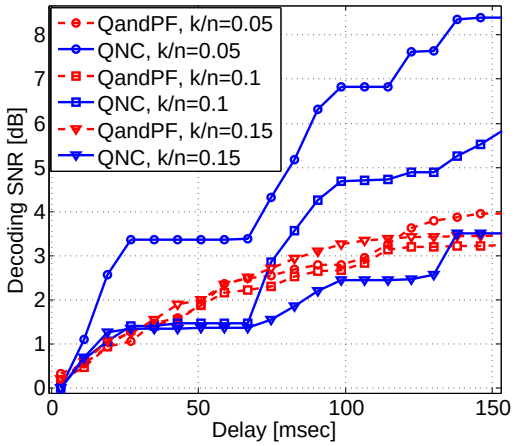
**Figure 5.12** Comparison of QNC and QandPF scenarios using IEEE 802.15.4 standard for the noise level of $-82$ dBm and $R_{\text{dim}} = 250$.



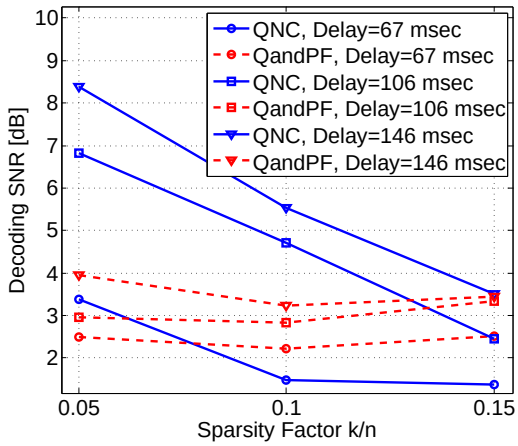

(a) Total TX Energy versus Delay



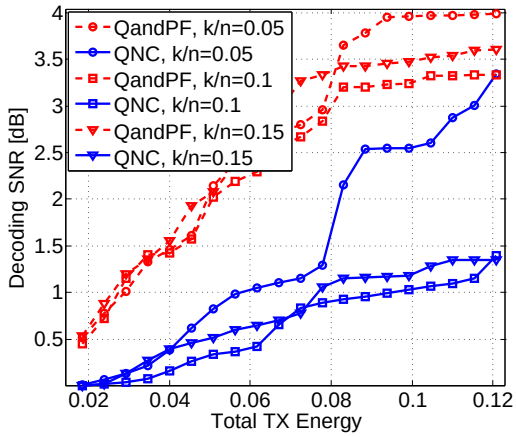

(b) Packet Drops versus Delay




(c) Decoding SNR versus Delay

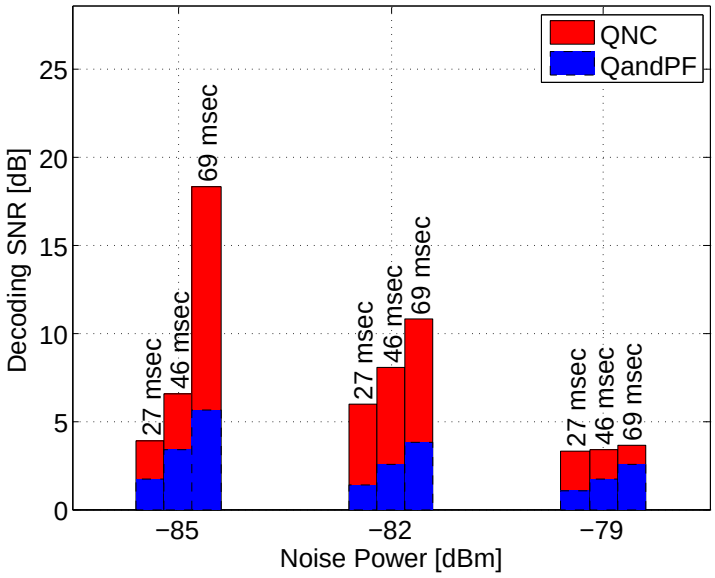



(d) Decoding SNR versus Sparsity Factor

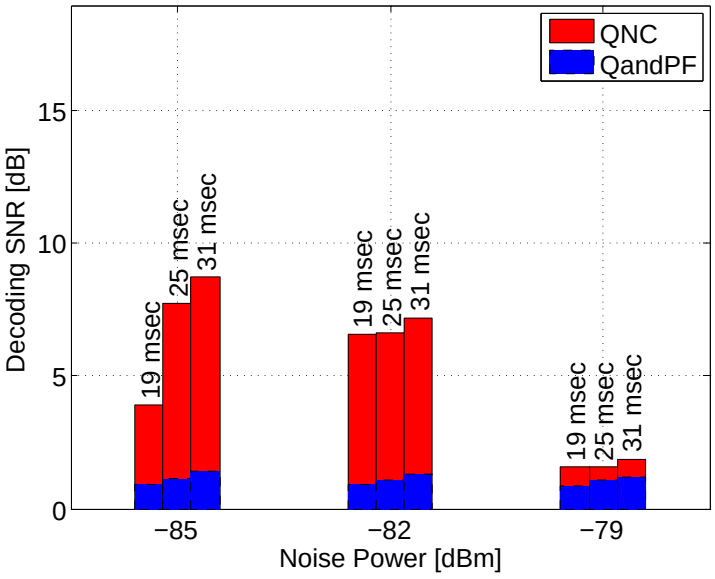



(e) Decoding SNR versus Total TX Energy

**Figure 5.13** Comparison of QNC and QandPF scenarios using IEEE 802.15.4 standard for the noise level of $-79$ dBm and $R_{\text{dim}} = 250$.




(a) $R_{\text{dim}} = 250$




(b) $R_{\text{dim}} = 400$

**Figure 5.14** Comparison of QNC and QandPF for different noise powers with $k/n = 0.05$: Each bar corresponds to the average decoding SNR obtained for a different delay in unit of millisecond.

# Chapter 6

# Conclusions and Future Works

In this thesis, we studied the possibility of using compressed sensing concepts during the network coding in sensor networks. We have proposed a new data gathering scheme and provided theoretical and numerical discussions around the decoding of messages. In this chapter, we provide a summary of this thesis and discuss some of the directions for the future works.

## 6.1 Thesis Summary

In Chapter 3, we proposed a new data gathering scheme called Quantized Network Coding (QNC) which incorporates the concepts of compressed sensing and sparse recovery within the network coding framework. We have presented a formal definition of QNC at each node and formulated end-to-end equations from the messages to the received packets at the gateway node. Using this linear mapping from the messages to the received packets, we studied the performance of quantized network coding by using the results available in the literature of compressed sensing. Specifically, we have used the restricted isometry property of the random matrices to propose an appropriate design for the network coding coefficients involved in quantized network coding. We also proposed to use an $\ell_1$-min decoding algorithm for decoding of messages from quantized network coded packets at the decoder node. Our design for the network coding coefficients helped us derive an upper bound on the decoding error for $\ell_1$-min decoding in Chapter 3. In Appendix A, we have presented a different bound on the decoding error which is based on a restricted eigenvalue condition. For the lossless networks, we have provided extensive simulation results to

show the capability of quantized network coding in performing *distributed compression* of correlated messages *without knowing their correlation* at the sensor nodes.

In Chapter 4, we introduced the Bayesian quantized network coding scenario in which the correlation of messages is known beyond their sparsity at the decoder node. For such scenario, we proposed an upper performance bound which helped us make conclusions on the asymptotic performance of quantized network coding. The obtained result showed that the required number of received packets to decode the messages is of an smaller order than the number of messages. Hence, quantized network coding is performing an embedded distributed compression during the transmission without the need to adapting the sensor nodes to the correlation between the nodes. Theoretically, the Bayesian scenario needs a minimum mean squared error decoder to perform appropriate decoding on the received packets at the decoder. However, the computational complexity of minimum mean squared error decoding does not make it a practical solution for Bayesian scenarios. We have addressed this issue by using a special form of belief propagation called Generalized Approximate Message Passing which is suitable for our quantized network coding scenario with dense measurement matrices. Our simulation results show the benefit of using such message passing based approximate decoding compared to $\ell_1$-min decoding in Bayesian quantized network coding.

In Chapter 5, we focused on implementation of quantized network coding for networks with noisy links. We have made changes to adapt quantized network coding within the IEEE 802.15.4 standard which characterizes low rate transmission for wireless sensor networks. Our adaptations include a different frequency and time allocation for the sensor nodes than those recommended in the standard. As an advantage of quantized network coding, we removed the acknowledgement procedure for the packet transmissions which result in lower delivery delay for data gathering. However, we had to modify the measurement noise bound to take into account the effect of the possible loss of packets due to the physical channel noise. We have simulated the adapted version of quantized network coding for the use in a practical wireless sensor network. Since we have made assumptions in our simulations that are very close to the reality in practice, we can claim a similar performance as those obtained in our simulations. Further, minor changes to the IEEE 802.15.4 standard makes our quantized network coding scheme a good option for implementation in practical scenarios.

## 6.2 Future Works

The research on merging the concepts of compressed sensing and network coding has recently drawn attention and it is still on the rise. As compressed sensing is a relatively new topic of research, there is still room to provide theoretical results for quantized network coding. Some other directions of research include more adaptation of quantized network coding in practical scenarios. Specifically, this is a necessity when the messages of the nodes are changing fast as discussed in the following.

### 6.2.1 Online Network Coding

In this thesis, we assumed that the messages are constant over time. This is because the traditional way of network coding works on each *generation* of the messages. In other words, network coding is done on a generation of messages until all of the messages of that batch are reconstructed at the destination nodes. The next generation of messages can be transmitted after the transmission of current generation is finished.

In applications where the generation of messages are changing quickly, the generation based quantized network coding may not meet the average end-to-end delay requirement. *Online* network coding was initially proposed in [81] to simultaneously incorporate information of different generations during the network coding. The proposed method in [81] uses the messages of generations which are still not decoded at the destination nodes to calculate and send linear combinations. In order to adapt the quantized network coding for practical cases, it is essential to transmit message of different batches in an online fashion.

### 6.2.2 Physical Layer Network Coding

We proposed to transmit quantized versions of real-valued linear combinations over the links to be able to adapt it to the conventional packet based transmission, similar to the IEEE 802.15.4 standard. This provides an isolation between different links in the network with no over-hearing of packets to other nodes. As in the relay channels where over-hearing of packets is useful, wireless sensor networks have that ability to over-hear packets transmitted to other nodes. Further, over-hearing may help in cancellation of interference signals, as discussed in Section 2.1.3.

As we have mentioned in Section 2.1.3, network coding over the physical layer signals

has already been studied for transmission in a single-hop proximity. As a future research direction, it is essential to study the physical layer network coding in a network where nodes are multiple hops away from each other. This will enable us to apply network coding in the real field without the need to perform quantization which causes distortion on the measurements. It was also discussed in Section 2.1.3 that physical layer network coding may not need the time sharing to prevent the interference. As a future work, it is interesting to investigate the feasibility of practical implementation of physical layer network coding for gathering of correlated messages.

### 6.2.3 Sparse Recovery in Finite Field

The theory of compressed sensing is well established in the real field. It is one of the other reasons that we had to perform network coding in the real field and then quantize them to match the limited capacity of links. Recently, the decoding of messages from linear combinations in the finite field was studied in [82]. Although the work in [82] has a limited extent, it shows the rising attention to the need for research on compressed sensing and sparse recovery in the finite field. In addition to the use of such results for our data gathering, it could be useful in many different fields of signal processing.

# Appendix A

# Restricted Eigenvalue Condition and its implication for QNC

Although different conditions have been used to study the sparse recovery from noise measurements, it is still not clear what conditions are both necessary and sufficient for minimum mean squared error decoding. As a sufficient but not necessary condition, we have introduced restricted isometry property in Section 2.3 and used it in Section 3.2.1 to design appropriate network coding coefficients. We also derived some probabilistic results for the $\ell_1$-min decoding of quantized network coded messages by using restricted isometry property. Unfortunately, the strictness characteristic of RIP makes it difficult to derive deterministic bounds for our QNC scenario since the total measurement matrices are structured.

A weaker type of conditions than RIP are called restricted eigenvalue conditions and have been used to obtain performance bounds for the decoding error of sparse recovery [63–65]. In [64], the authors show that restricted eigenvalue condition holds with high probability for a general class of Gaussian matrices with dependent entries, while RIP may be violated with high probability. A version of restricted eigenvalue condition which is developed by Bickel *et al.* [63], is shown to be less strict than the classical RIP and other version of restricted eigenvalue condition. Further, in [64], Raskutti *et al.* have introduced a condition equivalent to the restricted eigenvalue condition of Bickel *et al.* [63] and provide $\ell_2$ convergence rate for cases with very sparse messages [65].

## A.1 Implications of Restricted Eigenvalue Condition in QNC Scenario

Motivated by the restricted eigenvalue condition and the work in [65], we derive a performance bound for the sparse recovery of quantized network coded messages.

**Theorem A.1.1** *Consider the described QNC scenario for which the messages are exactly or at most k-sparse in transform domain $\phi$. In such case, using the $\ell_0$-norm estimator,*

$$\underline{\hat{X}}_{\ell_0}(t) = \phi \cdot \arg\min_{\underline{S}'} \|\underline{Z}_{\text{tot}}(t) - \Theta_{\text{tot}}(t)\underline{S}'\|_{\ell_2} \quad \text{s.t. } ||\underline{S}'||_{\ell_0} \leq k, \tag{A.1}$$

*the resulting estimation error is upper bounded as follows:*

$$\left\|\underline{X} - \underline{\hat{X}}_{\ell_0}(t)\right\|_{\ell_2}^2 \leq 8k \ \max_i\{\underline{N}_{\text{eff,tot}}(t)\}_i^2 \ \frac{\sum_{j=1}^n(\sum_{i=1}^m\{\Psi_{\text{tot}}(t)\}_{i,j}^2)}{\min_j(\sum_{i=1}^m\{\Theta_{\text{tot}}(t)\}_{i,j}^2)^2}\ , \tag{A.2}$$

*where $\Theta_{\text{tot}}(t) = \Psi_{\text{tot}}(t) \cdot \phi$.*

**Proof** Defining the error vector $\underline{\Delta} = \underline{\hat{S}}_{\ell_0}(t) - \underline{S}$, and $\underline{\Delta}' = \phi\underline{\Delta}$, with $\underline{\hat{S}}_{\ell_0}(t) = \phi^T \ \underline{\hat{X}}_{\ell_0}$, we have:

$$\begin{aligned}
\|\Theta_{\text{tot}}(t) \cdot \underline{\Delta}\|_{\ell_2}^2 &= \sum_{i=1}^m(\sum_{j=1}^n\{\Theta_{\text{tot}}(t)\}_{i,j} \ \{\underline{\Delta}\}_j)^2 && \text{(A.3)}\\
&\geq \sum_{j=1}^n\sum_{i=1}^m\{\Theta_{\text{tot}}(t)\}_{i,j}^2 \ \{\underline{\Delta}\}_j^2 && \text{(A.4)}\\
&\geq \min_j\sum_{i=1}^m\{\Theta_{\text{tot}}(t)\}_{i,j}^2\sum_{j=1}^n\{\underline{\Delta}\}_j^2 && \text{(A.5)}\\
&= \min_j\sum_{i=1}^m\{\Theta_{\text{tot}}(t)\}_{i,j}^2 \ \|\underline{\Delta}\|_{\ell_2}^2. && \text{(A.6)}
\end{aligned}$$

The criterion of the estimator (A.1) implies that:

$$\left\|\underline{Z}_{\text{tot}}(t) - \Theta_{\text{tot}}(t)\underline{\hat{S}}_{\ell_0}\right\|_{\ell_2}^2 \leq \|\underline{Z}_{\text{tot}}(t) - \Theta_{\text{tot}}(t)\underline{S}\|_{\ell_2}^2. \tag{A.7}$$

By replacing the norm operator with inner product, we have:

$$\Big(\underline{Z}_{\text{tot}}(t)-\Theta_{\text{tot}}(t)\hat{\underline{S}}_{\ell_0}\Big)^T\Big(\underline{Z}_{\text{tot}}(t)-\Theta_{\text{tot}}(t)\hat{\underline{S}}_{\ell_0}\Big)\leq\Big(\underline{Z}_{\text{tot}}(t)-\Theta_{\text{tot}}(t)\underline{S}\Big)^T\Big(\underline{Z}_{\text{tot}}(t)-\Theta_{\text{tot}}(t)\underline{S}\Big).$$

Now, using the measurement equation (3.14) and simple algebra, we can obtain:

$$\|\Theta_{\text{tot}}(t)\cdot\underline{\Delta}\|_{\ell_2}^2\leq 2\underline{N}_{\text{eff,tot}}(t)^T\cdot\Theta_{\text{tot}}(t)\ \underline{\Delta}, \tag{A.8}$$

implying:

$$\|\underline{\Delta}\|_{\ell_2}^2\leq 2\frac{\underline{N}_{\text{eff,tot}}(t)^T\cdot\Theta_{\text{tot}}(t)\ \underline{\Delta}}{\min_j\sum_{i=1}^m\{\Theta_{\text{tot}}(t)\}_{i,j}^2}\ . \tag{A.9}$$

And for the last term, we have:

$$\begin{aligned}
\underline{N}_{\text{eff,tot}}(t)^T\cdot\Theta_{\text{tot}}(t)\ \underline{\Delta} &= \sum_{j=1}^n(\sum_{i=1}^m\{\underline{N}_{\text{eff,tot}}(t)\}_i\{\Theta_{\text{tot}}(t)\}_{i,j})\{\underline{\Delta}\}_j && \text{(A.10)}\\
&\leq \sum_{j=1}^n\Big|\sum_{i=1}^m\{\underline{N}_{\text{eff,tot}}(t)\}_i\{\Theta_{\text{tot}}(t)\}_{i,j}\Big|\cdot\Big|\{\underline{\Delta}\}_j\Big| && \text{(A.11)}\\
&\leq \max_j\Big|\sum_{i=1}^m\{\underline{N}_{\text{eff,tot}}(t)\}_i\{\Theta_{\text{tot}}(t)\}_{i,j}\Big|\cdot\sum_{j=1}^n\Big|\{\underline{\Delta}\}_j\Big| && \text{(A.12)}\\
&= \Big|\Big|\underline{N}_{\text{eff,tot}}(t)^T\cdot\Theta_{\text{tot}}(t)\Big|\Big|_{\ell_\infty}||\underline{\Delta}||_{\ell_1}\ . && \text{(A.13)}
\end{aligned}$$

Representing the binary vector of support of $\underline{\Delta}$ by $\underline{1}_\Delta$, Cauchy-Schwarz inequality and the fact that $||\underline{\Delta}||_{\ell_0}\leq 2k$ imply:

$$\sum_{j=1}^n\{|\underline{\Delta}|\}_j\{\underline{1}_\Delta\}_j\leq\sqrt{\sum_{j=1}^n\{|\underline{\Delta}|\}_j^2}\sqrt{\sum_{j=1}^n\{\underline{1}_\Delta\}_j^2}, \tag{A.14}$$

Therefore:

$$\sum_{j=1}^n|\Delta_j|\leq\sqrt{2k}\sqrt{\sum_{j=1}^n\{\underline{\Delta}\}_j^2}, \tag{A.15}$$

implying:

$$||\underline{\Delta}||_{\ell_1}\leq\sqrt{2k}\|\underline{\Delta}\|_{\ell_2}. \tag{A.16}$$

Hence:

$$\|\underline{\Delta}\|_{\ell_2} \leq 2\sqrt{2k}\frac{\left|\left|\underline{N}_{\text{eff,tot}}(t)^T \cdot \Theta_{\text{tot}}(t)\right|\right|_{\ell_\infty}}{\min_j \sum_{i=1}^{m}\{\Theta_{\text{tot}}(t)\}_{i,j}^2} \ . \tag{A.17}$$

The numerator in the last equation can be simplified as:

$$\begin{aligned}
\left|\left|\underline{N}_{\text{eff,tot}}(t)^T \cdot \Theta_{\text{tot}}(t)\right|\right|_{\ell_\infty}^2 &= \left|\left|\underline{N}_{\text{eff,tot}}(t)^T \cdot \Psi_{\text{tot}}(t)\phi\right|\right|_{\ell_\infty}^2 \\
&\leq \left\|\underline{N}_{\text{eff,tot}}(t)^T \cdot \Psi_{\text{tot}}(t)\right\|_{\ell_2}^2 \qquad \text{(A.18)} \\
&= \sum_{j=1}^{n}(\sum_{i=1}^{m}\{\underline{N}_{\text{eff,tot}}(t)\}_i\{\Psi_{\text{tot}}(t)\}_{i,j})^2 \\
&\leq \sum_{j=1}^{n}\sum_{i=1}^{m}\{\underline{N}_{\text{eff,tot}}(t)\}_i^2\sum_{i=1}^{m}\{\Psi_{\text{tot}}(t)\}_{i,j}^2 \\
&\leq \sum_{i=1}^{m}\{\underline{N}_{\text{eff,tot}}(t)\}_i^2\sum_{j=1}^{n}\sum_{i=1}^{m}\{\Psi_{\text{tot}}(t)\}_{i,j}^2 \\
&= \left\|\underline{N}_{\text{eff,tot}}(t)\right\|_{\ell_2}^2 \cdot \|\Psi_{\text{tot}}(t)\|_{\ell_2}^2,
\end{aligned}$$

where the inequality (A.18) holds because $\|\phi\|_{\ell_2} = 1$. ■

It is essential to note that the estimator of (A.1) is a non-convex optimization problem. Therefore, the estimator of (A.1) does not offer a decoding algorithm for messages.

In this section, we assume that the network coding coefficients $\alpha_{e,v}(t)$'s are such that:

$$|\alpha_{e,v}(t)| = c_\alpha, \ \forall v \in \mathcal{V}, \ \forall t, \ \forall e \in Out(v). \tag{A.19}$$

For instance, they can be picked randomly from the set $\{-c_\alpha, +c_\alpha\}$. Such choice for the value of $\alpha_{e,v}(t)$'s results in the following theorem.

**Theorem A.1.2** *Consider the QNC scenario with the $\ell_0$ estimator of (A.1) and similar uniform quantizers with the step size of $\Delta_{\text{Q},0}$ over the interval $[-q_{\max}, +q_{\max}]$. If the network coding coefficients $\alpha_{e,v}(t)$'s $\beta_{e,e'}(t)$'s are picked such that* (A.19) *and*

$$\sum_{e' \in In(v)} |\beta_{e,e'}(t)| + |\alpha_{e,v}(t)| \leq 1, \ \forall t, \ \forall v \in \mathcal{V}, \ \forall e \in Out(v), \tag{A.20}$$

*hold then we have:*

$$\left\|\underline{X}-\underline{\hat{X}}_{\ell_0}(t)\right\|_{\ell_2}^2 \leq 2k\ \frac{\Delta_{\mathrm{Q},0}^2}{c_\alpha^2}\ R_{\mathrm{network}}^2(t), \tag{A.21}$$

*where:*

$$R_{\mathrm{network}}(t)=\frac{\sum_{t'=2}^{t}\sum_{e\in In(v_0)}(\sum_{t''=2}^{t'}\sum_{e'\in\mathcal{E}}|\{F_{prod}(t''+1;t')\}_{e,e'}|)^2}{\min_{v\in\mathcal{V}}\sum_{j=1}^{n}\sum_{t'=2}^{t}\sum_{e\in In(v_0)}\sum_{t''=2}^{t'}\sum_{e'\in Out(j)}\{F_{prod}(t''+1;t')\}_{e,e'}^2\phi_{j,v}^2}. \tag{A.22}$$

**Proof** Using (3.16) and the definition (3.17), we have:

$$\begin{aligned}
\left\|\underline{N}_{\mathrm{eff,tot}}(t)\right\|_{\ell_2}^2 &= \sum_{t'=2}^{t}\sum_{i'=1}^{|In(v_0)|}\{\underline{N}_{\mathrm{eff}}(t)\}_{i'}^2 && \text{(A.23)}\\
&= \sum_{t'=2}^{t}\sum_{e\in In(v_0)}\{\sum_{t''=2}^{t'}F_{\mathrm{prod}}(t''+1;t')\underline{N}(t'')\}_e^2 && \text{(A.24)}\\
&= \sum_{t'=2}^{t}\sum_{e\in In(v_0)}\Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}\{F_{\mathrm{prod}}(t''+1;t')\}_{e,e'}N_{e'}(t'')\Big)^2 && \text{(A.25)}\\
&\leq \sum_{t'=2}^{t}\sum_{e\in In(v_0)}\Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}|\{F_{\mathrm{prod}}(t''+1;t')\}_{e,e'}|\ |N_{e'}(t'')|\Big)^2 && \text{(A.26)}\\
&\leq (\frac{\Delta_{\mathrm{Q},0}}{2})^2\sum_{t'=2}^{t}\sum_{e\in In(v_0)}\Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}|\{F_{\mathrm{prod}}(t''+1;t')\}_{e,e'}|\Big)^2, && \text{(A.27)}
\end{aligned}$$

where the last inequality holds because (A.20) prevents overflow and hence $|N_e(t)| < \Delta_{\mathrm{Q},0}/2$. Similarly, for the total measurement matrix, (3.15) implies:

$$\begin{aligned}
\sum_{i=1}^{m}\{\Psi_{\mathrm{tot}}(t)\}_{i,j}^2 &= \sum_{t'=2}^{t}\sum_{i'=1}^{|In(v_0)|}\{\Psi(t)\}_{i',j}^2 && \text{(A.28)}\\
&= \sum_{t'=2}^{t}\sum_{e\in In(v_0)}\{\sum_{t''=2}^{t'}F_{\mathrm{prod}}(t''+1;t')A(t'')\}_{e,j}^2 && \text{(A.29)}\\
&= \sum_{t'=2}^{t}\sum_{e\in In(v_0)}\Big(\sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}\{F_{\mathrm{prod}}(t''+1;t')\}_{e,e'}\{A(t'')\}_{e',j}\Big)^2 && \text{(A.30)}
\end{aligned}$$

$$= \sum_{t'=2}^{t} \sum_{e\in In(v_0)} \Big( \sum_{t''=2}^{t'} \sum_{e'\in Out(j)} \{F_{\text{prod}}(t''+1;t')\}_{e,e'} \{A(t'')\}_{e',j} \Big)^2 . \quad \text{(A.31)}$$

Further, the assumption in (A.19) implies:

$$\begin{aligned}
\sum_{j=1}^{n}\sum_{t=1}^{m}\{\Psi_{\text{tot}}(t)\}_{i,j}^2 &\leq \sum_{j=1}^{n}\sum_{t'=2}^{t}\sum_{e\in In(v_0)} (\sum_{t''=2}^{t'}\sum_{e'\in Out(j)} |\{F_{\text{prod}}(t''+1;t')\}_{e,e'}|\ |\{A(t'')\}_{e',j}|)^2 \\
&= c_\alpha^2 \sum_{j=1}^{n}\sum_{t'=2}^{t}\sum_{e\in In(v_0)} (\sum_{t''=2}^{t'}\sum_{e'\in Out(j)} |\{F_{\text{prod}}(t''+1;t')\}_{e,e'}|)^2 \\
&\leq c_\alpha^2 \sum_{t'=2}^{t}\sum_{e\in In(v_0)} (\sum_{t''=2}^{t'}\sum_{e'\in\mathcal{E}} |\{F_{\text{prod}}(t''+1;t')\}_{e,e'}|)^2. \qquad \text{(A.32)}
\end{aligned}$$

Similarly, we have:

$$\begin{aligned}
\min_v \sum_{i=1}^{m}\{\Theta_{\text{tot}}(t)\}_{i,v}^2 &= \min_v \sum_{t'=2}^{t}\sum_{e\in In(v_0)} \{\sum_{t''=2}^{t'} F_{\text{prod}}(t''+1;t')A(t'')\phi\}_{e,v}^2 \\
&= \min_v \sum_{t'=2}^{t}\sum_{e\in In(v_0)} \Big( \sum_{t''=2}^{t'}\sum_{e'=1}^{|\mathcal{E}|}\sum_{j=1}^{n} \{F_{\text{prod}}(t''+1;t')\}_{e,e'} \{A(t'')\}_{e',j}\ \phi_{j,v} \Big)^2 \\
&= \min_v \sum_{t'=2}^{t}\sum_{e\in In(v_0)} \Big( \sum_{t''=2}^{t'}\sum_{j=1}^{n}\sum_{e'\in Out(j)} \{F_{\text{prod}}(t''+1;t')\}_{e,e'} \{A(t'')\}_{e',j}\ \phi_{j,v} \Big)^2 \\
&\geq \gamma_0^2\ \min_v \sum_{t'=2}^{t}\sum_{e\in In(v_0)} \sum_{t''=2}^{t'}\sum_{j=1}^{n}\sum_{e'\in Out(j)} \{F_{\text{prod}}(t''+1;t')\}_{e,e'}^2\ \phi_{j,v}^2 \\
&= \gamma_0^2\ \min_v \sum_{j=1}^{n}\sum_{t'=2}^{t}\sum_{e\in In(v_0)} \sum_{t''=2}^{t'}\sum_{e'\in Out(j)} \{F_{\text{prod}}(t''+1;t')\}_{e,e'}^2\ \phi_{j,v}^2, \qquad \text{(A.33)}
\end{aligned}$$

which concludes the proof. ■

The quantization step size $\Delta_{\text{Q},0}$ directly determines the level of measurement noises. Furthermore, $c_\alpha$ is the gain that the messages obtain before getting involved in the noisy measurements in QNC steps. As a result, the ratio $\Delta_{\text{Q},0}^2/c_\alpha^2$ in (A.21) can be considered as a noise to signal ratio. That the last term in (A.21), $R_{\text{network}}(t)$, only depends on the

structure of the network and the values of $\beta_{e,e'}(t)$'s.

## A.2 Proposed Bound versus Simulation Results

In this section, we compare the decoding error resulting of the simulations in Section 3.4 with the proposed bound in this section. The values of the $\ell_2$-norm of the $\ell_1$-min decoding error, resulting from our QNC simulations are obtained from the simulations carried out in Section 3.4.

To calculate the numerical values of the bound, we use the same settings for the network deployments as used in Section 3.4. The network coding coefficients $\alpha_{e,v}(t)$'s are selected according to (A.19) with $c_\alpha = 1$. The values of $\beta_{e,e'}(t)$'s are still selected in the same way as in Section 3.4. The resulting coefficients are then normalized to ensure (A.20) is satisfied. In the following, we discuss the trend of changes of the proposed upper bound with the changes of the parameters involved in QNC scenario. We will compare these trends with those of real simulation results, as discussed in the following.

### Sparsity Factor $k/n$

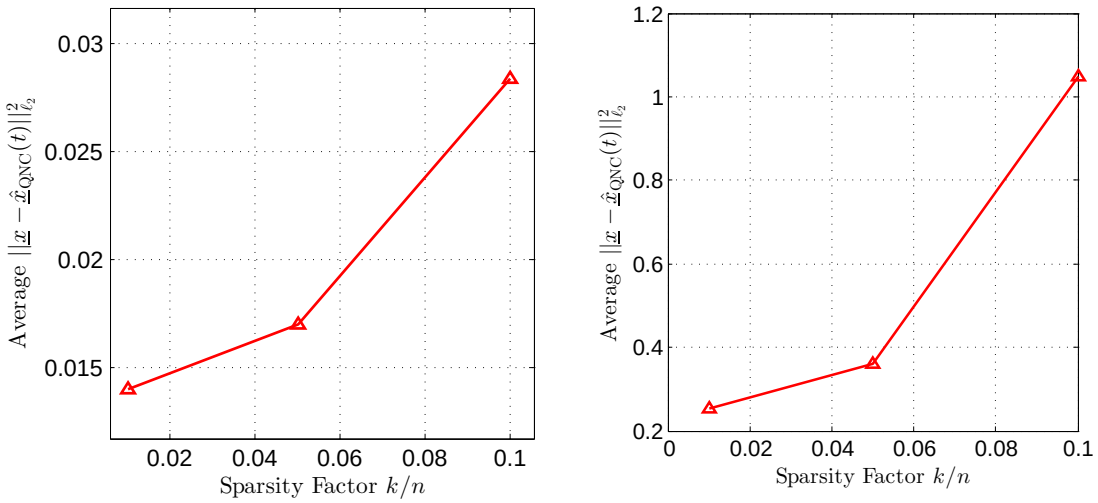
As the resulting bound in (A.21) suggests the estimation error may increase by increasing the value of $k$ (higher $k$ means less correlation between the messages). In Fig. A.1(a), the changes of real estimation error are shown when the sparsity factor is changed and other parameters of the QNC scenario are fixed. It is clear that the average norm of error increases by increasing the sparsity factor although it may not show a linear trend as suggested by our upper bound in (A.21).

### Packet Length $L$

Assuming uniform characteristic for the quantizers, used in QNC scenario, their step size is related to the adopted packet length as $\Delta_Q = 2q_{max}/2^{LC_0}$. Therefore, an increase in the packet length decreases the block length exponentially. Hence, in the logarithmic domain, we have:

$$\log_{10}(\Delta_Q^2) = 2\log_{10}(2q_{\max}) - 2\log_{10}(2)C_0L, \tag{A.34}$$

implying a linear relation between the logarithm of the proposed bound in (A.21) and the used packet length $L$.




(a) $t = 30, L = 25$ (b) $t = 35, L = 15$

**Figure A.1** Decoding error versus sparsity factor when $|\mathcal{E}| = 2400$.

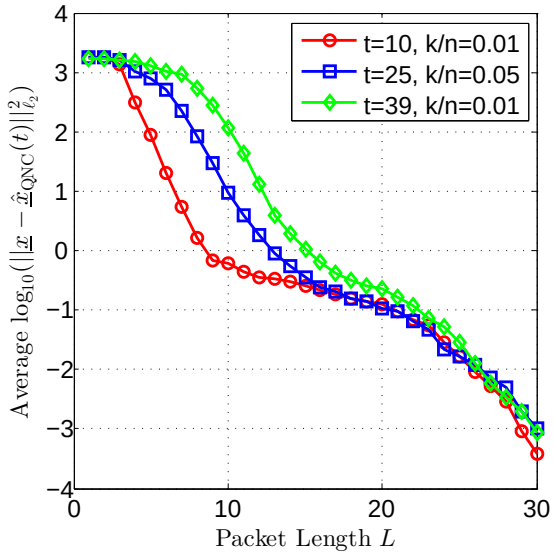



**Figure A.2** Decoding error versus packet length when $|\mathcal{E}| = 2400$.

In Fig. A.2, the logarithm of average recovery error norm is shown versus the changes of packet length, for a few different values of $t$ and $k/n$. It can be seen that the logarithmic real recovery error shows a *locally* linear characteristic in all of the cases. Moreover, the increase in the packet length reflects a decrease in the resulting recovery error in all the

cases.

### Time Index $t$

As in (A.21), our derived upper bound varies by the time index $t$ via the changes of $R_{\text{network}}(t)$, defined in (A.22). It can be inferred that both the numerator and denominator in (A.22) grow by increasing $t$ (*i.e.* by collecting more measurements at the decoder node). However, it is not easy to analyze its exact behavior theoretically and we need to use the numerical evaluations to understand its characteristics. In Fig. A.3, we have shown the values of $R_{\text{network}}(t)$ for different realizations of network deployments.

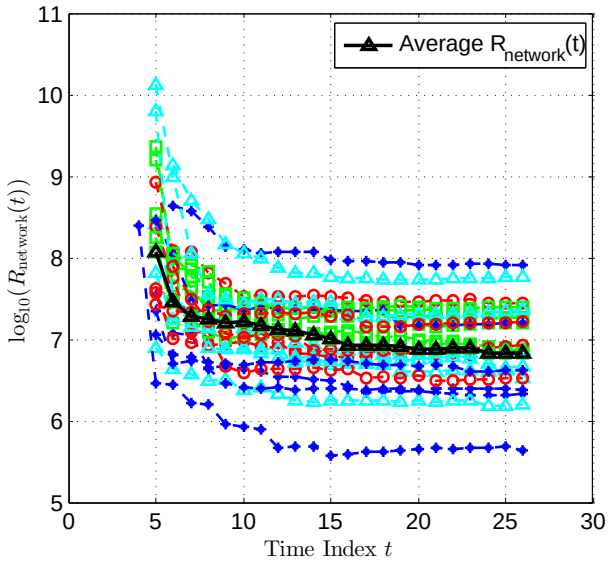



**Figure A.3** $\log_{10}(R_{\text{network}}(t))$ versus $t$ for different realization of network deployment with the same number of nodes and edges $|\mathcal{E}| = 2400$..

If the denominator of $R_{\text{network}}(t)$ in (A.22) is zero then the resulting bound will be infinity which is of no use. This happens when the received packets at the decoder does not have any information about one (or more) of the messages. Specifically, if the hopping distance from a node to the decoder node is more than the specific $t$, then the denominator would be zero. This is also reflected in the curves in Fig. A.3, as there is no value for some small values of $t$ in each case.

The other interesting conclusion from the curves in Fig. A.3 is that the resulting $R_{\text{network}}(t)$ generally shows a decreasing characteristic by increasing the time index $t$. This

is clearly as we would expect since collecting more received packets at the decoder should generally results in smaller decoding error. As it was shown in Fig. 3.4, this is true statement for our simulation results too. Unfortunately as shown in Fig. A.4, the proposed upper bound is very loose which makes it inapplicable for numerical evaluation of QNC.

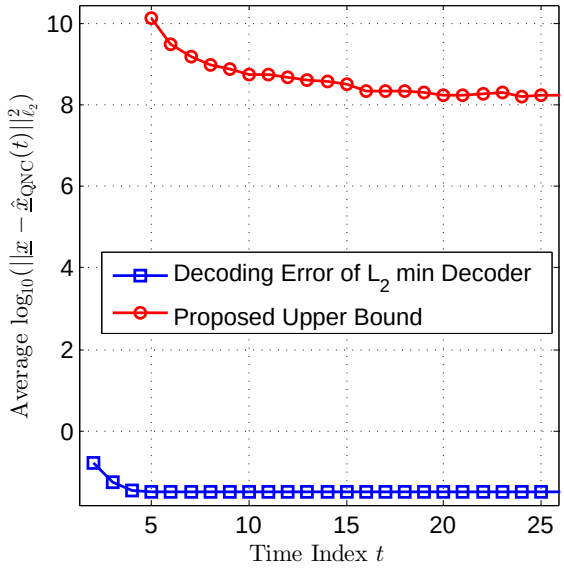



**Figure A.4** Decoding error versus time index for the proposed bound and simulation results obtained in Section 3.4 with $L = 14, |\mathcal{E}| = 1300, k/n = 0.1$..

# Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] V. C. Gungor and F. C. Lambert, "A survey on communication networks for electric system automation," *Computer Networks*, vol. 50, no. 7, pp. 877–897, 2006.

[3] N. Xu, "A survey of sensor network applications," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[4] C. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[5] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad hoc networks*, vol. 3, no. 3, pp. 325–349, 2005.

[6] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless communications, IEEE*, vol. 11, no. 6, pp. 6–28, 2004.

[7] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204 –1216, July 2000.

[8] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE International Symposium on Information Theory*, pp. 442–, june-4 july 2003.

[9] C. Fragouli, "Network coding for sensor networks," *Handbook on Array Processing and Sensor Networks*, pp. 645–667, 2009.

[10] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[11] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413 –4430, Oct. 2006.

[12] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[13] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, 2004.

[14] T. S. Han, "Slepian-Wolf-Cover theorem for networks of channels," *Information and Control*, vol. 47, no. 1, pp. 67 – 83, 1980.

[15] Y. Wu, V. Stankovic, Z. Xiong, and S. Kung, "On practical design for joint distributed source and network coding," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1709–1720, 2009.

[16] G. Maierbacher, J. Barros, and M. Médard, "Practical source-network decoding," in *6th International Symposium on Wireless Communication Systems*, pp. 283–287, IEEE, 2009.

[17] S. Cruz, G. Maierbacher, and J. Barros, "Joint source-network coding for large-scale sensor networks," in *IEEE International Symposium on Information Theory Proceedings*, pp. 420–424, IEEE, 2011.

[18] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289 –1306, April 2006.

[19] R. Baraniuk, M. A. Davenport, M. F. Duarte, and C. Hegde, *An Introduction to Compressive Sensing*. Connexions e-textbook, 2011.

[20] L. S. Committee *et al.*, "Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (lr-wpans)," *IEEE Computer Society*, 2003.

[21] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 174–185, ACM, 1999.

[22] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 14, no. 2, pp. 70–87, 2007.

[23] R. M. Karp, *Reducibility among combinatorial problems*. Springer, 1972.

[24] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 4, p. 24, 2008.

[25] Y. Zhu, K. Sundaresan, and R. Sivakumar, "Practical limits on achievable energy improvements and useable delay tolerance in correlation aware data gathering in wireless sensor networks.," in *SECON*, pp. 328–339, 2005.

[26] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 250–262, ACM, 2004.

[27] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[28] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[29] D. Ustebay, B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, "Greedy gossip with eavesdropping," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3765–3776, 2010.

[30] B. Dey, S. Katti, S. Jaggi, D. Katabi, M. Médard, and S. Shintre, "Real and complex network codes: Promises and challenges," in *Fourth Workshop on Network Coding, Theory and Applications*, pp. 1–6, 2008.

[31] S. Katti, S. Shintre, S. Jaggi, D. Katabi, and M. Médard, "Real network codes," in *Forty-Fifth Annual Allerton Conference*, pp. 389–395, 2007.

[32] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proceedings of the 12th annual international confernece on mobile computing and networking*, pp. 358 – 365, ACM, 2006.

[33] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 397–408, ACM, 2007.

[34] J. Barros and S. Servetto, "Network information flow with correlated sources," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 155 – 170, 2006.

[35] N. Harvey, R. Kleinberg, and A. Lehman, "On the capacity of information networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2345 – 2364, 2006.

[36] S. H. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung, "Multi-source noisy network coding," in *IEEE International Symposium on Information Theory Proceedings*, pp. 604 –608, 2010.

[37] S. Lim, Y. Kim, A. El Gamal, and S. Chung, "Noisy network coding," *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 3132–3152, 2011.

[38] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 789 –804, Mar. 2006.

[39] T. Berger, "Multiterminal source coding," *The information theory approach to communications*, vol. 229, pp. 171–231, 1977.

[40] T. Ho, M. Mdard, M. Effros, and R. Koetter, "Network coding for correlated sources," in *CISS*, 2004.

[41] A. Ramamoorthy, K. Jain, P. A. Chou, and M. Effros, "Separating distributed source coding from network coding," *IEEE/ACM Transactions on Networking*, vol. 14, no. SI, pp. 2785–2795, 2006.

[42] M. F. Duarte, S. Sarvotham, M. B. Wakin, D. Baron, and R. G. Baraniuk, "Joint sparsity models for distributed compressed sensing," in *Proceedings of the Workshop on Signal Processing with Adaptative Sparse Structured Representations*, 2005.

[43] K. Sayood, *Introduction to data compression*. Newnes, 2012.

[44] R. Baraniuk, M. Davenport, R. Devore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constr. Approx*, 2007.

[45] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, p. 969, 2007.

[46] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, pp. 4203 – 4215, December 2005.

[47] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[48] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathématique*, vol. 346, no. 9-10, pp. 589 – 592, 2008.

[49] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proceedings of the 5th international conference on Information processing in sensor networks*, pp. 134–142, ACM, 2006.

[50] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Processing Magazine*, vol. 25, pp. 92 –101, march 2008.

[51] C. Liu, C. Chigan, and C. Gao, "Compressive sensing based data collection in VANETs," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1756–1761, IEEE, 2013.

[52] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, MobiCom '09, (New York, NY, USA), pp. 145–156, ACM, 2009.

[53] F. Bassi, L. Chao, L. Iwaza, M. Kieffer, *et al.*, "Compressive linear network coding for efficient data collection in wireless sensor networks," in *Proceedings of the 2012 European Signal Processing Conference*, pp. 1–5, 2012.

[54] L. Iwaza, M. Kieffer, and K. Al-Agha, "MAP estimation of network-coded correlated sources," in *International Conference on Advanced Technologies for Communications (ATC)*, pp. 199–202, IEEE, 2012.

[55] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 269–280, 2010.

[56] S. Feizi, M. Médard, and M. Effros, "Compressive sensing over networks," in *48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1129–1136, IEEE, 2010.

[57] S. Feizi and M. Médard, "A power efficient sensing/communication scheme: Joint source-channel-network coding by using compressive sensing," in *49th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1048–1054, IEEE, 2011.

[58] C. Luo, J. Sun, and F. Wu, "Compressive network coding for approximate sensor data gathering," in *IEEE Global Telecommunications Conference*, pp. 1–6, IEEE, 2011.

[59] N. Nguyen, D. Jones, and S. Krishnamurthy, "Netcompress: Coupling network coding and compressed sensing for efficient data communication in wireless sensor networks," in *2010 IEEE Workshop on Signal Processing Systems*, pp. 356 –361, Oct. 2010.

[60] T. Kailath, *Linear systems*, vol. 1. Prentice-Hall Englewood Cliffs, NJ, 1980.

[61] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse problems*, vol. 23, no. 3, p. 969, 2007.

[62] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[63] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, "Simultaneous analysis of lasso and dantzig selector," *The Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.

[64] G. Raskutti, M. J. Wainwright, and B. Yu, "Restricted eigenvalue properties for correlated gaussian designs," *The Journal of Machine Learning Research*, vol. 99, pp. 2241–2259, 2010.

[65] G. Raskutti, M. J. Wainwright, and B. Yu, "Minimax rates of estimation for high-dimensional linear regression over $\ell_q$ -balls," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6976–6994, 2011.

[66] W. Dai, H. V. Pham, and O. Milenkovic, "Distortion-rate functions for quantized compressive sensing," in *IEEE Information Theory Workshop on Networking and Information Theory*, pp. 171–175, IEEE, 2009.

[67] A. Zymnis, S. Boyd, and E. Candes, "Compressed sensing with quantized measurements," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 149–152, 2010.

[68] L. Jacques, D. K. Hammond, and J. M. Fadili, "Dequantizing compressed sensing: When oversampling and non-Gaussian constraints combine," *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 559–571, 2011.

[69] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21." http://cvxr.com/cvx, Apr. 2011.

[70] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[71] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *Proceedings of the 6th international conference on Information processing in sensor networks*, pp. 331–339, ACM, 2007.

[72] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[73] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," *arXiv preprint arXiv:1010.5141*, 2010.

[74] M. Rosenblatt, "A central limit theorem and a strong mixing condition," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 42, no. 1, p. 43, 1956.

[75] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1561–1585, 2008.

[76] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," *CoRR*, vol. abs/1001.2228, 2010.

[77] A. Martinez-Sala, J.-M. Molina-Garcia-Pardo, E. Egea-Ldpez, J. Vales-Alonso, L. Juan-Llacer, and J. Garcia-Haro, "An accurate radio channel model for wireless sensor networks simulation," *Journal of Communications and Networks*, vol. 7, no. 4, pp. 401–407, 2005.

[78] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 517–526, 2004.

[79] K. S. Panchal, "Implementing physical layer (PHY) of IEEE 802.15.4g standard with direct sequence spread spectrum (DSSS) using offset quadrature phase shift keying (O-QPSK)," Master's thesis, San Diego State University.

[80] A. Boyaci, "A matlab implementation of greedy DSATUR coloring algorithm." `http://armanboyaci.com/?p=487`. Accessed: 2014-03-30.

[81] J. K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *IEEE International Symposium on Information Theory*, pp. 1651–1655, IEEE, 2008.

[82] T. Tošić, *Distributed detection, reconstruction and interpolation for sensor network data*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2013.