PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
SCHOOL OF ENGINEERING

MEDIUM ACCESS CONTROL FOR MIMO WIRELESS SENSOR NETWORKS

DIEGO KAULEN

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
CHRISTIAN OBERLI

Santiago de Chile, May 2015

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
SCHOOL OF ENGINEERING

MEDIUM ACCESS CONTROL FOR MIMO WIRELESS SENSOR NETWORKS

DIEGO KAULEN

Members of the Committee:
CHRISTIAN OBERLI
MARCELO GUARINI
DIEGO DUJOVNE
YADRAN ETEROVIC

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, May 2015

“What if I fall?”

“Oh but my darling,

what if you fly?”

ERIN HANSON

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, professor Christian Oberli, for believing in my abilities and always being available to answer my unceasing question and give me his advise on the several stages of this process.

Moreover, I also want to thank all the people I have worked with during the last year: Joaquín Aldunate, Santiago Barros, Jean Paul de Villers-Grandchamps, Carlos Feres, Marcelo Guarini, Felipe Kettlun, Fernando Rosas and Joaquín Venegas. I have learned a lot from each of them.

I would like to thank CONICYT Chile for supporting this research with the scholarship CONICYT-PCHA Magíster Nacional 2014 - 22141098 and the project CONICYT FONDEF IT13i20015 under which this thesis was developed.

Finally, I thank my family and friends, for being very supportive in the whole process. This final result would not be the same without the help they have given me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Sensor networks consist of autonomous wireless nodes that are networked together in an ad hoc fashion so as to monitor on or several variables that change over time and space. Transmissions between nodes are exposed to a varying MIMO (Multiple Input Multiple Output) channel. Channel state information (CSI) at both the transmitter and the receiver is needed to correctly transmit data between nodes. One way to obtain this information, is through the Reverse Channel Training (RCT) scheme, which quickly acquires the necessary information but does not have a functional MAC layer that supports the coexistence of more than two nodes. Existent MAC protocols are not compatible with this scheme, given that they need to send actual data in the first contact between nodes. Therefore, it is impossible to send operational data (such as node IDs, or addresses), jeopardizing the correct operation of the network addressing scheme.

In this thesis, a crossed-layer protocol is presented, where the addressing is achieved through multiple training signals (or *preambles*). These preambles must satisfy certain auto- and cross-correlation properties in order to provide correct medium access. Additionally, this protocol contemplates the development of a MAC layer compatible with the RCT scheme, allowing a correct initialization of the network.

It will be shown that ten preambles suffice for a correct operation of the crossed-layer protocol in low density networks, since it allows more than 90% of the nodes to initialize correctly. The effective operation of the protocol is tested at both a physical layer level and at a MAC layer level.

Keywords: Sensor networks, reverse-channel training, medium access control, dynamic address assignment, distributed algorithm, crossed layers.

RESUMEN

Las redes inalámbricas de sensores están formadas por nodos autónomos. Estos conforman una red dinámica, con el fin de monitorear alguna variable en el tiempo. Las transmisiones entre nodos están expuestas a un canal MIMO (*Multiple Input Multiple Output*) que cambia rápidamente. De él se debe estimar la información de su estado (CSI, *Channel State Information*) para lograr transmisiones que minimicen la probabilidad de error. Una forma de obtener esa información es a través del entrenamiento de canal reverso (RCT, *Reverse Channel Training*), esquema que adquiere la información necesaria rápidamente pero que no tiene una capa MAC funcional que justifique su implementación. Los protocolos MAC existentes en la literatura no son compatibles con este esquema ya que requieren enviar datos en la primera transmisión entre los nodos. Así, es imposible enviar datos operacionales (como por ejemplo, la dirección de los nodos), lo que compromete el direccionamiento de los nodos en la red.

En esta tesis, se presenta un protocolo de capa cruzada, en el que el direccionamiento de los nodos se logra a través de múltiples señales de entrenamiento (o preámbulos). Estos preámbulos deben satisfacer ciertas propiedades de autocorrelación y de correlación cruzada para que funcionen correctamente. Este protocolo también contempla el desarrollo de una capa MAC que es compatible con el esquema RCT, y que permite inicializar la red de manera eficaz.

Se mostrará que diez preámbulos son suficientes para que el protocolo funcione correctamente en redes de baja densidad –en donde el número de vecinos que cada nodo tiene es acotado–, ya que permiten que una red se inicialice con más del 90% de sus nodos. El funcionamiento de estos preámbulos es probado tanto en la capa física como en la capa MAC.

Palabras Clave: Redes de sensores, entrenamiento de canal reverso, control de acceso al medio, asignación de direcciones dinámico, algoritmo distribuido, capa cruzada.

1. INTRODUCTION

Wireless sensor networks (WSNs) have emerged as a solution to a number of data gathering applications, such as surveillance, environmental monitoring, hydrology, health care and wildlife monitoring (W. Ye, Heidemann, & Estrin, 2002). They consist of a number of small autonomous devices called sensor nodes. Each node is battery powered and equipped with integrated sensors for measuring its surrounding environment, data processing capabilities for synthesizing the sensed data, and short-range radio communications for sharing data with other nodes in the network (Schurgers, Kulkarni, & Srivastava, 2002).

Some applications are envisioned whereby very large geographic areas need to be monitored. To accomplish an effective WSN coverage of such large areas, networks with traditional single antenna nodes are impractical because they would require too many nodes in order to carry out the monitoring task with sufficient network connectivity. Beamforming techniques based on Multiple Input Multiple Output (MIMO) technologies can be used for significantly reducing the number of required sensor nodes without compromising network connectivity. In effect, beamforming yields a diversity gain that allows for sparser node placement. The highest diversity gain can be achieved using the strongest eigenmode of the MIMO channel matrix (Kettlun, 2014). The authors obtain full diversity gain by sending known training symbols –a *preamble*–, from the transmitter to the receiver and vice versa. This scheme is known as *channel dependent reverse-channel training* (RCT) (Bharath & Murthy, 2013), where receiver and transmitter estimate part of the MIMO channel matrix so that they can make use of the best eigenchannel. This procedure attains channel state information (CSI) at both the receiver and the transmitter, assuming a reciprocal channel (Venkataramani & Marzetta, 2003).

Under this scheme, any first transmission between two nodes may occur in a very low signal-to-noise (SNR) scenario. As a consequence, any transmission made by any node for discovering its neighbours will face two main challenges: (i) neighbours may not be able to detect the transmission because the beamforming gains between the transmitting

node and its neighbours are still unavailable, thus potentially leaving the transmissions below a minimum required SNR level; and (ii) neighbours that do detect the transmissions will not be able to identify operational information from the packet, such as the source or destination node IDs. Transmission detection is successfully overcome by means of the processing SNR gain inherent to the preamble transmission; however, operational information, such as destination node IDs, neighbour lists, and other useful data cannot be sent explicitly to the receiver node before achieving the beamforming SNR gain. This issue poses a major hurdle for regulating medium access control (MAC) when several nodes shall compose a network, which will be the main topic of this work.

This thesis was developed under the context of the project FONDEF D09I1094 and IT13I20015: Wireless sensor networks with multiple antenna technologies, which was carried out at the Wireless Technologies Laboratory (LATINA, by its Spanish acronym) of *Pontificia Universidad Católica de Chile* (UC). The project addresses the design and development of a digital wireless system for WSN applications using multiple antennas at both transmitter and receiver.

1.1 Motivation and Research Problem

Even though RCT is very energy efficient in point-to-point communications, it is practically useless if it does not have a MAC protocol underlying. Common MAC protocols cannot be used directly on top of this scheme, because they do not accept dispatching any data in the first contact. Existent MAC protocols send at least one address field in the first communication between nodes.

The lack of a compatible protocol motivates us to design one that fits the RCT scheme. The protocol must maintain the scheme's constraints: it must be energy efficient, adaptable and scalable.

The design will take advantage of the preambles used for detecting packets and for estimating some of the channel's physical parameters, such as the timing offset, the carrier

frequency offset (CFO), Rayleigh channel coefficients, amongst others. Different preambles will serve as physical addresses for nodes. This is a novel approach that, to the best of our knowledge, has not been proposed in the past.

1.2 Prior Art

A MAC protocol is used to address resolving potential contention and collision when the communication medium is used (Li, 2008). Many MAC protocols have been proposed for wireless networks, which often assume a common channel shared by mobile hosts. The following literature review summarizes the prior art on MAC protocols for wireless sensor networks. It consists of the review of two separate processes: MAC protocols and the initialization of the network.

1.2.1 On MAC Protocols

MAC protocols can be classified into two major categories: contention-free protocols and contention based protocols. Further classifications are shown in Figure 1.1.

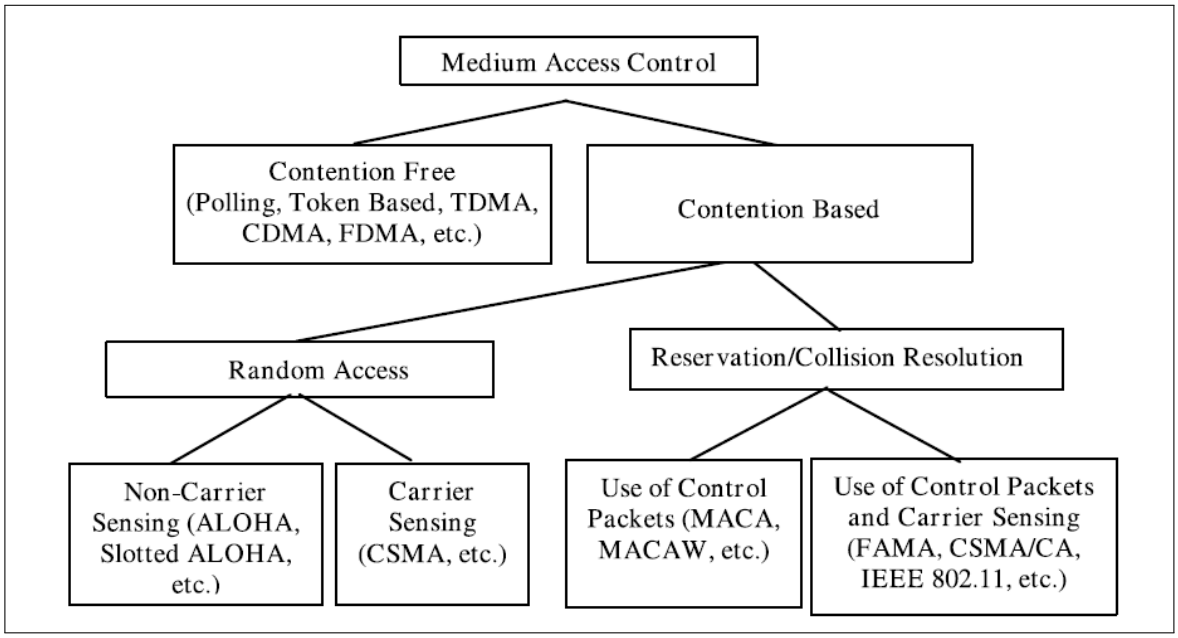


FIGURE 1.1. Classification of MAC protocols (Kumar et al., 2006).

Contention-free schemes work on different channels in order to avoid conflicts and collisions of packets. Although FDMA, CDMA and TDMA are plausible, the latter one is preferred in wireless sensor networks due to not requiring additional hardware. Diverse protocols have been proposed in the past (Bao & Garcia-Luna-Aceves, 2001; Chlamtac & Farago, 1994; Ephremides & Truong, 1990; Ju & Li, 1998; Rajendran, Obraczka, & Garcia-Luna-Aceves, 2003). It considers partitioning the medium in time-slots and scheduling so that nodes can transmit with a no-collision guarantee within their given time slots. This approach is generally energy efficient, because nodes may stay in idle state for most of the time and therefore avoid wasting energy. However, it requires precise time-synchronization between nodes, which tends to be hard to achieve in large networks. This kind of MAC scheme is more applicable to static networks and/or networks with centralized control.

Contention based protocols, on the other hand, are aware of the risk of collisions of transmitted data (Kumar et al., 2006). Among this category, we can further subdivide it into two major groups: random access protocols and collision resolution protocols. An example of the first group is ALOHA, where nodes access the channel as soon as it is ready. Naturally, many nodes may access the channel simultaneously, causing collisions. This is why this scheme is suitable under very low system loads only. An improvement of ALOHA, termed 'Slotted ALOHA' introduces synchronized time-slots similar to TDMA. In this scheme, nodes can only send messages at the beginning of these time slots, doubling the maximum achievable throughput, at the cost of necessary synchronization. CSMA-based protocols further reduce the number of collisions in any network. Some schemes are the Request-To-Send/Clear-To-Send (RTS/CTS) control packets to prevent collisions, such as Multiple Access with Collision Avoidance (MACA) (Karn, 1990), and MACA for Wireless LANs (MACAW) (Bharghavan, Demers, Shenker, & Zhang, 1994). Others combine carrier sensing and control packets, proposing severe energy savings (van Dam & Langendoen, 2003; W. Ye et al., 2002).

The development of MAC protocols for systems with directional antennas has been reviewed by Bazan and Jaseemuddin (2012). The authors classify the MAC protocols according to parameters, such as the omni-directionality of RTS/CTS, backoff mechanisms, the source of the beamforming information, amongst others.

This work presents a protocol solution for multiple antenna operation. The beamforming information is not obtained in any of the ways presented by Bazan and Jaseemuddin (2012). In this case, it is obtained on-the-fly as proposed by Bharath and Murthy (2013). The challenging part of the MAC protocol design is precisely the fact that the beamforming SNR gain is available only after there has been a preamble exchange between the communicating nodes. This MAC protocol will be explained in Chapter 4.

1.2.2 On the Initialization of the Network

The previously exposed protocols, assume that nodes are already initialized, which means that every node is assigned a unique address, knows who its neighbours are and how to refer to them. The procedure that nodes carry out in order to attain this state has also been studied in the literature.

Some authors divide the initialization problem into two steps: neighbour discovery and address assignment. McGlynn and Borbash (2001) take a time-slotted approach to carry out the neighbour discovery task. Nodes listen to transmissions only during a few time-slots, pursuing an energy efficient neighbour discovery after they have been deployed in an area. F. Ye and Pan (2009) survey addressing algorithms, classifying them into stateful and stateless, as shown in Figure 1.2. Stateful approaches assign unique addresses to new users, stateless approaches employ network wide Duplicate Address Detection (DAD) to ensure uniqueness of addresses. Amongst the stateful approaches, centralized and distributed schemes can be constructed. The first ones allow only one node in the network to assign addresses. In contrast, distributed schemes allow many nodes to assign addresses, decreasing overall overhead and complexity. An effective distributed stateful address assignment solution is presented in Schurgers et al. (2002), where authors propose a MAC address scheme. These addresses are dynamically assigned to nodes and are

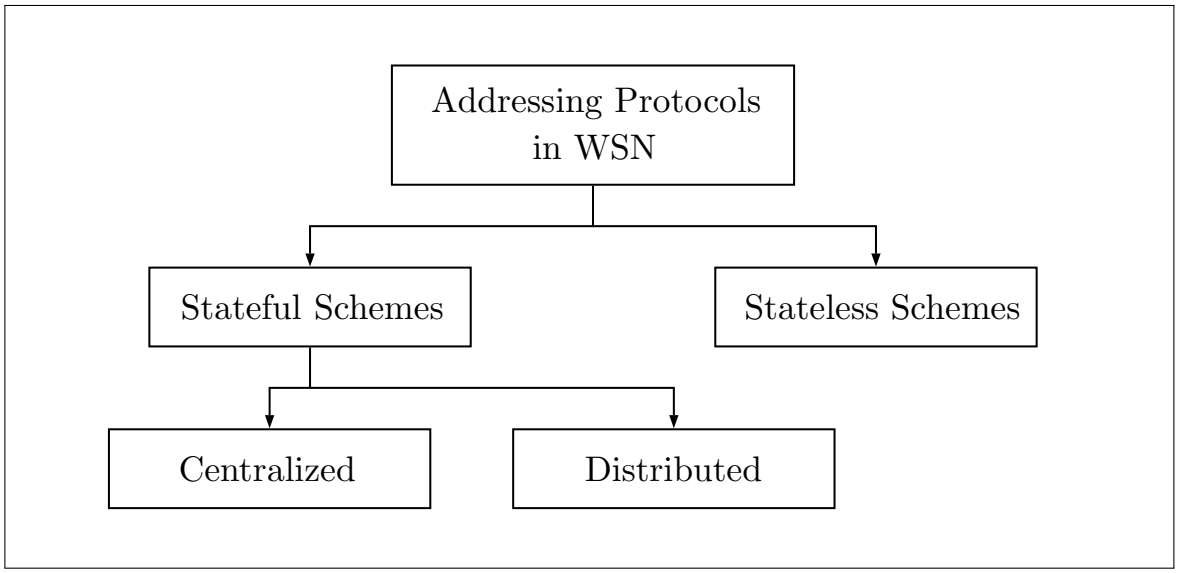


FIGURE 1.2. Classification of addressing protocols (F. Ye & Pan, 2009).

shorter than the unique, network-wide IDs, thus avoiding incorporating heavy overhead to every packet transmission. MAC addresses are unique only within a 2-hop neighbourhood, allowing reutilization of the addresses across the network.

Conversely, the initialization problem is treated as a whole by Ingelrest et al. (2010). The solution was extensively field-tested in several environmental monitoring applications. The nodes discover their neighbours by listening for a standard beacon transmission. This beacon is only transmitted by nodes that have already detected a beacon, except for the sink, which transmits the first beacon, thus triggering a neighbour discovery wave across the entire network.

This work will continue with the line exposed by Schurgers et al. (2002). In the presented protocol, nodes will have to choose among a fixed amount of addresses (in this case, preambles). Furthermore, nodes will not wake up upon reception of a beacon package, but in a random basis as it will be exposed in Chapter 4. Moreover, a conflict resolution protocol will be included (as in Schurgers et al. (2002)), in order to resolve potential conflicts where duplicate addresses exist in one neighbourhood.

1.3 Research Goals and Contributions

The main goal of this thesis is providing an effective means to achieve multiple access in a wireless sensor network under the channel-dependent reverse channel training scheme. In order to achieve this, the present work is structured around the following specific goals:

1. Design a protocol that provides medium access based on a finite number of local addresses. These addresses will concretely be a sequence of symbols, a *preamble*.
2. Design a set of preambles that fulfils the necessary conditions to provide effective multiple access.
3. Developing a complete communications simulator with the specifications of LATINA UC testbed, in order to obtain results of the performance of the proposed algorithms.

The main contribution of this work is a protocol that regulates medium access control under low SNR scenarios, where a regular SISO transmission cannot be carried out. The point-to-point scheme will be the same as in Kettlun (2014), which attains the CSI and beamforming SNR gain by reverse channel training. We propose a protocol that extends this scheme to a wireless network, by means of multiple preambles that act as physical addresses. These addresses are assigned dynamically, and are unique in every 2-hop neighbourhood of the network, following the approach taken in Schurgers et al. (2002). Multiple access is guaranteed by spatial division, where source and destination nodes beamform against each other.

1.4 Thesis Structure

This thesis is organized as follows. Chapter 2 will describe the problem and set requirements that the protocol must fulfil.

Chapter 3 will address the physical layer part of the protocol, determining and testing the way in which preambles can be used as physical addresses.

Chapter 4 will first explain the protocol at a higher layer level, and thereafter will evaluate its performance.

Finally, Chapter 5 will conclude the thesis and pose future lines of work.

2. DETAILED PROBLEM STATEMENT AND REQUIREMENTS

This Chapter will describe the problem that must be addressed, followed by the specific requirements that the protocol must fulfil.

2.1 Problem Description

RCT schemes need higher layer protocols that are compatible with the way in which data transmissions are performed in order to be implemented. The main problem that the RCT scheme faces is that nodes are not able to identify operational information (such as a source or destination address) from the packet prior to attaining the beamforming SNR gain. Under this situation, the effective SNR is too low to decode any data, making it impossible for a node to tell whether the packet was meant to be received by it or not. This way, if the scheme exposed in (Kettlun, 2014) was directly performed, many nodes would simultaneously respond to a preamble transmission call of a node, causing unintelligible collisions at the source node.

A physical addressing scheme is proposed to overcome this problem. This implies that different physical training sequences –preambles– will be assigned to nodes such that each node possesses a unique preamble in a given neighbourhood. These preambles will be assigned on terrain, making the protocol adaptable and scalable. The whole protocol will be extensively described in the following sections. The specific questions that must be answered in order to attain a functional protocol are set forth below.

1. *How do nodes send unicast messages?* Suppose a node Ω_{s} want to reach a destination node Ω_{d}. Since no data is allowed to be sent in the first transmission, it is not possible to send a node ID. Given this, how can Ω_{d} know the message was intended to be received by it? How will other nodes know the message was not intended to be received by them? As it will be stated later on, different preambles will be leased by nodes, being used as physical addresses, as in (Schurgers et al., 2002).

2. *How do nodes learn about the preamble they own?* It is desired that nodes learn what their preamble will be only once they are deployed, not before.
3. *How do nodes learn who their neighbours are?* Nodes have to know how to refer to their immediate neighbours. Every transceiver must know which nodes conform its neighbourhood, and what preamble each of them is leasing.
4. *How is the network established in the first place?* When a node Ω_0 is the first one to wake up in a network and finds no neighbours to join to, it must establish a brand new network on its own.
5. *How do nodes join an existing network?* It is also desired that, once a network is completely operating, new nodes could be added to the network so as to either expand the coverage area or add sturdiness to the network.
6. *How are the aforementioned preambles constructed?* Preambles are a sequence of L symbols. These symbols cannot be any sequence; they have to meet some properties that will be exposed.

These questions will be answered in detail in Chapters 3 and 4. They shall provide an effective solution to the multiple access provision problem.

2.2 Required Specifications for the Solution

In order to determine the extent to which the solution actually provides multiple access, some key indicators are drawn. The following requirements must be met.

1. *Connectivity Ratio* (ζ): A connected node is defined as one that can reach the accumulator either directly or using other nodes as relays. The connectivity ratio expresses the fraction of the total nodes in a network that are connected nodes after the initialization protocol has run. It will be required to meet $\zeta \geq \zeta_{\min}$.
2. *No duplicate PHY addresses in one neighbourhood*: Preambles will serve as physical addresses for nodes. Therefore, and following the argument exposed by Schurgers et al. (2002), no preamble must be repeated in any 2-hop neighbourhood.

3. *Scalability*: The protocol must operate correctly and initialize effectively regardless of the number of nodes conforming the network.
4. *Missed Packet Probability* (p_{miss}): At a physical layer level, this probability measures how often a correct packet is not detected by the intended receiver at the correct instant. The upcoming protocol must meet $p_{\mathrm{miss}} \leq p_{\mathrm{miss,max}}$
5. *False alarm packet detections* (p_{FA}): A false alarm detection is defined as a packet declaration when there should not have been one. False alarms can be triggered due to three causes. They are detection under noise only, detection of the correct preamble at a wrong instant, and detection of an alien preamble. However, the latter one broadly dominates over the other two. Hence, p_{FA} will be approximated by the occurrence of this event. The proposed protocol will satisfy $p_{\mathrm{FA}} \leq p_{\mathrm{FA,max}}$.

These performance indicators will be evaluated in Sections 3.4 and 4.5.

3. PREAMBLES AS AN ADDRESSING TOOL

This Chapter will thoroughly explain how different preambles serve as physical addresses and provide multiple access to the network. First, the signal model will be clarified, considering the whole transmission-reception chain. Then, we will explain how these preambles act as physical addresses and provide effective multiple access. Thereafter, we will state the way in which different preambles are obtained. Finally, a simulation will be shown in order to reaffirm the correct operation of these preambles.

3.1 Signal Model

MIMO communications consider the use of multiple antennas at both the source node Ω_{s} and the destination node Ω_{d}. A MIMO system with N_t transmit antennas and N_r receive antennas can be modelled as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{3.1}$$

where $\mathbf{x} \in \mathbb{C}^{N_t}$ is the transmitted vector, $\mathbf{y} \in \mathbb{C}^{N_r}$ represents the received vector, $\mathbf{n} \in \mathbb{C}^{N_r}$ is the complex AWGN vector with i.i.d. zero mean and ν^2 variance elements, and $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the Rayleigh channel matrix. If sufficient channel state information (CSI) is attained at both the transmitter and receiver, the signal can be precoded at the transmitter and weighted at the receiver to obtain maximum diversity gain (Kettlun, 2014). Equation 3.1 then reduces to

$$y = \sigma_1 x + n, \tag{3.2}$$

where x is the transmitted data symbol, y represents the received symbol and n is a complex AWGN value. σ_1 represents the equivalent SISO channel gain, which can be approximated by the gain of a Nakagami channel (Rosas & Oberli, 2013).

It is precisely because of this gain that data cannot be sent on the very first transmission between two nodes. Acceptable SNRs at the receiver are only reached when the channel state has been successfully estimated and a singular value gain σ_1 has been attained. A successful way to achieve this gain is through the Ping Pong Payload (PPP) scheme (Kettlun, 2014), which is a particular case of RCT. It proceeds as follows.

Since data cannot be immediately decoded, a group of L symbols –a *preamble*– is conveyed when a communication wants to be initiated. This first message is called a *Ping* transmission. The signal processing at the transmitting node takes place as depicted in Figure 3.1.

Preamble $\mathbf{a}$ is stored at the transmitter, and is differentially encoded, producing $\mathbf{c}$. The encoding-decoding procedure is performed so as to coherently add received signals from different independent antennas at the receiver. Afterwards, signal $\mathbf{c}$ is split in N_t pieces at the MIMO combiner, transmitting each of these pieces through a different antenna. This forms a staggered preamble $\mathbf{x}$ such that L/N_t training symbols are sent sequentially from each branch, as shown in Figure 3.2. This procedure ensures an energy efficient channel estimation at the receiver node (Muñoz & Oberli, 2012).

The received signal $\mathbf{y}$ at the receiver node at instant k after AWGN has been added is

$$\mathbf{y}[k] = A\mathbf{H}\mathbf{x}[k] + \mathbf{n}[k], \tag{3.3}$$

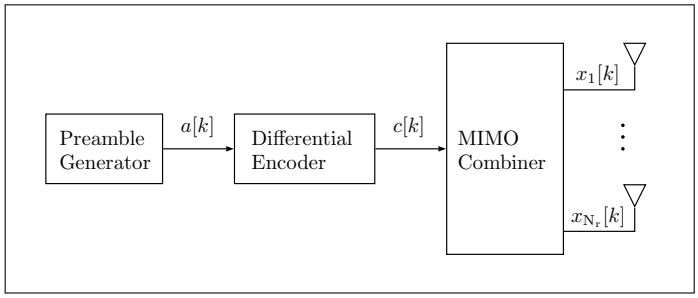


FIGURE 3.1. Transmitter Architecture.

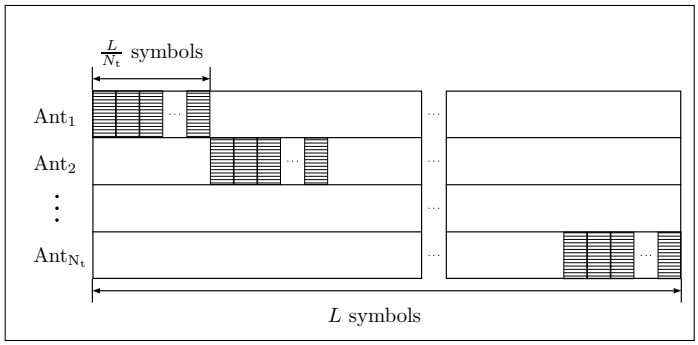


FIGURE 3.2. Distribution of training symbols per antenna, with a time-orthogonal preamble.

where $\mathbf{y}[k]$ is the received signal at instant k, A represents the large-scale propagation effects, $\mathbf{H}$ is the complex Rayleigh channel instance between transmitter and receiver nodes, $\mathbf{x}[k]$ is the sent symbol at instant k, and $\mathbf{n}[k]$ is AWGN noise of power N_0. The receiver signal processing is described in Figure 3.3.

This procedure involves an ideal Automatic Gain Control (AGC) and a differential decoder. The first step is performed so that the signal's amplitude remains constant regardless of the distance between the transceivers and the channel realization. After going though the ideal AGC and being differentially decoded, the signal is coherently added to get $\mathbf{z}$, the signal entering the correlator.

$$z[k] = A_{\mathrm{s}} a[k] + A_{\mathrm{n}} \omega[k], \tag{3.4}$$

where $\omega[k]$ is random coloured noise comprising signal-noise and noise-noise elements, with parameters $\mu_\omega = 0$ and $\sigma_\omega^2 = 1$. A_{s} and A_{n} are the signal and noise equivalent amplitudes respectively. Even though these are not constant for every receiving branch, they can be assumed constant under a low SNR scenario. These values comply

$$A_{\mathrm{s}}^2 + A_{\mathrm{n}}^2 = E_{\mathrm{ref}}, \tag{3.5}$$

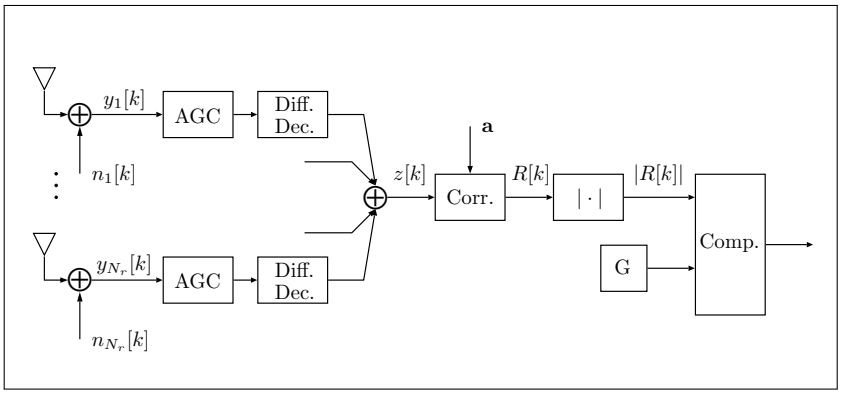


FIGURE 3.3. Receiver Architecture.

where E_{ref} is a value given by the AGC which for simplicity will be set to 1. If the receiver node is matched to the conveyed preamble $\mathbf{a}$, the output of the correlator is given by

$$R[k] = A_{\text{s}} \sum_{i=1}^{L} a[i+k]a^*[i] + A_{\text{n}} \sum_{i=1}^{L} \omega[i+k]a^*[i], \tag{3.6}$$

which is equivalent to

$$R[k] = A_{\text{s}} C_{\mathbf{aa}}[k] + A_{\text{n}} N[k]. \tag{3.7}$$

In this equation, $C_{\mathbf{aa}}[k]$ is the autocorrelation function of sequence $\mathbf{a}$, defined as

$$C_{\mathbf{aa}}[k] = \sum_{i=1}^{L} a[i+k]a^*[i], \tag{3.8}$$

and $N[k]$ involves the sum of all the noise accumulated in the samples in the correlator. Its mean value is $\mu_N = 0$, and its variance is σ_N^2. The preamble is fully matched with the correlator when $k = 0$. A packet detection must be triggered at this instant, and *must not* be triggered when $k \neq 0$. This decision is made comparing the correlator's output with a fixed threshold G. Therefore, the choice of G must be made so as to make sure the following two conditions are satisfied in most cases

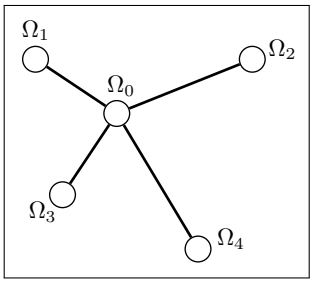


FIGURE 3.4. Example operating network. Every node is now leasing a different preamble, such that node Ω_i is leasing preamble $\mathbf{a}_i$.

$$|R[0]| > G \tag{3.9}$$

$$|R[k]| < G, \quad k \neq 0. \tag{3.10}$$

These conditions will be satisfied by finding a set $\mathcal{A}$ of preambles that have auto-correlation functions with low peak sidelobes (see appendix A), together with a smart threshold G choice. Both elements are studied and determined in this work.

Upon positive reception of the Ping, a channel estimation process takes place at Ω_{d}. It then replies with a *Pong* transmission, which is the same preamble $\mathbf{a}$ precoded with the acquired CSI. The same transmit-receive procedure takes place conversely (Ω_{d} is now the transmitter and Ω_{s} becomes the receiver). Upon correct reception of the Pong, Ω_{s} determines the necessary coefficients of $\mathbf{H}$, and an alternating back and forth communication of packets can now begin between both nodes. We denote these to be *Payload* transmissions. The MIMO gain has now been attained, and provides an SNR gain.

3.2 Overhearing Nodes

Take for instance, an already initialized operating network, such as the one in Figure 3.4, where each node owns a preamble. A node owning preamble $\mathbf{a}_0$ continuously listens to the preamble $\mathbf{a}_0$ during wake times. Let us suppose now that a node Ω_0 (owning preamble $\mathbf{a}_0$) wants to convey a message to node Ω_1 (owning preamble $\mathbf{a}_1$). In order to do

this, Ω_0 transmits preamble $\mathbf{a}_1$. The transmission-receive procedure followed is the same as exposed in Section 3.1. As reviewed in that Section, the signal entering the comparator at Ω_1 is

$$|R_1[k]| = |A_{1\mathrm{s}} C_{11}[k] + A_{1\mathrm{n}} N_1[k]| \,. \tag{3.11}$$

The value of $|R_1[k]|$ will determine whether Ω_1 responds with a Pong transmission or not, as studied in the previous Section.

At the same time, another node (e.g. Ω_2) overhears the preamble transmitted. It owns a preamble $\mathbf{a}_2$, different to $\mathbf{a}_1$. The output of the correlator at Ω_2 is

$$R_2[k] = A_{2\mathrm{s}} \sum_{i=1}^{L} \mathbf{a}_1[i+k]\mathbf{a}_2^*[i] \tag{3.12}$$

$$+ \quad A_{2\mathrm{n}} \sum_{i=1}^{L} \omega_2[i+k]\mathbf{a}_2^*[i], \tag{3.13}$$

which reduces to

$$R_2[k] = A_{2\mathrm{s}} C_{12}[k] + A_{2\mathrm{n}} N_2[k], \tag{3.14}$$

with $C_{12}[k]$ being the cross-correlation between sequences $\mathbf{a}_1$ and $\mathbf{a}_2$, defined as

$$C_{12}[k] := \sum_{i=1}^{L} \mathbf{a}_1[i+k]\mathbf{a}_2^*[i]. \tag{3.15}$$

Since in this case, packet detection is not desired, the condition

$$|R_2[k]| < G, \quad \forall k \tag{3.16}$$

must hold at all times. This condition, along with the ones in equations (3.9) and (3.10) are crucial to determine the optimal value of threshold G, which will be determined using simulations in Section 3.4. The determination of adequate preambles will be exposed in the next Section.

3.3 Preamble Construction

In order to provide effective multiple access, a number N_A of preambles are needed. For the purpose of this work, and based on expressions (3.9), (3.10) and (3.16) derived in the previous sections, we need a set of N_A preambles $\mathcal{A} = \{\mathbf{a}_0, ..., \mathbf{a}_{N_A-1}\}$ which meet the following two conditions.

1. The peak sidelobe (PSL) (see Appendix A for the definition) of each autocorrelation function is lower than a maximum allowable PSL $\psi_{\max}$

$$\mathrm{PSL}(\mathbf{a}_i) \leq \psi_{\max} \quad 0 \leq i < N_{\mathrm{A}} \tag{3.17}$$

2. For any $\mathbf{a}_i$, $\mathbf{a}_j \in \mathcal{A}$, the maximum absolute value of their crossed correlation function (CCF) ξ_{ij} is lower than a maximum acceptable $\xi_{\max}$.

$$\xi_{ij} \leq \xi_{\max} \quad 0 \leq i, j < N_{\mathrm{A}} \tag{3.18}$$

The first condition has broadly been studied by researchers. The optimum is achieved with Barker sequences (Barker, 1953), which attain

$$\mathrm{PSL}(\mathbf{a}) = 1 \tag{3.19}$$

Nevertheless, such sequences only exist for binary vectors of length 2, 3, 4, 5, 7, 11 and 13. Longer sequences are sought so as to conform a packet detection preamble for wireless communications. This stimulus leads to the following optimization problem formulation

Algorithm 1 Sequence finding

1: Find set $\mathcal{B}$ of all binary sequences such that $\forall \mathbf{b} \in \mathcal{B}$, $\mathrm{PSL}(\mathbf{b}) = \psi^*$ using the algorithm by Cohen et al. (1990).
2: Set $\xi_{\max} \leftarrow 0$, $\mathcal{A} \leftarrow \emptyset$
3: **while** $\#\mathcal{A} < N_{\mathrm{A}}$ **do**
4: $\quad \xi_{\max} \leftarrow \xi_{\max} + 1$
5: $\quad$ Build Ξ, a symmetric matrix containing the cross-correlation maxima ξ_{ij} between every pair of sequences $\mathbf{b}_i$, $\mathbf{b}_j \in \mathcal{B}$.
6: $\quad$ Find $\mathcal{A} \subset \mathcal{B}$, a set of sequences such that for every $\mathbf{a}_i$, $\mathbf{a}_j \in \mathcal{A}$, $\xi_{ij} \leq \xi_{\max}$. It is only needed to look at the matrix Ξ already built.
7: **end while**

$$\begin{aligned} \psi^* = \min_{\mathbf{a}} \quad & \mathrm{PSL}(\mathbf{a}) \\ \text{s.t.} \quad & \mathbf{a}[k] = 1, -1 \quad \forall k \in 1, 2, ..., L \end{aligned} \tag{3.20}$$

This optimization problem has gathered a lot of interest among researchers, and to the best of our knowledge, the minimum PSL has been found for sequences up to length 74 (Leukhin & Potekhin, 2013). Vectors this long are far beyond the preamble length considered in this work. A method for finding such sequences is described in (Cohen, Fox, & Baden, 1990), and improved in (Mow, 1993). This algorithm is made on a Branch and Bound basis. As an input, it needs the length L of the sequence and the maximum PSL accepted $\psi_{\max}$, for it to retrieve the set $\mathcal{B}$ containing all sequences $\mathbf{b}$ of length L that achieve

$$\mathrm{PSL}(\mathbf{b}) \leq \psi_{\max} \tag{3.21}$$

In this case, we select $\psi_{\max} = \psi^*$. Algorithm 1 is used to find the set $\mathcal{A}$ of N_A sequences, each of length L which fulfil both conditions exposed above simultaneously. Starting from the set of sequences $\mathcal{B}$ given by the algorithm in (Cohen et al., 1990), cross correlation maxima are calculated for every pair of sequences in this set. Using this information, a set of preambles are selected according to the second condition exposed above. We are interested in setting $\xi_{\max}$ as small as possible.

After the execution of this algorithm, the desired set $\mathcal{A}$ of N_A sequences of length L each is found. It matches the conditions

$$\begin{array}{rcccll} \max\limits_{1\leq k<L} |C_{ii}[k]| & = & \mathrm{PSL}(\mathbf{a}_i) & \leq & \psi_{\max} & \forall \mathbf{a}_i \in \mathcal{A} \\ \max\limits_{-L<k<L} |C_{ij}[k]| & = & \xi_{ij} & \leq & \xi_{\max} & \forall \mathbf{a}_i, \mathbf{a}_j \in \mathcal{A} \end{array} \tag{3.22}$$

which is the set consisting of the preambles that will be used as addresses. The next Section will test the performance of the determined preambles using a simulator. Different received SNRs will confront various thresholds G. This will lead us to choosing a threshold that guarantees a correct multiple access provision.

3.4 Physical Layer Simulation Results

A set of preambles $\mathcal{A}$ was found using the algorithm proposed in Section 3.3. It is hypothesized that 10 preambles are enough to satisfy the forthcoming higher layer requirements. This will be confirmed with simulations in Section 4.5. The set of preambles found meet the conditions shown in Table 3.1. The explicit preambles selected are detailed in Appendix B.

A MATLAB-based simulator was built in order to recreate the process of transmitting a preamble, depicted in Figures 3.1 and 3.3. The simulator was calibrated using theoretical bit-error probabilities. Two types of simulations were carried out. The first one involves a receiver node with a preamble correlator matched to the sender's one. These simulations are useful to measure how often a node misses a correct preamble, p_{miss}. The second type of simulation involves a receiver with a preamble that is not matched with the sent one.

TABLE 3.1. Physical Layer Simulation Parameters

$\#\mathcal{A}$	Number of preambles	10
$\psi_{\max}$	Maximum PSL accepted	3
$\xi_{\max}$	Maximum Cross Correlation Level accepted	10
L	Preamble length	32
N_{r}	Receive Antennas	4
N_{t}	Transmit Antennas	4

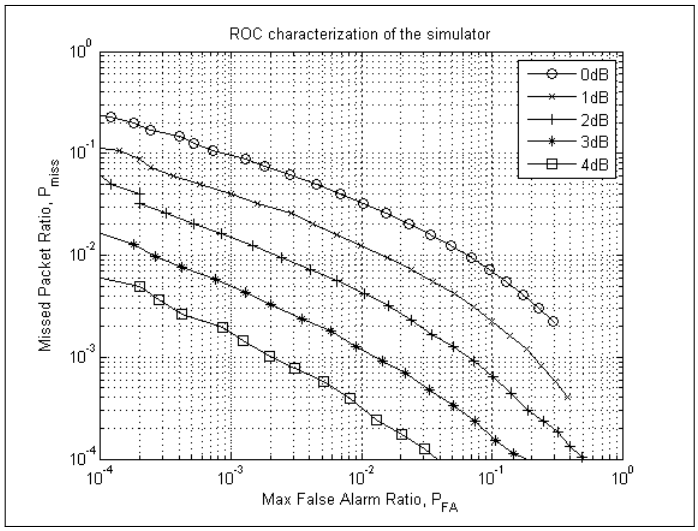


FIGURE 3.5. Receiver Operating Characteristics for a 4×4 MIMO system. Points in the curves denote different thresholds G used to detect packets. For instance a fixed threshold of $G = 11.25$ provides $p_{\text{miss}} \leq 10^{-2}$ and $p_{\text{FA}} \leq 10^{-2}$ for SNR $= 2\text{dB}$.

These simulations will estimate p_{FA}, the probability of falsely declaring a preamble, under the presence of an alien preamble. As discussed in Chapter 2, this is not the only source of false alarms. Other sources of false alarms are detection under noise only and detection of a correct packet at a wrong instant. Nevertheless, it will be assumed that the false alarms coming from an alien preamble detection broadly dominate the statistics, since they are the source of more than 99% of the declared false alarms.

For the first type, each preamble $\mathbf{a}_i \in \mathcal{A}$ is sent from a source node, and received with another node which is expecting to receive the same preamble. It was done 10^5 times with each preamble, totalling $10 \cdot 10^5$ simulations. For the second type, each of the preambles was transmitted and then correlated at receivers owning the other nine preambles, totalling $90 \cdot 10^5$ simulations. This procedure allows us to estimate the values of p_{miss} and p_{FA}. Different SNRs are also considered, ranging from 0dB to 4dB. Since various pairs of preambles perform differently, the worse performance pair is the one considered to establish p_{FA}.

Figure 3.5 depicts the receiver operating characteristics (ROC) for the system previously described. Each node has four antennas. The graph shows the existing contrast between the maximum incorrect code ratio, p_{ic} and the missed packet ratio p_{miss}. Each line represents the system's characteristics under a different average SNR measured at the input of the correlator at the receiver node. Under a given SNR, the degree of freedom is given by the threshold G. A higher value of G will suit the system in the left part of Figure 3.5, where low p_{FA} and high p_{miss} are achieved. On the other hand, a lower value of G will provide a high p_{ic} and a low p_{miss}. Depending on the system's requirements, the value of the fixed threshold G must be chosen. For instance, if both probabilities were to be minimized jointly, a fixed threshold of $G = 11.25$ provides $p_{\mathrm{miss}} \leq 10^{-2}$ and $p_{\mathrm{FA}} \leq 10^{-2}$ for SNR $= 2\mathrm{dB}$.

4. PROTOCOL OPERATION AT THE MAC AND NETWORK LAYERS

Until now, we have concluded that a set of 10 preambles that work sufficiently well at a physical layer level exists. We have not yet proven that ten preambles suffice at a network level to provide effective multiple access.

This Chapter will first explain how the protocol works when it has already been initialized. Then, the initialization process will be depicted, followed by the exposure of some situations that can show up in a network. Thereafter, a network model will be presented and subsequently applied into simulation that will finally confirm the validity of the preambles' addressing scheme at a network level.

4.1 Steady State Operation

The network is in steady state. Every node in the network leases one of N_{A} preambles $\mathbf{a}_i$, with $i = 0, \ldots, N_A - 1$. The lease is granted for a given amount of time. The preamble distribution is such that there are never two nodes leasing the same preamble in any 2-hop neighbourhood. Further, every node keeps two tables: one with the 1-hop neighbours it has directly communicated with, and 2-hop (hidden) neighbours it has heard about from its 1-hop neighbours. Both tables contain, for each node, its network wide, unique node ID and the preamble index $i \in \{0, \ldots, N_A - 1\}$ that indicates the preamble currently leased by the corresponding node.

Regular communication between two neighbouring nodes happens as follows. During wake times, each node listens continuously for its own preamble $\mathbf{a}_i$. Thus, for communicating with a desired destination node Ω_{d}, source node Ω_{s} looks up Ω_{d}'s preamble index i from its 1-hop neighbour table and transmits preamble $\mathbf{a}_i$. We call this a *Ping* transmission. The data field of a Ping is empty.

Upon positive reception of the Ping, node Ω_{d} replies with the same preamble $\mathbf{a}_i$, followed by data. We call this a *Pong* transmission. The data field of a Pong contains the node ID of Ω_{d} and a corresponding checksum (why these are necessary becomes clear when we

look at how new nodes enter the network and how preamble conflicts are resolved). Upon correct reception of the Pong (valid checksum) and validation of Ω_{d}'s node ID by Ω_{s} in its 1-hop neighbour table, an alternating back and forth communication of packets with arbitrary payload can begin. We denote these to be *Payload* transmissions. They always begin with preamble $\mathbf{a}_i$, followed by the source node ID (an *address* field, as in Pongs), followed by an address checksum and the actual payload (*data* field), which contains descriptive parameters such as length of the data field, as well as data itself, ACKs, etc. Before decoding the data field of any Payload frame, the address checksum must be validated and the node ID verified in the 1-hop neighbours table. The exchanges of payload data can be bi-directional or uni-directional. Payload that flows in one direction typically entails opposite-direction acknowledgements.

In summary:

1. Ping = preamble,
2. Pong = preamble + source address + address checksum
3. Payload = preamble + address + address checksum + data,

whereby it is to be noted that Payload and Pong transmissions are, in essence, the same kind, because both entail the transmission of a preamble followed by data. In the context of this work, however, Pong frames have a more specific definition than Payload frames. Further Payload frame "flavours" shall be defined as needed in order to implement the functionality described herein.

The tail instant of every Ping, Pong and Payload transmission triggers a timer t_l at the transmitting node. The timer expires at $t_l = T_L$ seconds, after which the node declares the Ping-Pong over and begins listening for Pings again, rather than a Pong or Payload frame.

The leased preamble is primarily a PHY address. It provides a means for uniquely identifying every node in the network at the PHY level. Medium access control is primarily regulated in a CDMA manner by means of the preamble owned by each node. Because preambles may be reused only by nodes 3-hops or further away from a given source node, the node ID can be used at the MAC level as a secondary means for validating that the

medium was accessed (i.e. the link was established) with the correct destination node and for ruling out that a link was spuriously established with a 3-hop (or further) neighbour.

The Ping-Pong link-establishment phase described above with node IDs and preambles can be kept confined exclusively to the PHY and MAC layers. In a strict sense, only the content of the data field of Payload transmissions needs forwarding to higher layers, as if it was a network with traditional single antenna nodes. Thus, in principle, any suitable protocol stack can be put on top of the PHY/MAC stack described above. It is to be noted, though, that such protocols must meet the latency requirements so that the Payload-ACK back and forth cycle is kept within the coherence time of the channel. It may be desirable to make the node ID also available to higher layers.

Nodes can freely speak to each other using different preambles to access logical channels. Once the beamforming link is established and data starts to be transmitted, nodes taking part of this communication will only transmit and listen in the direction of the other, making their transmission undetectable for other nodes. Likewise, other node's transmissions will not affect theirs since they will be beamformed towards each other, ignoring incoming transmissions from other directions.

4.2 Network Initialization

An *unstructured* network (Li, 2008) is a set of nodes that have neither initial information about the network topology nor an assigned channel for communication. The goal of the upcoming procedure is to turn an unstructured network into one in which each node knows about their neighbours, and is assigned a unique preamble among a 2-hop neighbourhood, as described in Section 4.1. The starting point is a set of nodes that have been deployed over a terrain, and have been powered on so that they can conform a network. Nodes have a list of N_A hard-coded preambles in their memory. The following procedure assigns each node a locally unique preamble that acts as a PHY/MAC address, and lets every node in the network know about its 1- and 2-hop neighbours.

4.2.1 Node Initialization

After being powered on, a new node Ω_n triggers a timer t_{I}. The timer expires at $t_{\mathrm{I}} = T_{\mathrm{I}}$, after which Ω_n starts its initialization. $T_{\mathrm{I}} \sim U(0, T_{\max})$ is a random uniformly distributed variable which is different for every node. $T_{\max}$ is chosen high enough so as to ensure there is a low collision environment while initializing. Beginning with $i = 0$, with i an index that cycles through $i = 0, ..., N_A - 1$, Ω_n transmits preamble $\mathbf{a}_i$ and immediately thereafter switches to listening mode for preamble $\mathbf{a}_i$. The listening is done for at most T_L seconds. If this time window expired and no response with preamble $\mathbf{a}_i$ was detected, Ω_n switches back to transmit mode, transmits preamble $\mathbf{a}_{i+1|N_A}$, shifts to listening for $\mathbf{a}_{i+1|N_A}$ and so on.

Two situations may happen while Ω_n is subsequently transmitting all N_A preambles $\mathbf{a}_i$. The first one, is that an already initialized node Ω_x listens to an $\mathbf{a}_x$ call of Ω_n. The second option is that, after having cycled through all N_A preambles, no response was found and consequently, Ω_n deduces it is the first initialized node within its neighbourhood.

Let us first demonstrate how a node Ω_n joins an existing network. Eventually, node Ω_x, camped in the network and owner of preamble $\mathbf{a}_x$, responds to the $\mathbf{a}_x$ call of Ω_n. Because for Ω_x this is a regular Ping transmission, it replies with a Pong providing its node ID. Ω_n adds then Ω_x to its 1-hop neighbour table. An alternating back and forth data communication of packets with preamble $\mathbf{a}_x$ followed by arbitrary data can begin between Ω_n and Ω_x. First, Ω_n informs Ω_x of its node ID (regular address field) and that it wishes to enter the network (data field, indicating that it does not own a preamble yet). Ω_x will provide a list of its 1-hop neighbours (with their owned preambles and network IDs). By revising Ω_x's neighbour list, Ω_n will update its 1- and 2-hop neighbour lists.

After searching for neighbour nodes for all preambles, Ω_n picks a preamble that is free in both 1-hop and 2-hop neighbour tables and informs each of its 1-hop neighbours about its choice, along with the 1-hop table constructed by it. Every neighbour (e.g. Ω_x) adds Ω_n to its 1-hop neighbour table. By revising the 1-hop neighbour list provided by Ω_n, every neighbour notifies its own neighbours about the newly found node Ω_n, and each

of them updates its 2-hop neighbours. In other words, the 2-hop neighbours of Ω_n update their 2-hop neighbour table.

It may happen, nevertheless, that after a node has exchanged neighbour information with every neighbouring node (and therefore has successfully constructed its 1- and 2-hop neighbour tables), there are no preambles available for it to lease. In this situation, the entering node Ω_n declares the network is too saturated for it to enter and does not get in the network.

The recently explained process fully describes how a node enters an already existing network. Now, suppose the new node did not find any neighbours to report to, leading the node Ω_n having to establish a brand new network itself. This happens when the above search yields an empty 1-hop table, leading Ω_n to pick a random preamble. Then, it inspects the 1-hop table for neighbours that must be informed of the chosen preamble, which trivially leads to not informing anyone about it. Thereafter, the node considers itself camped on the network –a 1-node network in this case. From a logical point of view, however, if Ω_n truly was the first node of the network, then the network is formally established this way in steady state operation (all its nodes listening for their leased preambles). Future nodes can enter the network by the method described above.

It may happen that Ω_n is to become linked to a larger, already established network, but was not yet able to do so because one (or several) bridging nodes between Ω_n and the network are still missing. Further new nodes that find Ω_n may appear and build jointly an isolated network, until eventually a bridging node Ω_b finds 1-hop neighbours in both networks. If, by chance, the set of preambles used by the 1-hop neighbours of Ω_b in the isolated network is disjoint from the corresponding set used by the 1-hop neighbours of Ω_b in the larger network, then both networks will be merged without difficulty. Otherwise (i.e. Ω_b ends with two 1-hop neighbours with the same preamble), a preamble conflict ensues and Ω_b cannot yet enter the network.

How a preamble conflict is found and solved is detailed in the following Subsection.

4.2.2 Conflict Detection and Resolution

A preamble conflict takes place when a bridging node Ω_b results with two (or more) neighbours with the same preamble assigned as a PHY address. In this case, Ω_b will not be able to enter the network, because its Ping transmission with the conflicting preamble $\mathbf{a}_c$, will generate colliding Pong responses. The collision is revealed by an invalid address checksum. After waiting for at least T_L seconds, Ω_b will reattempt to Ping transmit preamble $\mathbf{a}_c$. After a total of N_F attempts (e.g. $N_F = 3$), it will enter listening mode for $\mathbf{a}_c$ one last time for T_L seconds. If no valid Pong response is received after that, Ω_b will blacklist preamble $\mathbf{a}_c$ as a conflicting preamble and move on with the scan for other 1-hop neighbour camped with the remaining untested preambles. Upon finishing its scan, Ω_b will go back to sleep mode again for T_S seconds before trying to adhere to the network again, hoping that the preamble conflict is solved by then.

The conflicting $\mathbf{a}_c$ neighbours, on the other hand, keep a counter of Ping attempts received c_p. The counter is reset as soon as a successful Payload is received (address checksum valid) or after a given time $T_R \gg T_L$ went by since the previous Ping (every new Ping attempt received within this time window increments the counter and resets this timer). Whenever c_p reaches N_F (or perhaps $N_F - 1$ to be lenient for some over-the-air lost Ping transmissions), it will be an indication that a neighbour is trying to reach this node but cannot obtain a valid Pong source address. The conflicting nodes can infer a preamble conflict exists and are obligated to resolve it.

These nodes reset their chosen preamble and start looking for another one, as if they had not been initialized before. They follow the same procedure detailed in Section 4.2.1, Ping transmitting all preambles, $\{\mathbf{a}_0, ..., \mathbf{a}_{N_A-1}\}$ and then choosing a random new preamble from the available ones. The conflicting nodes may now choose different preambles, making it now possible for Ω_b to freely join the network after its timer t_{s} expires. In this situation, the conflict would be successfully solved. However, if both nodes again choose the same preamble, Ω_b will again trigger a conflict alarm after a time T_{S} and running over all preambles. This will lead the conflicting nodes to try to select again other preambles.

Under the knowledge that this process could end on an endless loop, a node that cannot enter the network N_H (e.g. $N_H = 3$) consecutive times decides that the network is too saturated and does not gets in it.

This procedure builds up a collision-free network, because 1-hop and 2-hop (hidden) neighbours are guaranteed to own different preambles and thus communicate over a different logical channel.

4.3 Further Situations

A few circumstances that need to be studied can emerge when the network is healthily operating. A group of new nodes may be intended to enter the network, a node may run out of power and address conflicts may occur due to changes in the network pattern. These situations are detailed in this Section.

4.3.1 A new node arrives

A network that is correctly operating may want to be expanded adding a group of new nodes to it. The incorporation of these new nodes to the network is not, in any way, different to the process described in Section 4.2.1. New nodes start searching for their neighbours and choose a preamble that is different from all its 1- and 2-hop neighbours' preamble.

4.3.2 A node runs out of power

Suppose a node Ω_{o}, who owns preamble $\mathbf{a}_{\mathrm{o}}$ has run out of power. A neighbouring node Ω_{n} is trying to reach it by sending a Ping transmission. After waiting for a time T_{L}, Ω_{n} tries to reach him again. It retries N_{D} times, before declaring the node dead and taking it out of its 1-hop neighbour table. Furthermore, Ω_{n} informs its neighbours that preamble $\mathbf{a}_{\mathrm{o}}$ is free to use again.

4.3.3 A node finds an address conflict

Preamble conflicts occur not only when nodes are being initialized, but also when topology changes occur due to changing propagation conditions or other mobility patterns

of the environment. New links can be created, making nodes to become neighbours when they were not in the past.

For instance, suppose a node Ω_b has two or more neighbouring nodes that respond to a conflicting preamble $\mathbf{a}_c$. This conflict was generated due to an unexpected change in the network topology. Node Ω_b will detect the existence of a conflict upon reception of multiple Pong transmissions. The detection and resolution is the same as the one explained in Section 4.2.2.

4.4 Network Model for Simulation at the MAC Layer

A network is described by a directed graph $\mathcal{G} = (N, E)$, where N is the set of nodes and E is the set of links. We say that two nodes Ω_i and Ω_j are neighbours if $(i, j) \in E$. We also assume that $(i, j) \in E \iff (j, i) \in E$.

In order to learn the conditions under which two nodes are neighbours, the received power must be studied. Path loss is modelled with the Friis Equation (Goldsmith, 2005). The average received power P_{r} with a transmitted power P_{t} is

$$P_{\mathrm{r}} = K\psi P_{\mathrm{t}} \left(\frac{\lambda}{4\pi d_0}\right)^2 \left(\frac{d_0}{d}\right)^{\alpha} \tag{4.1}$$

where K is a constant gain that includes antenna gains, diversity and array gains, amongst others; ψ is the log-normal shadow fading, such that $10\log\psi \sim N(0, \sigma_{\psi_{\mathrm{dB}}})$, where $\sigma_{\psi_{\mathrm{dB}}}$ ranges from 4 to 13 dB (Goldsmith, 2005); λ is the carrier's wavelength; d_0 is a reference distance for the antenna far-field; d is the distance between the nodes and α is the path loss exponent.

The criteria to determine whether two nodes are neighbours will be SNR based when a Ping transmission is in the air. This is, when the beamforming gain has not been attained yet. The SNR at this point is given by the expression

$$\begin{aligned} \mathrm{SNR\ dB} &= 10\log_{10} P_{\mathrm{r}}\ \mathrm{dBm} - 10\log_{10}(N_0 W)\ \mathrm{dBm} \\ &\quad -N_{\mathrm{f}}\ \mathrm{dB} - D_{\mathrm{loss}}\ \mathrm{dB} \end{aligned} \tag{4.2}$$

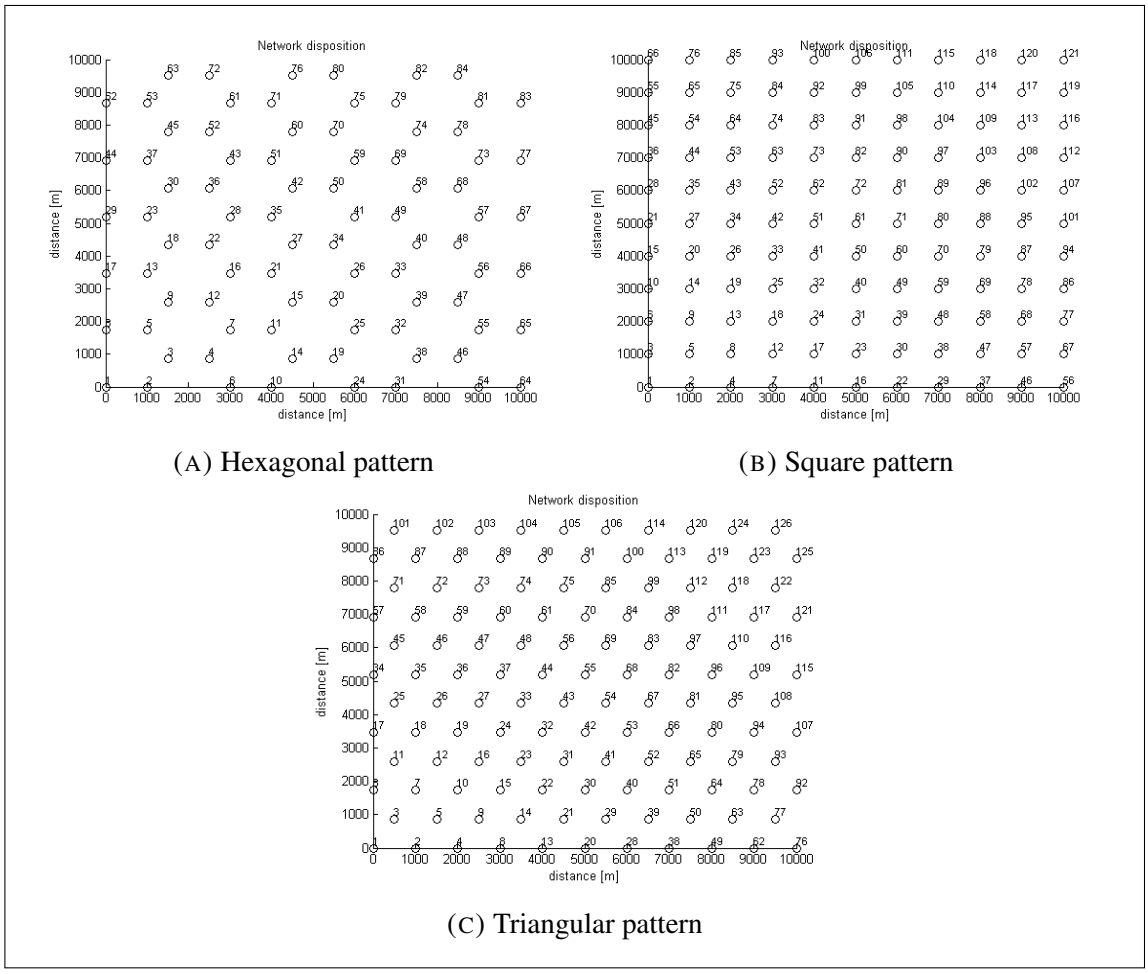
where SNR is the average signal-to-noise ratio measured at the entrance of the correlator, P_{r} is the received signal power, N_0 is the AWGN power, W is the transmitted signal bandwidth, N_{f} is the noise figure at the receiver, and D_{loss} is the loss in SNR due to differential encoding.

Two nodes will be considered neighbours if the SNR is higher than a threshold G_{SNR}. Let us emphasize the fact that, under a fixed disposition of nodes over a terrain, the links that conform a network $\mathcal{G}$ can vary dramatically due to the randomness of the wireless channel. Nevertheless, we will assume that the channel remains static during the length of the initialization process of a network.

4.5 MAC Layer Simulation Results

Random based disposal of nodes is not suitable, since this would lead to some nodes being physically disconnected from the network, given that they are just too far to be reached. In this work, three different node configurations are carried out. These are an hexagonal pattern, a square pattern and a triangular pattern (Figure 4.1). The patterns are sorted according to their node coverage density, being the hexagonal one the least dense, and the triangular the densest. The terrain is always a 10000×10000 metres region, with different number of nodes placed on it, according to the configuration. Adjacent nodes are always 1000 metres apart.

The protocol described in Sections 4.1 and 4.2 is tested for these node configurations using MATLAB-based simulations. A random node is chosen to be the *accumulator* (or *sink*), meaning that it is the final destination of every packet in the network. Time is arbitrarily parcelled in $T_{\max} = 10^3$ segments. Each node wakes up randomly in one of these slots, and carries out the neighbour recognition procedure mentioned. The only



(A) Hexagonal pattern

(B) Square pattern

(C) Triangular pattern

FIGURE 4.1. Nodes deployed in a 10000 $\times$ 10000-metre terrain. Adjacent nodes are 1000m metres apart. The channel realization will determine which nodes can listen to which others. Different channel realization will imply different network topologies.

exception to this rule is the accumulator node, which always wakes up at $t = 0$. It is assumed that the network's links do not change during all initialization procedure (i.e. the network links are assumed static). On every transmission, the SNR at the input of the correlator is calculated with Friis equation and link budget shown in Section 4.4. The parameters used are listed in Table 4.1.

In this higher-layer simulation, preambles are assumed to be detected validly if no interference was in the air. This is, the only source of erroneous detections is the interference of another simultaneously transmitting node. This assumption is made given that the physical simulations carried out in Chapter 3 guarantee that under the scenario where $\mathrm{SNR} > 2\mathrm{dB}$, less than 1% of the packets will be incorrectly detected, and less than 1% of the matched preamble packets will be missed. In other words, if a preamble $\mathbf{a}_0$ is conveyed to a node owning that same preamble, detection will be assumed to occur always. Similarly, when a node owning an alien preamble $\mathbf{a}_1$ receives $\mathbf{a}_0$, detection will be assumed to never occur. It is to be noted that, even though these assumptions bias the results from the real values, the deviation is not significant due to the results exposed about the PHY layer in Chapter 3. Along the same line, it will be assumed that signals that reach the correlator with $\mathrm{SNR} < 2\mathrm{dB}$ will never trigger a packet detection, regardless of whether the receiver is matched to the sent preamble or not. Hence, nodes are considered neighbours if the received SNR at the input of the correlator is higher than 2dB, and not considered neighbours if the received SNR is lower.

Figure 4.2 shows examples of situations where the channel has already been set, for each of the proposed patterns. Links are formed according to the SNR at the input of the correlator for a given shadow fading. If it is higher than $G_{\mathrm{SNR}} = 2\mathrm{dB}$, nodes can hear

TABLE 4.1. MAC layer simulation parameters

K	Antenna and Array gains	6.6 dB
P_{t}	Transmitted Power	3 dBm
$\sigma_{\psi_{\mathrm{dB}}}$	Shadow Fading Standard Deviation	4 dB
c	Speed of Light	$3 \cdot 10^8$ m/s
f_c	Carrier Frequency	900 MHz
λ	Wavelength ($= c/f_c$)	0.33 m
d_0	Reference Distance	10 m
α	Attenuation Exponent	3.5
N_0	Noise Power	-174 dBm
W	Transmission Bandwidth	13 kHz
N_{f}	Noise Figure	10 dB
D_{loss}	Differential Encoding Loss	2.3 dB

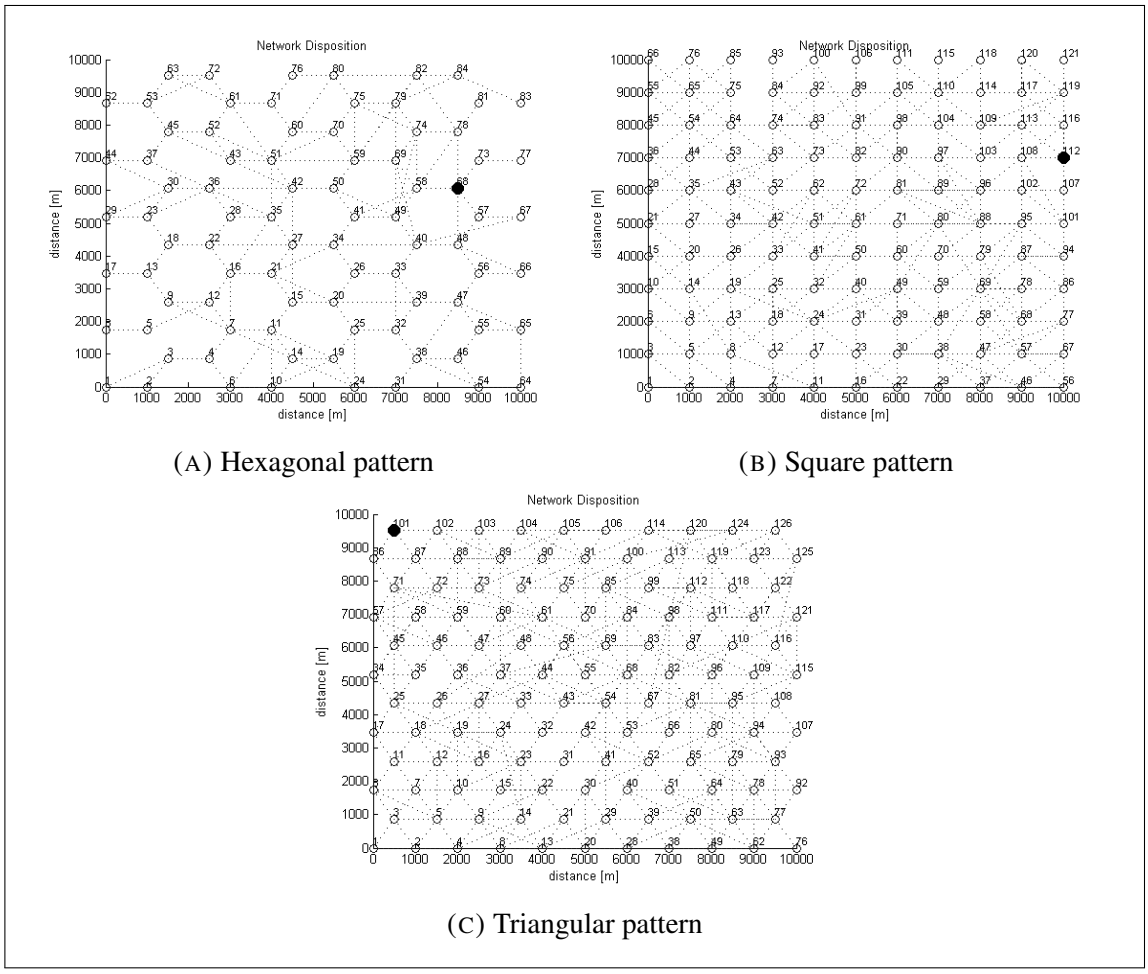


FIGURE 4.2. Link establishment for given node disposal, transmission power and channel realization. The accumulator is chosen randomly in each configuration, and it is shown in black in each of the cases. Nodes will start waking up randomly and form a network using the described protocol.

each other's transmissions, and are considered neighbours. As mentioned, these links will be assumed constant for the whole initialization process. These configurations introduce a first bound to the connectivity ratio, since the random shadow fading deprives some nodes of becoming reachable given a transmission power P_{t}. From here, the *potential extension* is defined as the maximum node ratio that can potentially become connected to the accumulator.

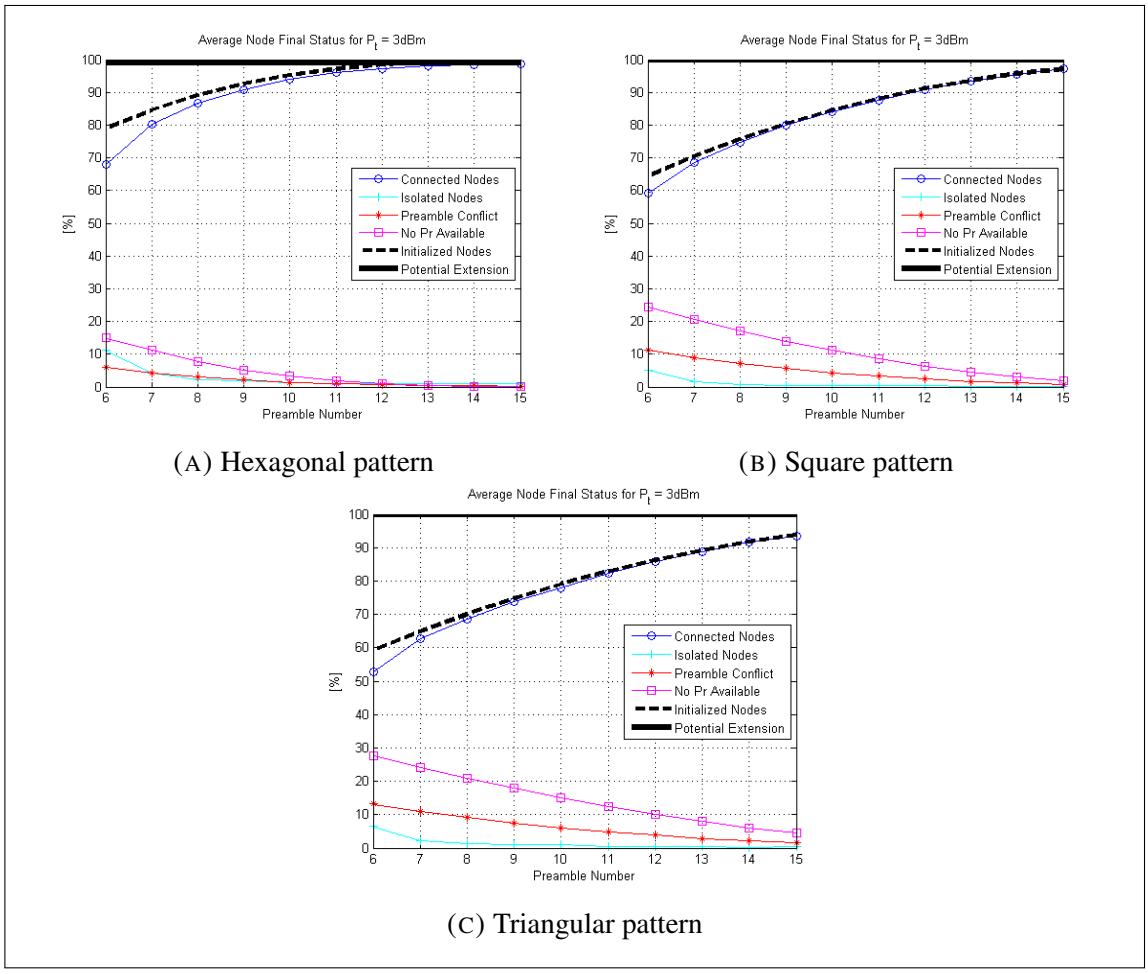
Under the aforementioned assumptions, and with connectivity ratio the key variable to be measured in this simulation, nodes can end up in one of four states.

1. Connected Node: A node is initialized and is able to reach the accumulator either directly, or using other nodes as relays.
2. Isolated Node: A node is initialized and cannot reach the accumulator.
3. Unavailable Preamble Node: A node is not initialized due to unavailable preambles.
4. Conflict Node: A node is not initialized due to an unresolved conflict.

For different number of available preambles, these four states and its statistics are shown in Figure 4.3 after simulating 10^4 times for each of the patterns. Channel realization and the order in which nodes wake up are different for each simulation. It can be seen that ten preambles suffice to correctly initialize more than 90% of the nodes for the hexagonal configuration, confirming the initial hypothesis taken at the beginning of this Section. However, denser node patterns require more preambles to ensure most nodes will be connected to the accumulator.

Another interesting trade-off shows up. Given a constant number of preambles and a node disposal, the transmission power P_{t} affects the overall connectivity of the network. For instance, if the transmission power was very high, every node in the neighbour would have more neighbours than it can handle, making it impossible for the protocol to correctly operate. On the other hand, if the transmission power was set too low, packets would not be able to correctly reach their destinations. Connectivity would decrease to zero due to the impossibility of the messages to reach the receiver. Thus, there is an optimal transmission power that can be determined. In Figure 4.4 and using 10 preambles, the optimal transmission power is determined for the three mentioned patterns.

This tradeoff illustrates the limitations of having a finite number of addresses to dispose of. The optimal solution implies a certain amount of redundancy in the network. This quantity is determined by the number of neighbours that each node has, on average. A driver that handles these values is the transmitted power, as depicted before. Under a



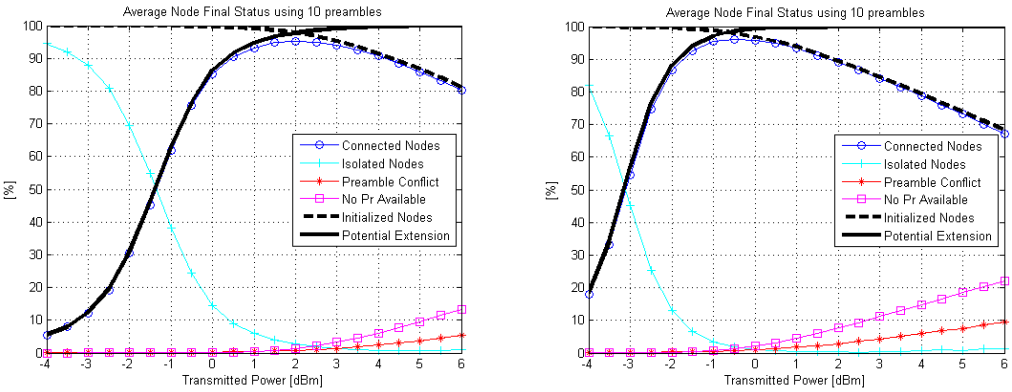
(A) Hexagonal pattern

(B) Square pattern

(C) Triangular pattern

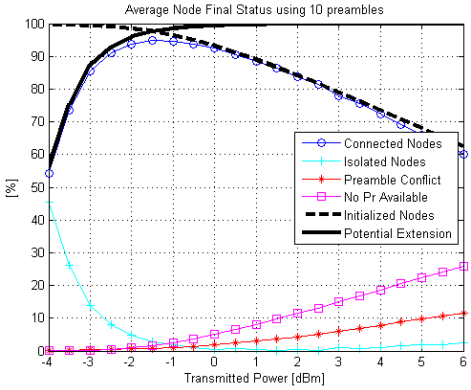
FIGURE 4.3. Nodes average final status after running the initialization method for different number of preamble options. It can be seen that, as the number of available preambles increases, the protocol performance always improves. A connectivity ratio of 90% is achieved with 10 preambles for the hexagonal pattern. Transmitting power was set to 3dBm. As it will be stated later on, performance can improve adjusting this parameter.

given topology which is density uniform, the transmitting power must be set carefully if a limited number of local addresses exist. Thus, for sparsely deployed networks, a higher transmitting power is required. For denser monitoring, a low transmitting power must be set so that the number of neighbours of each nodes remains confined. This fact is clearly shown in Figure 4.4, where the optimal transmission power for the hexagonal (sparse) pattern is $P_{\mathrm{t}} = 0$dB, and lower for the denser configurations. When the transmitted power



(A) Hexagonal pattern

(B) Square pattern



(C) Triangular pattern

FIGURE 4.4. As the transmitted power P_t increases, the connectivity of the network varies. A very low P_t will make nodes be very far apart. On the other hand, a very high transmission power will make nodes to have numerous neighbours, making it impossible for the protocol to correctly operate. Different network patterns require a different optimal transmission power.

is varied, the 90% connectivity threshold is surpassed for every configuration, even for the denser ones.

Regarding the scalability of the protocol, the initialization time was arbitrarily set to 10^3 time segments for these simulations, since it proved to be sufficient to ensure a connectivity ratio that meets the requirements. A denser network, however, could require a longer initialization time in order to maintain the connectivity ratio in the proposed values, because every node has more neighbors to collide with when initializing. Nevertheless,

if density remained constant, the connectivity ratio would stay approximately unchanging even if the number of nodes increased dramatically,given the distributed nature of the protocol. This idea makes the protocol highly scalable, thus meeting the requirement presented in Section 2.

This statement must be confirmed through simulations that were not performed in this thesis.

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this thesis we examined the problem of setting up an effective Medium Access Control protocol for channel dependent reverse-channel training MIMO wireless sensor networks. We proposed a method that considers the fact that no data can be sent before attaining channel state information at both transmitter and receiver. The method provides a functional MAC protocol that consistently works in a low SNR environment and is compatible with the scheme presented by Kettlun (2014). In this scheme, two opposed burst transmissions are essential to estimate the channel state at both transceivers. A MIMO diversity gain is achieved once both nodes can weigh the transmitted and received signals in their multiple antennas, respectively.

Medium access control was provided by means of the physical preamble used for packet detection. Different preambles were constructed, each of them having minimum peak sidelobe autocorrelation function. Additionally, every pair of preambles must have a bounded crossed-correlation value at every time. In this work, ten 32-symbol different preambles were constructed, each of them having an autocorrelation function with a peak sidelobe equal to 3, and a maximum crossed-correlation value of 10. These values proved to be sufficient to provide medium access control, since the preambles achieved a missed packet probability (p_{miss}) and a false alarm probability (p_{FA}) of at most 1% at SNR values of 2dB or higher. In addition, the receiver can adjust its reception threshold G so as to prioritize one probability over the other. For instance, a higher value of G provides a greater value of p_{miss}, but a lower value of p_{FA}, whereas a lower value of G provides a smaller value of p_{miss}, but a higher value of p_{FA}.

On top of these preambles, a random-based protocol was designed in which nodes access different logical channels to transmit to other nodes. As preambles are detectable only by nodes expecting to receive it and sheltered by the SVD beamforming channel access, nodes can simultaneously transmit data without interfering communications.

This protocol was tested through simulations, and proved to be successful in three different deployment scenarios. A connectivity ratio of more than 90% was achieved for each of these configurations. Connectivity ratio can be improved adjusting two key variables: the number of preambles N_A and the transmitted power P_t. A higher number of preambles will always improve the overall connectivity of the network, at the necessary cost of finding the required physical preambles. Increasing the transmitting power, however, does not always imply higher connectivity of the network. Given that there is a finite number of preambles, a very high transmission power leads to nodes having many neighbours. This is negative in any protocol with a limited number of addresses, since addresses must be unique in 2-hop neighbourhoods. On the other hand, a very low P_t makes nodes invisible to each other, decreasing the potential extension and therefore, the total connectivity of the network. Hence, an optimal P_t exists and depends on how dense the nodes were deployed. A denser network will necessarily require a lower transmission power than a sparser one.

5.2 Future Work

Even though the proposed protocol solves the multiple access problem inherent to the reverse-channel training scheme, the ultimate goal is to provide an energy efficient protocol. Since in wireless sensor networks one key issue is to minimize power consumption, an energy evaluation of the proposed solution may be performed. This analysis has to consider mainly the energy spent in idle listening, collisions, protocol overhead and overhearing, which conform 99% of the energy consumption in a MAC protocol (van Dam & Langendoen, 2003).

Additionally, further modelling of the network as a time-varying one would enhance the representation of the physical and MAC layers fusion. In time-varying networks, the neighbours of a node change over time. This will set a major challenge since a new definition of when two nodes are neighbours has to be settled. Probabilistic transmission approaches will further model the behaviour of a node in a network and its implications in the underlying MAC protocol.

Finally, an implementation of this algorithm using FPGA-based nodes is imperative in order to experimentally validate the operation of the algorithm. It would confirm the functioning of the presented work in experimental devices and further characterize the algorithm under real conditions.

Each of the above open questions provides sufficient material for a master's thesis.

References

Bao, L., & Garcia-Luna-Aceves, J. J. (2001). A new approach to channel access scheduling for ad-hoc networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking* (pp. 210–221).

Barker, R. (1953). Group synchronizing of binary digital systems. *Communication Theory*.

Bazan, O., & Jaseemuddin, M. (2012, Second). A survey on MAC protocols for wireless ad-hoc networks with beamforming antennas. *Communications Surveys Tutorials, IEEE*, *14*(2), 216-239.

Bharath, B., & Murthy, C. (2013, Jan). Channel training signal design for reciprocal multiple antenna systems with beamforming. *Vehicular Technology, IEEE Transactions on*, *62*(1), 140-151.

Bharghavan, V., Demers, A., Shenker, S., & Zhang, L. (1994). MACAW: a media access protocol for wireless LAN's. In *ACM SIGCOMM Computer Communication Review* (Vol. 24, pp. 212–225).

Chlamtac, I., & Farago, A. (1994, Feb). Making transmission schedules immune to topology changes in multi-hop packet radio networks. *Networking, IEEE/ACM Transactions on*, *2*(1), 23-29.

Cohen, M. N., Fox, M. R., & Baden, J. M. (1990). Minimum peak sidelobe pulse compression codes. In *Radar Conference, 1990., Record of the IEEE 1990 International* (pp. 633–638).

Ephremides, A., & Truong, T. (1990, Apr). Scheduling broadcasts in multihop radio networks. *Communications, IEEE Transactions on*, *38*(4), 456-460.

Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press.

Ingelrest, F., Barrenetxea, G., Schaefer, G., Vetterli, M., Couach, O., & Parlange, M. (2010, March). SensorScope: Application-specific sensor network for environmental monitoring. *ACM Trans. Sen. Netw.*.

Jedwab, J. (2008). What can be used instead of a Barker sequence? *Contemporary Mathematics*, *461*, 153–178.

Ju, J.-H., & Li, V.-K. (1998, Jun). An optimal topology-transparent scheduling method in multihop packet radio networks. *Networking, IEEE/ACM Transactions on*, *6*(3), 298-306.

Karn, P. (1990). MACA-a new channel access method for packet radio. In *Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Network Conference.*

Kettlun, F. (2014). *SVD-based beamforming communications over narrow-band quasi-static MIMO channels* (Unpublished master's thesis). Pontificia Universidad Católica de Chile.

Kumar, S., Raghavan, V. S., & Deng, J. (2006). Medium Access Control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks*, *4*(3), 326 - 358.

Leukhin, A., & Potekhin, E. (2013, Oct). Optimal peak sidelobe level sequences up to length 74. In *Microwave Conference (EuMC), 2013 European* (p. 1807-1810).

Li, X. (2008). *Wireless ad-hoc and sensor networks*. Cambridge University Press.

McGlynn, M. J., & Borbash, S. A. (2001). Birthday protocols for low energy deployment and flexible neighbor discovery in ad-hoc wireless networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing.*

Mow, W. (1993). Enumeration techniques for best N-phase codes. *Electronics Letters*, *29*(10), 907–908.

Muñoz, C., & Oberli, C. (2012). Energy-efficient estimation of a MIMO channel. *EURASIP Journal on Wireless Communications and Networking*, *2012*(1), 1–10.

Rajendran, V., Obraczka, K., & Garcia-Luna-Aceves, J. J. (2003). Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems* (pp. 181–192).

Rosas, F., & Oberli, C. (2013, April). Nakagami-m approximations for multiple-input multiple-output singular value decomposition transmissions. *Communications, IET*, *7*(6), 554-561.

Schurgers, C., Kulkarni, G., & Srivastava, M. (2002, Oct). Distributed on-demand address assignment in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, *13*(10), 1056-1065.

van Dam, T., & Langendoen, K. (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems* (pp. 171–180).

Venkataramani, R., & Marzetta, T. L. (2003). Reciprocal training and scheduling protocol for MIMO systems..

Ye, F., & Pan, R. (2009). A survey of addressing algorithms for wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on* (pp. 1–7).

Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the*

IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 3, p. 1567-1576 vol.3).

APPENDIX

APPENDIX A. AUTO- AND CROSS-CORRELATION OF SIGNALS

Consider a length L binary sequence $\mathbf{a}_0$ to be a vector which elements satisfy

$$\mathbf{a}_0[k] = \begin{cases} -1 \text{ or } 1 & \text{for } 1 \leq k < L \\ 0 & \text{otherwise} \end{cases} \tag{A.1}$$

We define the *aperiodic autocorrelation function* (ACF), $C_{00}[k]$ as

$$C_{00}[k] := \sum_{i=-\infty}^{\infty} \mathbf{a}_0[i+k]\mathbf{a}_0^*[i] \text{ for integer } k \tag{A.2}$$

which measures the extent to which a binary sequence resembles a shifted copy of itself (Jedwab, 2008). Note that the ACF meets the following symmetry condition

$$C_{00}[k] = C_{00}[-k] \tag{A.3}$$

The ACF function reaches it maximum when $k = 0$, and its value is

$$C_{00}[0] = \sum_{i=-\infty}^{\infty} |\mathbf{a}_0[i]|^2 = L \tag{A.4}$$

The peak sidelobe (PSL) of sequence $\mathbf{a}_0$ is defined as

$$\mathrm{PSL}(\mathbf{a}_0) = \max_{1 \leq k < L} |C_{00}[k]| \tag{A.5}$$

Now, for two sequences $\mathbf{a}_0$ and $\mathbf{a}_1$, their *aperiodic cross-correlation function* (CCF) $C_{01}[k]$ is defined as

$$C_{01}[k] := \sum_{i=-\infty}^{\infty} \mathbf{a}_0[i+k]\mathbf{a}_1^*[i] \text{ for integer } k \tag{A.6}$$

and measures the extent to which a binary sequence $\mathbf{a}_0$ resembles shifted copies of sequence $\mathbf{a}_1$. The CCF does not meet the symmetry condition (A.3). However, it does meet the antisymmetric condition

$$C_{01}[k] = C_{10}[-k] \tag{A.7}$$

The maximum value ξ_{01} of the cross-correlation between sequences $\mathbf{a}_0$ and $\mathbf{a}_1$ is

$$\xi_{01} := \max_{-L<k<L} |C_{01}[k]| \tag{A.8}$$

and due to Equation A.7, we have that $\xi_{01} = \xi_{10}$.

TABLE B.1. Preambles Set for $N_t = 1$.

#	Sequence
1	X1111001 11001010 10100110 11000000
2	X1101101 10000100 01110100 00101000
3	X1101011 01000110 00001011 10001000
4	X1010100 01100110 00011111 10110100
5	X1000000 01011100 01011001 10101100
6	X0110001 10110110 10101000 00011110
7	X0111100 11111110 00100101 01100100
8	X0011110 00010011 01110101 00001000
9	X1010011 10100100 01100101 11110111
10	X0101110 10010111 10111111 00110001

APPENDIX B. SET OF SELECTED PREAMBLES

Using the algorithm proposed in Section 3.3, two sets of preambles were found. The first one is suitable when $N_t = 1$, and the second one suits $N_t = 4$. Both have preamble length $L = 32$. Sets must be different according to the number of transmit antennas, because of the need of staggering a preamble when multiple transmitting antennas are used. Differential encoding is able to retrieve all symbols at the receiver except from the first one of each transmission. This is why an L–symbol preamble will have exactly N_t sections, each of them being transmitted through a different antenna. The first symbol of each of these transmissions will not be able to be retrieved at the receiver, becoming irrelevant which symbol is sent.

The sequences are shown in Tables B.1 and B.2. Irrelevant symbols are displayed with 'X's.

TABLE B.2. Preambles Set for $N_{\mathrm{t}} = 4$.

#	Sequence
1	X0011101 X0011010 X1101011 X0000000
2	X0010100 X0100010 X0110011 X1100000
3	X0010101 X1001011 X0011011 X1111000
4	X0010001 X1111110 X0111001 X1101001
5	X0001111 X0010000 X0001001 X0101011
6	X0011111 X0001100 X0100010 X0100100
7	X0011011 X0010010 X0001000 X1110101
8	X0010101 X0000001 X0011100 X0110110
9	X0010101 X1101101 X0110011 X0000111
10	X0001010 X0000110 X1100010 X0111111