

CHANNEL ASSIGNMENT IN MULTI-RADIO NETWORKS

by

Amalya Mihnea

A Dissertation Submitted to the Faculty of  
The College of Engineering and Computer Science  
in Partial Fulfillment of the Requirements for the Degree of  
Doctor of Philosophy

Florida Atlantic University

Boca Raton, FL

May 2015

Copyright 2015 by Amalya Mihnea

CHANNEL ASSIGNMENT IN MULTI-RADIO NETWORKS

by

Amalya Mihnea

This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Mihaela Cardei, Department of Computer & Electrical Engineering and Computer Science, and has been approved by the members of her supervisory committee. It was submitted to the faculty of the College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

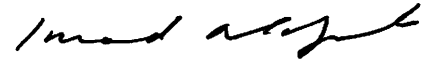
SUPERVISORY COMMITTEE:



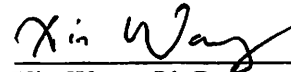
Mihaela Cardei, Ph.D.  
Dissertation Advisor



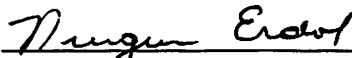
Ionut Cardei, Ph.D.



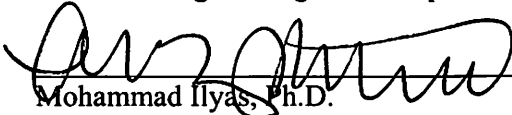
Imad Mahgoub, Ph.D.



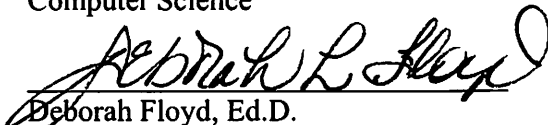
Xin Wang, Ph.D.



Nurgun Erdol, Ph.D.  
Chair, Department of Computer &  
Electrical Engineering and Computer Science



Mohammad Ilyas, Ph.D.  
Dean, College of Engineering and  
Computer Science



Deborah Floyd, Ed.D.  
Dean, Graduate College

4/7/15

Date

## ACKNOWLEDGMENTS

I would like to gratefully acknowledge the outstanding supervision of Dr. Mihaela Cardei, whose persistence, patience and support are greatly appreciated. I would also like to express my deepest gratitude to all members of my dissertation committee who offered useful suggestions and comments. Without their support this dissertation could not have been completed.

## ABSTRACT

Author: Amalya Mihnea  
Title: Channel Assignment in Multi-Radio Networks  
Institution: Florida Atlantic University  
Dissertation Advisor: Dr. Mihaela Cardei  
Degree: Doctor of Philosophy  
Year: 2015

Channel assignment in multi-radio networks is a topic of great importance because the use of multiple channels and multiple radios reduces interference and increases the network throughput. The goal of our research is to design algorithms that maximize the use of available resources while providing robustness to primary users that could reclaim one or more channels. Our algorithms could be used in ad hoc networks, mesh networks, and sensor networks where nodes are equipped with multiple radios. We design algorithms for channel assignment which provide robustness to primary users without assuming an accurate primary user behavior model. We also compute bounds for capacity in grid networks and discuss how the capacity of a network changes when multiple channels are available. Since preserving energy is very important in wireless networks, we focus on algorithms that do not require powerful resources and which use a reduced number of messages.

## CHANNEL ASSIGNMENT IN MULTI-RADIO NETWORKS

LIST OF TABLES .....	viii
LIST OF FIGURES .....	ix
1. INTRODUCTION .....	1
1.1 Motivation .....	1
1.2 Problem Definition .....	3
1.3 Contribution .....	4
2. MULTI-CHANNEL PROTOCOLS .....	5
2.1 Challenges and Classifications of Multi-Channel Protocols .....	5
2.2 Capacity, Interference and Topology Control in Wireless Sensor Networks ....	13
2.3 Power and Traffic Aware Protocols .....	20
2.4 Radio and Multi-Radio Issues .....	31
2.5 Conclusions .....	38
3. TWO CHANNEL ASSIGNMENT ALGORITHMS FOR MULTI-RADIO MULTI-CHANNEL NETWORKS .....	40
3.1 Introduction .....	40
3.2 Problem Definition .....	44
3.3 Solutions for the CA-RTC Problem .....	45
3.4 Simulations .....	54
3.5 Conclusions .....	60

4. ROBUSTNESS TO PRIMARY USERS THAT RECLAIM MULTIPLE CHANNELS AT ONCE .....	61
4.1 Introduction and Problem Definition .....	61
4.2 Channel Assignment .....	62
4.3 Theoretical Analysis.....	69
4.4 Finding the Distribution of the Assigned Channels .....	74
4.5 Algorithm for Channel Assignment in a Grid.....	81
4.6 Analysis of Interference .....	84
4.7 Another Version for Channel Assignment Algorithm in a Grid .....	89
4.8 Conclusions .....	90
5. EFFICIENT WIRELESS COMMUNICATION IN GRID NETWORKS.....	92
5.1 Introduction and Problem Definition .....	92
5.2 Capacity for Unidirectional Communication .....	92
5.3 Capacity for Bidirectional Communication .....	104
5.4 Formula for the Number of Time Units .....	108
5.5 Conclusions .....	112
6. CONCLUSIONS .....	113
BIBLIOGRAPHY.....	116

## LIST OF TABLES

Table 3.1: Simulation parameters .....	56
Table 5.1: Values of $n_{BC}$ and $n_{BL}$ for some values of $r/d$ .....	96
Table 5.2: Bounds for capacity when $s = 100$ .....	103
Table 5.3: Bounds for capacity when $s = 1000$ .....	103
Table 5.4: Bounds for capacity when $s = 10000$ .....	103



## LIST OF FIGURES

Figure 2.1: The multi-channel hidden terminal problem .....	7
Figure 3.1: Grid-based Channel Assignment.....	47
Figure 3.2: Channel assignment for cycles of length 4.....	51
Figure 3.3: Example using the Distributed Channel Assignment Algorithm .....	53
Figure 3.4: Distributed-CA - Comparisons between multi-radio and single-radio WSNs (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio. ....	59
Figure 3.5: Comparisons between Distributed-CA (alg. 2) and Grid-CA (alg. 1). Multi-radio versus single-radio WSNs when $p = 100\%$ . (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio.....	59
Figure 3.6: Comparisons between Distributed-CA (alg. 2) and Grid-CA (alg. 1). Multi-radio versus single-radio WSNs when $p = 30\%$ . (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio. ....	59
Figure 3.7: Comparisons between Distributed-CA (alg. 2) and Grid-CA (alg. 1). The impact of PU on WSN performance. (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio. ....	60
Figure 4.1: Monitored area divided into grids and cells' representatives (red).....	62
Figure 4.2: Channel assignment for $C=8$ , $Q=5$ , $k=3$ and repeating blocks in a large network .....	64
Figure 4.3: Construction of the first row/column of a grid for $C=5$ , $Q=3$ .....	65

Figure 4.4: Tables with values of $j$ and $n_{\text{cycles}}$ for specific values of $C$ , $Q$ and $k$ .....	68
Figure 4.5: Construction of the first row/column of a grid for $Q=3$ , $k=1$ .....	69
Figure 4.6: Patterns of skipped elements for $C$ a multiple of $Q-k$ and $n_{\text{cycles}}=1$ .....	70
Figure 4.7: Patterns of skipped elements for $C=9$ , $Q=8$ , $k=2$ .....	71
Figure 4.8: Blocks for $Q-2k=0$ .....	72
Figure 4.9: Blocks for $Q-2k<0$ .....	73
Figure 4.10: Blocks for $Q-2k<0$ and $Q-k$ divides $Q$ .....	73
Figure 4.11: Blocks for $Q-2k<0$ and $Q-k$ does not divide $Q$ .....	74
Figure 4.12: Configuration for $C=10$ , $Q=9$ , $k=7$ .....	75
Figure 4.13: Table P for $C=10$ , $Q=9$ , $k=7$ .....	76
Figure 4.14: Table P for $C=10$ , $Q=7$ , $k=5$ .....	76
Figure 4.15: Constructing tables P .....	77
Figure 4.16: The structure of an array constructed by concatenating the rows of table P .....	78
Figure 4.17: Patterns of assigned channels for $k=2$ .....	83
Figure 4.18: Sensors within interference range of node $e$ .....	85
Figure 4.19: Horizontal edges within interference range of edge $e$ .....	86
Figure 4.20: Table P for $Q-2k<0$ ; $C=10$ , $Q=6$ , $k=2$ , $j=5$ .....	86
Figure 4.21: Tables with values of $p(e)$ and $p'(e)$ for specific values of $C$ , $Q$ and $k$ .....	88
Figure 5.1: Top border of a grid (unidirectional communication) .....	94
Figure 5.2: Non-interfering horizontal active edges .....	94
Figure 5.3: Non-interfering active edges for four cases .....	97
Figure 5.4: Intercalating active edges for a new channel .....	99

Figure 5.5: Top border of a grid (bidirectional communication).....	106
Figure 5.6: Non-interfering horizontal active edges .....	106
Figure 5.7: Time units for unidirectional communication, one radio and one channel .....	107
Figure 5.8: Non-interfering active edges for one channel .....	109

# 1. INTRODUCTION

## 1.1 Motivation

Due to the recent growth of wireless applications, the communication on the unlicensed spectrum (e.g. ISM) has become congested, while the utilization of the licensed spectrum varies between 15% and 85% temporally and geographically [1]. Cognitive radio networks (CRNs) constitute a promising solution used to address the issue of inefficient spectrum usage.

A cognitive radio uses a software defined radio technology. The communication functions are implemented on software instead of hardware. The radio senses the environment, tracks changes, and reacts based upon its findings.

Cognitive radios are designed to operate on a wide spectrum range and can switch to a different frequency band with limited delay. This technology allows primary users (PUs) to share the spectrum with secondary users (SUs), where SUs communicate through un-assigned spectrum bands without disrupting the regular usage of the PUs. CRNs allow SUs to take advantage of unoccupied spectrum in an opportunistic manner using dynamic spectrum access strategies.

To avoid interference with a PU, a SU must vacate the spectrum when the channel is being used by a PU. This affects ongoing communication of the SUs. The challenge occurs due to the difficulty to predict when a PU will appear in a given spectrum.

To use other channels, SUs have to spend a considerable amount of time for spectrum sensing and channel switching [2]. In addition, a change in an SU channel may trigger other nodes to change their channels in a ripple effect in order to maintain the desirable topology.

In the presence of PUs, the robustness constraint requires that if a channel is reclaimed by a PU, then the resulting SU topology still preserves the connectivity between any two nodes. The PUs can affect part of the network or the entire network (e.g. transmission of the TV tower). If two sensors  $u$  and  $v$  communicate on a channel that is reclaimed by a PU, then the packet is re-routed from  $u$  to  $v$  through another channel of  $u$  and possibly another radio of  $v$ . Thus packet dropping and significant delay can be avoided.

Wireless sensor networks (WSNs) constitute the foundation of a broad range of applications related to national security, surveillance, military, health care, and environmental monitoring.

Many results and channel assignment schemes proposed for wireless ad hoc networks and mesh networks cannot be directly applied to sensor networks, which have different characteristics such as smaller packet size, less powerful radios, or fewer radios. There are also differences related to energy source, power, computational capacity, and memory. The main type of communication used by WSNs for data gathering is *convergecast*, where data travels from many nodes (e.g. sensor nodes) to a single node called sink or base station (BS).

With a single radio and a single channel, WSNs cannot provide reliable and timely communication in case of high data rate requirements because of radio collisions and

limited bandwidth. Therefore, designing multi-channel based communication protocols is essential for improving the network throughput and providing quality communication services.

Multi-channel protocols consist of two major components: channel assignment (CA) and medium access control (MAC). In some protocols these two are combined: channels are selected at every access to the medium. A good channel assignment is one that reduces interference among concurrent transmissions, maximizes the capacity of the network, mitigates packet congestion within a single channel, and preserves robustness to the presence of a PU.

## **1.2 Problem Definition**

Energy efficiency becomes more and more important as the creation of smaller devices is pursued. These smaller devices usually run on batteries and are limited in terms of energy consumption and computational capabilities. Complicated algorithms that work for general networks might not work well if the networks are comprised of such devices. Some existing algorithms require multiple negotiations between nodes and may require cascaded switching of multiple users.

There is a lack of research which focuses on simplified and distributed algorithms that assign channels to radios while taking into account the presence of PUs especially PUs that could reclaim multiple channels simultaneously. The main goal of our research is to find channel assignments which assign to each node's radio a channel such that the resulting topology is connected, robust to PUs, and has a reduced interference. The second goal is to find results that would help design such algorithms.

### 1.3 Contribution

In the presence of PUs, the network has to maintain connectivity. Also, if devices operate on batteries, the lifetime has to be maximized. Our channel assignment algorithms address these problems and do not assume an accurate primary user behavior model, which is hard to achieve. We focused on devices with limited computational capabilities and on algorithms that require a reduced number of messages.

This dissertation contains four chapters. Chapter 2 gives an overview of multi-channel issues in wireless networks and presents a literature review. Chapter 3 introduces two channel assignment algorithms and a comparison between them using the Ns-3 simulator. These algorithms are robust to PUs that reclaim one channel at a time.

Chapter 4 focuses on theoretical aspects related to a more general algorithm that provides robustness to PUs that reclaim multiple channels at once. Our algorithm performs certain calculations and a node assigns its channels based on its position in the network. A theoretical analysis helped us to obtain results on how to balance the distribution of channels in the network when this algorithm is used.

In Chapter 5 we compute bounds on capacity in grid networks that use one channel and discuss how this capacity changes when multiple channels are available. We also present a scheduling method for a general communication pattern. A formula for the number of time units is given, such that the communication between any two horizontally or vertically adjacent nodes takes place in all directions, i.e. all the edges are covered, without interference.

This research is relevant because it could be applied to different kinds of networks, including those with limited resources like WSNs.

## 2. MULTI-CHANNEL PROTOCOLS

### 2.1 Challenges and Classifications of Multi-Channel Protocols

Medium access control (MAC) methods are designed for coordinating communication: sharing the wireless medium and alleviating conflicts. They are contention based such as carrier sense multiple access (CSMA), or schedule based such as time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA). Some other approaches use power control and directional antennas to further reduce interference [3].

Channel assignment schemes can be static, semi-dynamic, or dynamic. The static ones assign channels for permanent use, at deployment time or during runtime. Even if the assignments can be renewed, radios do not change their frequencies during communication. Semi-dynamic schemes assign constant channels to radios but these can change during runtime while in the dynamic approaches there is no initial assignment of channels to radios and the channels can change between successive data transmissions.

Some dynamic schemes use a dedicated control channel which can only be used for exchanging control messages (negotiations of the channels for data transmission) while data is exchanged using data channels. These approaches do not need time synchronization and are easier to implement.

Another classification can be related to implementation/execution. If the channel assignment is done by a central scheduler, then the implementation is centralized.



Otherwise, if nodes negotiate and more nodes are involved in assigning channels, then the implementation is distributed. The communication between devices involved in the distributed protocols is done by exchanging messages. Centralized approaches have limitations such as lack of a global common control channel to support centralized control and poor scalability due to the difficulty of capturing consistent global information in a dynamic environment [4].

If switching channels is allowed, some problems that reduce network performance are: multi-channel hidden terminal problem, deafness problem, and broadcast support [5].

The multi-channel hidden terminal problem (Figure 2.1) can occur in carrier-sense multiple access with collision avoidance (CSMA/CA)-based protocols when the control packets (RTS/CTS) sent on a specific channel are not received by the nodes communicating on other channels. For example, suppose that we have four nodes A, B, C, and D that use a common control channel, let us say channel 1. Assume that A and B have successfully established a communication on channel 2. Assume that the node C was busy receiving on another channel when B sent the CTS to A, thus C is not aware of A communicating with B on channel 2. If subsequently C initiates a communication with D on channel 2, this will cause collisions at the node B. The cause of this problem is that nodes may listen to different channels when other nodes exchange RTS/CTS control messages on the common control channel [6].

In order to communicate, a sender and a receiver have to be on the same channel, otherwise the deafness problem occurs in multi-channel communication. Suppose that a transmitter sends a control packet to initiate communication and that the receiver is tuned

to another channel. If the sender does not get any response after sending multiple requests, then it may conclude that the receiver is not reachable anymore [5].

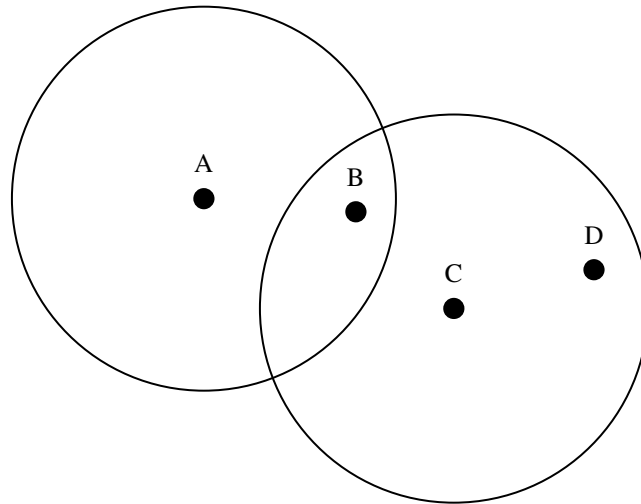


Figure 2.1: The multi-channel hidden terminal problem

Broadcast support refers to the difficulty of supporting successful broadcasts when the nodes change their channels frequently. Usually protocols use a broadcast channel to support broadcasts. If nodes operate on multiple channels, channel switching introduces delays and computational overhead.

Some additional problems that are mentioned in the literature are idle listening when nobody is sending, and overhearing of messages, when a node receives messages that are destined for other nodes. It is known that the energy spent for receiving a message can even be a bit higher than the energy spent for transmitting a message [3]. All these problems are connected to power consumption and they could shorten the lifetime of the network. They have to be taken into account when designing power-aware algorithms. Later, we will present some power- and traffic-aware protocols.

In [5] CA methods are classified using criteria such as: assignment method, control channel, implementation/execution, synchronization, medium access, broadcast support, channel model, interference model, and objective. We elaborate on the classification into fixed (static), semi-dynamic, and dynamic methods and we give a few examples for each category. The algorithms are assumed to be single radio unless otherwise specified.

### *Fixed Channel Assignment*

An idea related to this category is to take advantage of *clustering* the nodes such that all the nodes in a cluster use a unique frequency that is different than the frequencies of other clusters. The goal is to avoid or to minimize interference.

Another idea is to use *component-based channel assignment* [7], which assigns the same channel to all nodes belonging to a component formed by nodes belonging to mutually intersecting flows. Two flows are said to be intersecting, if there is a common node in the set of active nodes for each flow, which serves both flows. If flow  $f_1$  intersects with flow  $f_2$ , and flow  $f_2$  intersects with flow  $f_3$ , then all nodes on the paths traversed by these three flows are assigned the same channel.

Different connected components can potentially operate on different channels. A connected component in a flow graph is defined as the largest subgraph, such that there exists a path from any node in the subgraph to all other nodes in the subgraph. The authors of [7] show through theoretical and quantitative analysis that this simple strategy can improve the network performance. They also propose centralized and distributed routing layer algorithms that implement this strategy effectively.

The main advantage of fixed channel assignment is its simplicity since nodes maintain their assignments. But there are also disadvantages, such as not being adaptive to changes in the network topology due to traffic changes or unstable links, and no possibility of communication between two nodes that have different channels. These issues, which could lead to poor performance and network partitions, could be solved by renewing channel assignments from time to time.

The multi-hop scenario used in WSNs assumes that data travels from source nodes through intermediate nodes towards one or several BSs. Therefore the routing topology is a tree or a forest.

In [8] a Tree-based Multi-Channel Protocol (TMCP) for data collection applications in WSNs is proposed. The authors assume that there is a single BS equipped with multiple radio transceivers, each of which works on different channels. The network is partitioned into multiple vertex-disjoint subtrees all rooted at the BS. Each tree is allocated a different channel and a data flow is forwarded only along its corresponding subtree. TMCP is distributed and it works with a small number of channels, without any time synchronization requirement.

TMCP has three components: Channel Detection (CD), Channel Assignment (CA), and Data Communication (DC). Given  $k$  orthogonal channels, the CA module partitions the whole network into  $k$  subtrees and one unique channel is assigned to each subtree. We do not have inter-tree interference so the goal of partitioning is to divide the network into subtrees with low intra-tree interference. The DC component manages the data collection through each subtree.

TMCP uses the *protocol model* (a graph-based interference model) to estimate interference: Two nodes interfere with each other if the distance between them is smaller than a threshold value. The size of a node's interference set is used in subsequent subtree partition. However, the distance-based interference model does not hold in practice as shown by recent empirical studies.

In [9] TMCP has been extended to employ interchannel RSS (Received Signal Strength) models for interference assessment in channel allocation. A novel algorithm is proposed, which can significantly reduce the overhead of multi-channel interference measurement by exploiting the spectral power density (SPD) of the transmitter.

In [4] three algorithms are presented: node-based, link-based, and node-link-based. The best of them, the node-link-based distributed algorithm, partitions the network into "stars", which resemble 2-level trees, and uses maximal matching between channels and adjacent links by the Hungarian algorithm. Channels are assigned to links that minimize channel-conflict probability by computing channel weights based on the conflict probability of every available channel on each link.

In dense networks this algorithm could be used for the backbone consisting of cluster heads with longer transmission range while the communication within clusters could be done by using CSMA. Cluster heads could solve inter-cluster interference by assigning inner-cluster communicating channels.

In [10] network robustness and channel interference are jointly considered when developing centralized and distributed algorithms. Backup channels are used to avoid network partition but the requirement to adjust channels for previously assigned links might be unsuitable for WSNs, which have limited resources. The proposed solutions

outperform existing interference-aware approaches when PUs appear and achieve similar performance at other times. The algorithms from these last two papers are not specifically designed for WSNs.

### *Semi-Dynamic Channel Assignment*

Methods in this category, which appear to be the most popular ones, assign fixed channels either to senders or receivers but the assignments can change during communication. Graph coloring algorithms are useful in such approaches.

Network partitioning could be eliminated by using a coordinated channel switching between senders and receivers, which need to be on the same channel at the same time in order to communicate.

In [11] a distributed Game Based Channel Assignment algorithm (GBCA) is proposed to solve the problem of multi-channel assignment in WSNs. Unlike previous static assignment protocols, this algorithm takes into account both the network topology information and the transmission routing information. Simulations show that GBCA achieves better network performance than MMSN in terms of delivery ratio, throughput, packet transfer delay, and energy consumption.

The MMSN protocol [12] is the first multi-frequency MAC protocol designed specifically for WSNs and it consists of two aspects: frequency assignment and media access. Frequency assignment allows users to choose one of four available frequency assignment strategies to evenly assign different channels among 2-hop neighbors. In media access design, potential conflicts are solved by accessing the shared physical frequencies in a distributed way. Both GBCA and MMSN are distributed approaches.

### *Dynamic Channel Assignment*

In these approaches, mostly distributed, a channel selection takes place before every data transmission. The channel selection can be measurement based or status based. The first category is related to communicating parties measuring signal-to-interference noise ratio (SINR) and the second one to the status of the channels: idle (available) or busy.

When the traffic is light, many multi-channel MAC protocols for WSNs are less energy-efficient than single channel MAC protocols. In contrast to these, Y-MAC [13] is energy-efficient and maintains high performance under high-traffic conditions. It is also the first protocol that uses dynamic channel assignment in WSNs.

In this TDMA-based multi-channel MAC protocol, a send time slot is used for data transmission and a receive time slot for data reception. An exclusive send time slot in 2-hop neighborhood guarantees collision-free access to the medium, which reduces the energy wasted by contention and collisions. However, energy is wasted due to overhearing and idle listening, since all nodes have to wake at every time slot to avoid missing incoming messages.

Initially, a base channel is used to exchange messages. Sensor nodes hop to the next radio channel if they have additional pending messages for the receiver (bursty traffic). Y-MAC improves the performance of the network (increased throughput, reduced message delivery latency) under high traffic conditions and uses multiple channels with low energy consumption. Other dynamic protocols such as MAC and MuChMAC are presented in [5].

## 2.2 Capacity, Interference and Topology Control in Wireless Sensor Networks

A special type of WSNs are Wireless Multimedia Sensor Networks (WMSNs), which enable advanced surveillance, traffic monitoring, and healthcare systems and require larger bandwidth. A challenge of WMSNs is an increased bandwidth demand in the presence of higher levels of interference. Using multiple channels for parallel transmissions could improve network capacity.

The most common communication types in WSNs are broadcast and data collection. The main goal of broadcasting is to send a message to all the nodes in the network and the goal of data collection is to send data messages from source nodes to the sink(s). Broadcast could be used as an initial step in data collection to determine shortest paths between nodes and the sink(s). Communication in WSNs has to take several factors into account, such as link length, number of hops to the BS, and node degree [14].

For a given network topology, different routing trees or channel assignment mechanisms have impact on the maximum achievable network throughput. With multiple channels there is a need for channel coordination: The sender and the receiver have to transmit and to listen on the same channel at the same time.

In the *receiver-based channel allocation*, a fixed channel is assigned to each sensor node and that channel is used to receive messages. A neighbor that wants to send a message to this node should use the receiver's channel to send. In this allocation, the nodes that do not receive any message are not assigned any channel.

In the *link-based channel allocation*, every link or edge is assigned a channel and every transmission along that link uses that channel. A difference between this and the



receiver-based channel allocation is that here, for the same receiver, various senders can use different channels, resulting in less interference.

In data gathering WSNs where source and sink nodes are all equipped with half-duplex transceivers, the maximum throughput per node is  $W/n$ , where  $n$  is the number of source nodes and  $W$  is the transmission capacity. According to [3], the maximum throughput can be reached only if the sink is 100% busy receiving packets and if the schedules of all nodes are aligned for interference-free communication for the given network topology.

In [15] it was proved that minimizing the schedule length for an arbitrary network in the presence of multiple frequencies is NP-hard. Also, finding the minimum number of frequencies that are necessary to remove all the interfering links in an arbitrary network is NP-hard.

The authors use an *aggregated convergecast model* [16] where each node has the ability to aggregate all the packets from its children as well as its own data into a single packet before transmitting it to its parent. The routing structure used in data collection is a tree rooted at the sink and the frequency assignment strategy is receiver-based.

Each node has a single, half-duplex transceiver, so it can either transmit or receive a single packet at any given time slot. The radio cannot receive multiple packets simultaneously so assigning different frequencies to the transmitters that are children of the same parent does not help in reducing the schedule length. A contention-free multiple access protocol such as TDMA is used and each node generates only one packet at the beginning of every frame.

A graph-based interference model is used, where the interference range of a node equals its transmission range. Two types of interference for concurrent transmissions on two edges are considered: *primary interference*, if the two edges are adjacent, and *secondary interference*, if the receivers of both edges are on the same frequency and at least one of the receivers is within the communication range of the other transmitter.

The authors give an upper bound on the maximum number of frequencies required to remove all the secondary interfering links and also propose a polynomial time algorithm that minimizes the schedule length under this scenario. A secondary interfering link is removed if the two receivers of an edge pair are assigned different frequencies. Because half-duplex radios are used, the primary interference cannot be removed using multiple frequencies.

A closely related work [16] describes a realistic simulation-based study on a tree-based data collection utilizing transmission power control, multiple frequencies, and efficient routing topologies. It was shown that the data collection rate becomes limited by the maximum degree of the tree once all the interfering links are removed by assigning multiple frequencies. This rate can be further increased on degree-constrained trees.

In [17], a multi-path scheduling algorithm for the snapshot data collection in single radio multi-channel WSNs is proposed. A tighter lower bound for its achievable network capacity is given compared to the results in [18]. Also, a novel continuous data collection method for dual-radio multi-channel WSNs is shown to speed up the data collection process and improve the network capacity. Most of the previous works related to network capacity consider just single-radio single-channel WSNs. The protocol used in [18] is the

*protocol interference model* but the results can be extended to WSNs under the *physical interference model* [19].

The protocol interference model assumes that all nodes have the same interference range  $R$ . If a node  $X_i$  sends data to a node  $X_j$  over a channel, the transmission is successful if the destination node is far enough from the source of any other simultaneous transmission on the same channel or  $|X_k - X_j| \geq (1 + \Delta) |X_i - X_j|$ , where  $\Delta > 0$ , for any node  $X_k$  transmitting over that channel. This model can take advantage of the graph-coloring based scheduling algorithms.

The physical interference model (SINR model) is considered better because it can capture the interference from multiple simultaneous senders. If  $\{X_k; k \in T\}$  is the subset of nodes transmitting simultaneously at some point in time over a certain channel, and  $P_k$  is the power level chosen by the node  $X_k$  for  $k \in T$ , then the transmission from a node  $X_i, i \in T$ , is successfully received by a node  $X_j$  if

$$\frac{\frac{P_i}{|X_i - X_j|^\alpha}}{N + \sum_{k \in T, k \neq i} \frac{P_k}{|X_k - X_j|^\alpha}} \geq \beta$$

where  $\beta$  is the minimum SINR for successful receptions,  $\alpha$  represents the exponent for signal loss due to distance, and  $N$  is the level of the ambient noise [5].

In [20], the authors analyze the capacity limits of multi-hop paths in a WSN with multiple channels. Also, a control channel based MAC protocol is implemented and analyzed using IEEE 802.15.4 based networks with 16 orthogonal channels. This protocol is based on the split phase approach described later, which does not require any

time synchronization and is simple to implement, but could suffer from saturation of the control channel.

Previous MAC protocols such as McMAC, CMAC, and MAC are not considered efficient. The channel coordination mechanisms are divided into four categories: (1) Dedicated Control Channel, (2) Common Hopping, (3) Split Phase, and (4) McMAC. The differences of these mechanisms are related to the number of radios they use: a single radio or two radios.

The dedicated control channel mechanism is used with multiple radios. There is a specified control channel used for one radio to transmit information related to channel selection. After a channel is selected, data are transferred through that channel between the sender and the receiver.

The common hopping mechanism is used with a single radio. There is a common pattern of shifting channels followed by all nodes. An RTS/CTS handshake takes place using the current common channel when two nodes want to communicate. During this handshake, the given sender-receiver pair stops hopping and stays tuned to this common channel. After data transfer is done, the two nodes resume the same hopping sequence.

The split-phase approach is used with a single radio and has two phases: control and data. The control phase includes agreements that are made between sender(s) and receiver(s) regarding the channel on which data have to be transferred. In the data phase, the data are transmitted using the chosen channel.

The fourth category includes the McMAC [21] protocol, which uses random hopping of channels for each node. When a node A has to transfer data to a node B, if data to be

transferred are large, then A follows B's hopping pattern. After data transfer, A resumes its own hopping sequence.

In addition to MAC protocols, other methods to alleviate interference are: transmission power control (transmitting signals with sufficient power instead of maximum power) and use of directional antennas instead of omnidirectional antennas.

Topology control is a very important technique in WSNs which deals with sensor nodes' power control and network structure. Some of the design goals of topology control are: minimum energy consumption, low interference, small node degree, connectivity, and planarity.

Some topology control algorithms that have been proposed in the last few years are inappropriate because they do not address both communication types in WSNs: message dissemination and data collection. Very often the robustness of the topology is neglected. The node failure is handled by most of the algorithms by simply resetting the whole network, which has a high cost in terms of energy consumption. Other algorithms try to establish several disjoint routes between sensor nodes and the BS or between sensor nodes in order to improve the network robustness, which is not an easy task.

The simplest topology control strategy is called Unit Disk Graph (UDG), in which all sensor nodes communicate with each other using their maximum power so that all possible communication links are preserved. Other approaches try to eliminate all or some of the redundant links, keeping some links to improve the network's tolerance to node failure and network capacity, or try to improve the robustness of the network by a specially designed edge weight function, which contains link length and number of hops to the BS.

Other topology control algorithms partition the network into several disjoint parts or clusters, each of which has one selected cluster head and several cluster members. The cluster head is responsible for inter-cluster communication, and all cluster members only communicate with their own cluster head in a TDMA manner. Each cluster head is elected periodically in order to balance energy consumption.

In [14] the authors propose a novel tree-based topology control algorithm which contains a Fast Dissemination Tree (FDT) for message broadcast and a Balanced Data Collection Tree (BDCT) for data collection. This algorithm has better performance than the existing ones and helps balance energy consumption between nodes.

FDT uses the maximum power and chooses the nearest neighbor for transmission. Node  $i$  chooses as its parent the nearest node  $j$  which has the smallest number of hops to the BS, so that the message can be received and relayed as quickly as possible. The BS is placed in the center of the network and each node has information about its neighbors, such as ID, number of hops to the BS, energy left and location, which are used in the construction of the BDCT.

The objective of the BDCT is to achieve a balance among different design goals: link length, number of hops to the BS, remaining energy, and robustness. The selection of a parent node is based on the link weight whose settings give priority to a node with more residual energy. This helps balance the energy consumption throughout the whole network, which in turn prolongs the lifetime of the network.

Robustness is related to the number of critical nodes. Node  $i$  is said to be a critical node if, once  $i$  fails, the network is no longer connected. A tree based topology has a severe drawback: once a parent node fails, all its children lose connection with the BS. A

solution to this case is topology reconstruction but this approach is expensive in terms of energy and time. Another solution is to establish multiple paths between a node and the BS, but computing multiple paths increases the computational complexity. A better solution proposed in [14] is to choose a network topology that resembles a spider web which has to be reinitialized just in the case of multiple node failures.

### **2.3 Power and Traffic Aware Protocols**

Many WSNs applications need data to be transmitted in a timely manner, such as WSN-based disaster warning systems or a warning surveillance system that has to notify authorities when intruders are detected. It is known that wireless links are lossy and retransmissions increase the end-to-end delay. A solution to improve link quality is to increase transmission power but this may increase interferences and channel contention, which leads to a decreased network capacity. An alternative solution is to use multi-channel protocols to increase network capacity and to reduce interference and delays.

In some MAC protocols the energy efficiency is improved by reducing idle listening. Nodes sleep and wake up when they receive a message. Other protocols use a distributed time slot selection mechanism for scheduling senders or receivers. In LMAC [22] each sender owns an exclusive time slot in a 2-hop neighborhood and all nodes wake up at every time slot to avoid missing incoming messages. Receivers are scheduled in Crankshaft [23] as follows. Each node wakes up for data reception at a different offset from the start of the super frame in order to reduce the number of nodes overhearing unrelated messages.

Energy consumed to receive a message is considered to be greater than that to transmit a packet due to the sophisticated de-spreading and error correction techniques. Therefore scheduling receivers is more energy-efficient than scheduling senders if the traffic is light, because each node samples the medium only in its own receive time slot [13].

A multi-channel protocol based on LMAC is presented in [24]. First nodes communicate on a basic channel and when all the time slots are exhausted new channels are used. Two nodes that are on different channels communicate through bridge nodes, even if they are within one hop from each other. This can increase latency and energy consumption by bridge nodes.

In [25] the network is clustered and each cluster head collects request messages from the cluster members, and then assigns channels to both source and destination nodes. One disadvantage of this method is that cluster head nodes consume more energy than the other nodes. Also, the maximum network throughput of each cluster is limited by the number of packets that the cluster head can manage.

Some MAC protocols that focus on the efficient use of energy do not address bursty traffic, which became a major issue due to the ability of the latest operating systems for WSNs to run multiple applications. Newer MAC protocols combine the advantages of contention-based protocols and time-slotted protocols.

The multi-radio MAC protocol presented in [26], which uses two radio transceivers on each sensor node, separates the transmission of control packets and data packets into low- and high-frequency bands, respectively. The information necessary for selecting high-frequency channels is exchanged in micro preamble frames using the low-frequency



band. Data transmission will be then carried out using the selected channel. The high-frequency band radio is turned on only for a very short time for burst data transmissions. The control channel is also used for small data packets. The multi-radio protocols are not considered an economical solution in terms of energy and most approaches focus on devising a multi-channel MAC protocol using a single radio transceiver.

In [27] a Multi-Channel Real-Time (MCRT) communication protocol that uses both multiple channels and transmission power adaptation for real-time WSNs is presented. MCRT uses a small number of orthogonal channels that are allocated to network partitions formed based on many-to-one data flows in a WSN. The goal is to minimize the channel contention among different flows.

This algorithm outperforms the one presented in [28], called the Real-time Power-Aware Routing (RPAR) protocol, which dynamically adapts its transmission power and routing decisions based on packet deadlines. Practical issues in WSNs, such as lossy links, scalability, and severe memory and bandwidth constraints, are also addressed by this algorithm. A novel neighborhood manager dynamically discovers eligible forwarding choices and manages the neighborhood table. New forwarding choices are discovered by adapting the transmission power to a neighbor that is already in the neighbor table (*power adaptation*) or by discovering new neighbors (*neighbor discovery*).

This paper handles smaller holes through power control and larger holes through face routing mechanisms. Holes may appear in the network topology due to voids in node deployment or node failures.

A protocol that deals with power management issues has to minimize the energy consumed for packet transmission and reduce the energy wasted on idle listening. The

Real-time Power-Aware Routing (RPAR) protocol is discussed in connection with several power-management techniques.

In [3], reinforcement learning is used to make sensor nodes learn how to achieve successful transmissions and receptions while minimizing energy consumption. This distributed algorithm reduces the energy wasted by collisions, idle listening, and deafness problem. The protocol is based on a combination of TDMA and FDMA techniques.

Time is divided into frames of fixed length, composed of a number of time slots. The length of a slot allows the transmission of a single data message and an acknowledgment message. A channel-hopping scheme is used instead of a fixed frequency assignment for each node.

In each frame, a node periodically switches its channel at each time slot according to a chosen channel-hopping pattern, called the *default sequence*, obtained with a pseudorandom number generator. When establishing communication, the sender tunes its radio to the current channel of the receiver and transmits data.

When it knows the receiver address, which is the seed of its generated default sequence, the sender can reproduce the listening channel of the receiver. With this method, parallel transmissions between several pairs of nodes can take place without exchanging information or negotiating a communication channel, which represent a major challenge for energy and bandwidth constrained WSNs.

Each node in the network has multiple parents that it can use to forward data to the sink and it decides which one to use based on which channel the parent is listening to. A node can choose from the following pool of actions: stay on its home channel and listen, deviate to a chosen channel to transmit in a certain time slot of the frame, or withdraw

from communication in order to save energy. The main objective of the scheduling algorithm is to coordinate the action of each node in each time slot in order to reduce failed transmissions.

First each node chooses a random action for each slot. The actions are performed simultaneously and the nodes that were successful in establishing communication receive positive feedback. If an action was successful, it will be repeated in the same slot in the next frame (“win-stay”); otherwise an action from the set of available actions will be randomly selected when the executed action failed (“lose-shift”).

The actions could be chosen using a *uniform* or *biased* probability distribution. With the uniform scheme, all actions in the set of available actions have equal selection probability. With the biased distribution, the probability of an action to be chosen as the next action to perform is exponentially proportional to the probability of successfully performing that action (this is updated in every frame).

To avoid interference with an already successfully coordinated communication, a Clear Channel Assessment (CCA) check is done by nodes before transmitting. At the beginning of each slot, a contention window is used for performing CCA. The length of this contention window is decreased for nodes that have previously established successful transmissions in that slot. Nodes that performed successful transmissions will get higher priority with this mechanism.

This protocol is compared with McMAC [21] which is a frequency-hopping protocol with multiple rendez-vous. The proposed collision-free schedule protocol performs better than McMAC since McMAC does not provide contention-free access. It is also more energy-efficient compared to McMAC.

In [3] the authors define  $\alpha$  as the ratio of the number of time slots per frame to the number of generated data messages at all source nodes in the network. For large enough  $\alpha$ , the optimal solution for their proposed contention-free multi-channel protocol is achieved more easily and more often, and also the latency and the packet loss are reduced. But increasing  $\alpha$  will result in more idle listening. When the number of slots per frame is larger, nodes might try to transmit in more slots before the receiver node concludes that the action of listening is a good one.

The performance of the algorithm also depends on the sleep threshold, which should not be set too large, otherwise a lot of nodes might decide to sleep. The protocol uses a schedule-based approach and requires time synchronization for the whole network.

The scheme presented in [29] uses two metrics, with two separate goals: to minimize the energy consumption and to minimize the delay when delivering packets to the destination. The proposed scheme can adapt to various types of traffic occurring in the network and it tries to minimize both energy and delay for the respective traffic classes. Two types of packets are considered: urgent and normal packets where urgent packets need to be delivered with a minimum delay to the end-user, while normal packets do not have this requirement and therefore are delivered with a minimum energy.

This scheme could be used for example in environmental monitoring applications, where sensed data that are not critical or urgent can be forwarded with a minimum energy, while those having more important information (such as a fire detection) must be delivered faster to the destination.

An important issue in WSNs is node failures. Path discovery is usually done by flooding. When more paths from a node to the sink are maintained, the network can

usually recover from failures on the primary path without invoking network-wide flooding for path discovery. This is very important in sensor networks because flooding involves energy consumption and this reduces network lifetime.

In [30] multipath routing in WSNs is studied using two different approaches to construct multi-paths between two nodes: one with disjoint paths, in which the alternate paths do not intersect the primary path or each other, and one with braided paths, in which there are no completely disjoint paths but rather many partially disjoint alternate paths.

The *resilience* of a scheme refers to the likelihood that, when the shortest path fails, an alternate path is available between a source and the sink. But with multiple paths, there is *maintenance overhead*, which is considered a measure of the energy required to maintain these alternate paths using periodic keep-alives. Becoming more resilient typically consumes more energy so there is a tradeoff here.

Two different failure modes are considered in [30]: *isolated* node failures, where each node has an independent probability of failure, and *patterned* failures, in which all nodes within a certain fixed radius fail simultaneously. For the same patterned failure resilience, the braided multi-paths are preferred over disjoint paths because they have about 50% higher resilience to isolated failures and a third of the overhead for alternate path maintenance. Also, it is considered that disjoint alternate paths are hard to construct in localized algorithms due to lack of information.

DRCS [31] is a multi-channel protocol that performs channel selection and routing together, in order to improve the battery lifetime in WSNs. A node from the network has a receiver channel (the channel used for receiving all incoming packets) and a transmit

channel (the channel to which a node temporarily switches to transmit a packet), which is the receiver channel of its intended destination. Nodes listen on their receiver channels by default.

Selection of channels and parents (nodes to send to) is done based on a battery health parameter  $H$  and a path metric that is calculated using a link quality parameter (ETX). The battery health-metric  $H$  of a node represents its remaining battery lifetime which depends on its currently estimated energy usage. The quality of a route is estimated using a path metric that is calculated as the sum of the expected number of transmissions (ETX) on each of its links. An ETX for a link is defined as the expected number of transmission attempts required to deliver a packet successfully over the link.

A node chooses the route with the lowest path metric to the sink. When choosing the transmit channels, the goal is to prolong the lifetime of the neighboring node with the worst battery health-metric. Channel selection is connected to the parent selection and this will determine the route that the message takes to get to the destination, therefore the proposed approach leads to a joint channel selection and routing in the WSNs. DRCS generates a higher packet delivery ratio in comparison to the tree-based multi-channel protocol TMCP.

Traffic patterns change significantly during runtime and some nodes or segments of the network may have more traffic than others. Channel assignment algorithms that do not take traffic patterns into account may waste channels on nodes with no traffic while assigning too few channels to nodes with heavy traffic. Channel assignment strategies that can dynamically adapt to traffic pattern changes are desirable.

The traffic-aware channel assignment algorithm presented in [32] collects traffic information from 2-hop neighbors, and uses a specific algorithm to assign channels among 2-hop neighbors. This traffic-aware frequency assignment is incorporated into the existing MMSN MAC and it is compared with two conventional frequency assignment methods: even selection and eavesdropping.

An important aspect of this algorithm is assigning a traffic weight to each node, from which we could infer its future reception data rate. The goal is to minimize the maximum load of any channel within the 2-hop neighborhood of any node in order to decrease the number of radio collisions. This problem is similar to the load balancing job scheduling problem, which is NP-hard.

This traffic-aware algorithm is a greedy algorithm. First, nodes exchange information so that each node knows its 2-hop neighbors' IDs and traffic weights. Then nodes choose their channel in the decreasing order of their traffic weight: The node with the greatest traffic weight among its two communication hops selects the available channel with the least load, and then beacons the channel choice in its 2-hop neighborhood. The corresponding channel is then updated. Before they make their decision, nodes wait for nodes with greater weight and for nodes with equal weight but lower node ID.

This algorithm uses just one radio and can only use one channel at one time. Two groups of experiments were considered, by varying the system loads and the number of available channels, respectively. Simulation results show that this algorithm greatly improves multi-channel MAC performance, especially when the number of channels is medium, such as 4, 6, 8, and when the system load is light or heavy. It significantly

enhances the packet delivery ratio and throughput, while reducing channel access delay and energy consumption.

This traffic-aware channel assignment algorithm has a more efficient channel usage than the algorithm in [33] which also has the capability of dynamically changing the radio frequency. In this algorithm nodes make local decisions to dynamically change the radio frequency. A *home frequency* is assigned to each node such that the network throughput is maximized. Initially all nodes have the same home frequency (channel) and when the channel becomes overloaded the nodes move to another channel.

When a node wants to send data to a node that has a different home channel, it temporarily switches to that channel to send the data. The main rule of this algorithm is to “cluster” into the same channel the nodes that communicate frequently and to separate into different channels those that do not. The protocol is distributed and a node decides to switch channels based on an estimated communication success probability that is obtained after nodes exchange state information.

The switching is done based on a probability and two goals are alleviating congestion and avoiding having all nodes jump to the new channel. The nodes that behave predominantly as sinks have priority to switch channels first (i.e., initiate the cluster split). If a node communicates heavily with another node that already switched, then it follows the node into the new cluster and in this way the communication between different clusters is minimized. Unlike other algorithms, this algorithm does not neglect the time to switch between two channels.

A novel interference-aware multi-channel media access control (IMMAC) protocol which can be easily implemented in WSNs is presented in [34]. This single-interface



multi-channel protocol addresses issues such as dynamic traffic, non-negligible channel switching overhead, and multi-channel utilization support for broadcast. IMMAC does not require time synchronization.

Assuming that the relative traffic on each link of the network is known a priori, each node is assigned an initial channel for data reception such that the maximum concurrency is achieved. Channels are assigned three priorities when choosing the initial channel for each node: high priority, middle priority, and low priority. Channels that are not used in the 2-hop neighborhood have the high priority. Channels not used by 2-hop away neighbors but used by one-hop neighbors have the middle priority. The other channels have the low priority. The initial channels are adjusted according to dynamic traffic.

Each node maintains a local channel assignment table (CAT), in which it records the node ID and channels assigned among its 2-hop neighbors. To maintain fairness on every channel, the scheduler uses a round-robin method to transmit on each nonempty channel queue. A node maintains a nonempty queue channel to transmit until the queue becomes empty, or the staying time is longer than a threshold value  $T_{\max}$ , which depends on the tradeoff between switching overhead and packet delay. The protocol uses receiver-based channel switching to support unicast and broadcast communication.

The traffic is not always known prior to channel assignment and it could also change over time. CRNs are able to perceive the environment and adapt according to the conditions of the network (neighboring nodes' channels, traffic demand etc.). A cognitive process is able to make future decisions based on the success of past decisions and to adapt its channel assignment when traffic changes.

Next we present some aspects related to radios and multi-radio networks. The previous algorithms were mainly single-radio multi-channel.

## 2.4 Radio and Multi-Radio Issues

Battery life of sensor nodes is a very important issue in WSNs and energy efficiency could be obtained by designing sensors that consume low power, by energy harvesting, and by efficient scheduling at the node and network level. Most of the energy is spent on wireless communication.

There are two types of data transmission in traditional wireless networks: unicasting (one-to-one) and multicasting (one-to-many). In addition to these, in WSNs several data source nodes sense the data and send them back to the sink node. This is called *reverse multicasting* (many-to-one) or *data aggregation routing*.

If two children sensor nodes use two different channels to transmit data to the same sensor node, then this sensor node will need two radios to receive data simultaneously. Otherwise, a single radio will have to switch its channel to receive from its children nodes and this will increase the latency. Therefore, for minimum latency, any sensor node that is on the data aggregation tree has to have a number of radios that is greater than or equal to the number of children nodes. Each radio is assigned to a channel from each sending node.

The same is true for sinks: a sink with a single radio that receives data from nodes on different channels has to constantly switch channels, so multiple radios are needed to improve the performance at the sink. It was shown that if we increase the number of sinks and/or increase the number of radio interfaces in the sink, then we can obtain a better

performance at the sink, which results in an overall performance improvement within the network [35].

There are different types of radios in a multi-radio sensor node that may differ in terms of their communication capabilities, energy efficiency, and usage. High bandwidth, long range radios are more energy-efficient in terms of energy expended per bit transmitted than low bandwidth, short-range radios. But high bandwidth radios consume more power than low bandwidth radios when they are idle.

Therefore, when there is not a lot of data to be transmitted it is not recommended to activate many high bandwidth radios because this may waste energy. On the other hand, long-range radios have a greater reach and can reduce the network diameter. Using long range radios will decrease the latency involved in delivering sensory data to a specific destination. Radios with higher power and longer range may need to be activated in order to achieve a connected network.

In networks with multiple types of radios, the problem of *energy-efficient radio activation* is to minimize the total energy spent by the active radios across all nodes in order to maintain a connected network. This problem is NP-hard and four approximation algorithms are presented in [36].

Most of the work on multi-radio networks was not focused on WSNs and the results cannot be directly applied to them. The approaches could be divided into two categories: centralized and distributed. Centralized approaches can further be divided into: flow based, graph based, and partition based.

When assigning channels, some approaches give priority to high load edges or to the edges that are closer to the sink. The routing algorithm uses shortest path routing or sets

of paths between two communicating nodes. Most approaches are not realistic because they assume that the traffic is constant while network traffic can be bursty.

Channel scheduling in Multi-Radio Multi-Channel (MR-MC) wireless networks has been studied under the assumption that the communication range equals the interference range. In reality, this assumption cannot be satisfied because a node's interference range is usually larger than its communication range. This means that two nodes might interfere with each other even if they cannot communicate directly with each other.

Under the *physical interference-free model*, two nodes interfere with each other if and only if their physical distance is less than or equal to the interference range. Under the *hop interference-free model*, two nodes interfere with each other if and only if their hop distance is less than or equal to a number of hops  $H$ . When node location is unavailable, the hop interference-free model can be applied, even if it is not as precise as the physical interference-free model.

In [37] the authors proved that channel scheduling is NP-hard under both of these models in MR-MC wireless networks. Also, they proposed a polynomial-time approximation scheme (PTAS) framework that could be used for channel scheduling under both interference models in such networks.

The factors that affect the energy consumption of a radio are: the sensor node hardware architecture, the background network traffic, and the network topology. The authors of [38] draw conclusions related to designing energy-efficient data transmission protocols that are congestion aware. Their experimental results show that always transmitting packets of maximum size can have a negative impact on radio's energy consumption. By properly adjusting the run-time parameters, such as the packet size and

the packet generation period (packet rate), the radio's energy consumption can be significantly reduced under heavy network traffic.

Some approaches use two or more radios with different energy and throughput characteristics to minimize the total energy consumption. In [38] the authors studied a heterogeneous sensor node equipped with 802.15.4 and 802.11b radios, and performed an extensive set of experiments, while varying various network parameters. They conclude that by properly adjusting some network parameters, such as packet size and transmission period, energy savings of up to 50% can be achieved under heavy network traffic conditions when a CSMA-based MAC is used. Also a proper pairing of processor and radio is crucial for taking full advantage of the energy efficiency of higher bandwidth radios.

Chipcon's 802.15.4 compliant radio is a low power radio that can provide data rates of 250Kbps, which are sufficient for many sensor network applications. High-bandwidth radios such as 802.11b have higher power consumption but they provide significantly higher data rates (11Mbps). This means that high bandwidth radios can transmit more data in less amount of time compared to low power radios and they can be more energy-efficient. For example, the energy per bit of the 802.15.4 radio (979nJ/bit) is almost 9 times higher than the energy per bit of an 802.11b radio (112nJ/bit) [39].

In addition to the energy per bit metric, there are other system aspects that affect radio's energy dissipation. Besides high power consumption, radios such as 802.11b have a large startup time. The time it takes to power up and configure the radio is orders of magnitude higher (approximately 2 to 3) compared to that of low power radios such as

802.15.4 (less than 2 ms). So every time we power up the radio a fixed energy overhead is created, which is independent of the size of data to be transmitted.

Taking into account this high startup cost, high bandwidth radios are more energy-efficient only when a large number of bytes have to be transmitted because the high startup cost gets amortized as more and more bytes are transmitted. These radios should not be used if there is little or no data to send or if data needs to be sent only occasionally.

The low-bandwidth radios are less energy-efficient but consume much less energy when idle. Also, they are able to quickly change their state from sleep to active in order to send data, and then deactivate. Therefore they are suitable for transmitting small amounts of data as well as remaining “vigilant” for long time periods.

In [39] the dual-radio system has the following characteristics: The primary processor and high-bandwidth radio remain off until triggered by the application while the secondary processor and radio operate on low power and remain vigilant. The secondary radios form a network that is always available while the primary network exists only when it is needed. We assume that the low-bandwidth network is connected and not partitioned but the two network topologies are not necessarily the same. A neighboring node over the low-bandwidth radio is not necessarily a neighboring node over the high-bandwidth radio and vice versa.

In the centralized approach, there is an always-on node called *topology controller* (possibly co-located with the sink) that handles requests for end-to-end paths received from the low-bandwidth radios. Based on stored information about pre-established routing paths, this node sends wakeup requests to low-bandwidth radios, which turn on

their CPU and high-bandwidth radio for data transfer. Just the nodes that are required for the path are woken up. In the distributed approach, each node keeps in non-volatile memory a path to all potential destinations in its routing table.

If a node fails, all the paths that contain that node are invalidated. When the controller wants to send a wakeup message to a node, if no path is found, the wakeup request is sent to all the nodes in the network.

In both approaches the low-bandwidth radios communicate using routing trees or flooding. An important advantage of the centralized approach in terms of energy is the fact that if multiple concurrent transfers are involved, then the controller does not have to turn on any extra nodes if the network is sufficiently connected. This algorithm provides energy savings of more than 60% compared to alternative approaches while incurring only a moderate increase in application latency.

The conclusion of [40] is that the pairing of two complementary radios with different range characteristics (maximum ranges 80 meters vs. 800 meters) enables greater range diversity at lower energy cost than a single radio. The radios are divided into two main categories: a long-range radio that enables communication over long distances but requires more power and a shorter range radio that is more power-efficient but cannot provide communication over longer distances.

Experimental results show that choosing a long range radio and using its lower power settings for short range communication is far less efficient than using a short range radio. Directional antennas can increase the communication range of short range radios but directional antennas are bulky and they are not suitable for many mobile WSNs.

The new multi-radio sensor platform used in [40] is called the Arthropod and it pairs two radios - CC2420 and XE1205 - that offer diversity in frequency, power, and range, and it is a good solution for range-adaptive mobile WSNs. Also, a unified link layer is presented that, with the help of an adaptive algorithm based on reinforcement learning, chooses which radio to use for communication between a pair of nodes. The reinforcement technique that is used is Q-Learning, which provides a simple reward for making correct decisions and an ability to explore other operating points periodically [40].

This algorithm introduces energy and resource overhead as it has to continually monitor and learn channel characteristics for the two radios and determine which one is more efficient in terms of energy. Its main advantages are: up to 52% improvements in energy efficiency over using only one of the two radios on the platform, and the ability to easily implement the learning algorithm with limited memory and computational overhead on a mote-class sensor platform.

A two-stage multi-radio algorithm that could adapt to traffic demand is presented in [41]. Using a greater number of radios increases the cost in terms of energy consumption, so this algorithm initially assigns a minimum number of transceivers to provide network connectivity over multiple channels (traffic independent or TI channel assignment) and then enables additional links to respond to increased traffic demands (traffic dependent or TD channel assignment).

Some literature discusses the optimal placement of nodes and the challenges resulted from the fact that obstructions to the line-of-sight between nodes are usually dynamic, so the optimal placement cannot be decided at the deployment time. Also, connectivity at



deployment time can change over the network lifetime. WSNs have to adapt to changes in network conditions and the existence of alternative routes is helpful. Channel assignment has to take these aspects into consideration.

IEEE 802.15.4 standard uses fixed channel assignment but the beacon node can change the operating frequency of its network if the nodes report excessive levels of interference. There is also literature that explores interference between co-existing networks. A solution for avoiding interference is adjusting the transmission power to a level that is sufficient to reach the receiving node.

## **2.5 Conclusions**

By using multiple channels and multiple radios in WSNs, we can achieve a better support for applications that require high network throughput.

To provide efficient communication, sinks and cluster heads need multiple radios so they can communicate with multiple nodes simultaneously. Also, they need sufficient power supply. Nodes in a cluster could take turns in assuming the role of cluster head.

A disadvantage of the protocols that use a control channel is the existence of a potential single point of failure: the control or primary interaction channel. The connectivity is lost if there is interference or severe fading on this channel. Also, in the presence of PUs that reclaim specific channels, the connectivity of the network has to be maintained.

The distributed protocols are considered more practical and suitable for real-world applications than the centralized ones. In the case of WSNs, it is important to balance the energy consumption and this is achieved by distributing the work load among sensors

with the purpose of equalizing their remaining lifetimes. Some traffic could be redirected so as to ensure a balanced utilization of all the sensor nodes.

The network has to adapt to failure of nodes which could deactivate paths between a node and the BS, and determine changes in traffic. Maintaining multiple paths is useful in such cases but this increases energy consumption. In the presence of PUs the availability of channels changes, so channel assignment algorithms have to adapt accordingly.

The use of multiple channels in WSNs allows parallel transmissions that improve the performance of the network, resulting in increased throughput, decreased collision probability, and thus better energy efficiency. Because WSNs are limited in terms of power and processing capabilities, developing simple algorithms that could be scaled to large networks is of major importance in improving the performance of such applications. We focus on developing and analyzing algorithms that work for such networks also.

### 3. TWO CHANNEL ASSIGNMENT ALGORITHMS FOR MULTI-RADIO MULTI-CHANNEL NETWORKS

#### 3.1 Introduction

CRNs allow SUs to communicate using the underutilized spectrum. The presence of PUs may disrupt the SUs' communication since they cannot use any longer the channels reclaimed by the PU. This leads to a decrease in network performance or even network partition. The use of multiple radios and channels can improve the performance of such networks.

The robustness constraint requires that if a channel is reclaimed by a PU, then the resulting secondary user topology still preserves the connectivity between any two nodes. In this chapter we propose two distributed algorithms for channel assignment which have low overhead, are scalable with the number of sensor nodes, and satisfy the robustness constraint.

These channel assignment methods could be used in WSNs, which have limited resources and therefore a low number of radios, or in mesh networks where the mesh nodes (e.g. routers, gateways) are more powerful devices that could have a higher number of radios. In the United States and Canada there are 11 channels available for use in the 802.11b 2.4GHz WiFi Frequency range. This standard is defined by the IEEE.

Next we present some work related to channel assignment in multi-radio multi-channel networks other than WSNs. These algorithms are computationally expensive and are not energy-efficient; therefore they are not suitable for networks like WSNs.

In [42] the authors introduce a centralized channel assignment algorithm, MCCA (Maxflow-based Centralized Channel Assignment), developed for multi-radio wireless mesh networks in order to maximize network capacity and reduce interference. The assignment is independent of any particular traffic profile and is done such that the most critical links (e.g. those carrying large flows) experience the least possible interference. This paper does not address the issue of channel switching in the presence of a PU. Also the centralized mechanism proposed here is not scalable for a large network such as a WSN.

Another centralized channel assignment algorithm called UBCA (Utility Based Channel Assignment) is presented in [43]. This is a traffic independent algorithm in which the delivery probability of the wireless links and their usefulness are taken into account to make a better decision for assigning good channels to good links. This channel assignment algorithm assigns channels starting from links with higher utility, where the utility of a link is defined as the number of times a link  $e$  participates in constructing the shortest paths between the gateway and other nodes.

UBCA is compared with three relevant channel assignment algorithms: the Common Channel Assignment (CCA) [44], the Connected Low Interference Channel Assignment algorithm (CLICA) [45], and the distributed channel assignment (ROMA) [46]. UBCA achieves significant improvement in terms of reducing the interference and increasing the network capacity.

CCA applies the same channel assignment pattern for all nodes, i.e. the first radio of all nodes is tuned to the first channel, the second radio is tuned to the second channel etc. The number of channels is therefore equal to the number of radios. CLICA is a centralized algorithm that finds connected and low interference topologies by visiting nodes in the order of their priority, which depends on their distance to a reference node (the gateway) and the number of free radios they have. ROMA is a distributed algorithm that can be used in a network with at least one gateway. Each gateway produces a channel sequence  $(c_1, c_2, \dots, c_n)$  and broadcasts it. The node which is  $i$  hops away from the gateway will select channels  $c_{i-1}$  and  $c_i$ . At the end, each node will have a common channel with its previous node on the path to the gateway, and a common channel with its neighbors at the same and lower level.

The algorithm in [47] is a static and traffic independent channel assignment algorithm that tries to minimize the overall network interference by using Tabu search. This is the first work that establishes good lower bounds on the optimal network interference. In [48], the authors proposed a semi-dynamic and distributed channel assignment mechanism called SICA that uses game theory and takes the co-channel interference into account. It uses an online learning method to assign the best channel to each radio using information gathered during the channel sensing periods. The nodes continuously refine their decision based on changes in the wireless environment. SICA outperforms Urban-X [49], another interference-aware channel assignment mechanism, even using fewer radio interfaces per node (2 instead of 3).

Urban-X assigns channels giving priority to nodes based on the number of active flows they have: nodes having higher priority have more chances to occupy the best

channels. Nodes broadcast control messages over a common channel up to 2-hop neighbors. Unlike Urban-X and many other channel assignment algorithms, SICA does not use a common channel between all nodes but the synchronization is achieved through exchanging messages. The use of a common channel can be wasteful when only a few interfaces are available.

In [10] network robustness and channel interference are jointly considered when developing centralized and distributed algorithms. The problem that we address is similar to the one presented in this article but our methods are much simpler and they could be used for devices with limited resources, like the ones from large scale WSNs.

The centralized and distributed algorithms presented in [10] require multiple negotiations between nodes and may require cascaded switching of multiple users. The proposed solutions outperform existing interference-aware approaches when PUs appear, and achieve similar performance at other times. Compared to existing channel assignment methods for multi-hop multi-radio networks [47] [50], in [10] the channels are carefully assigned so that the PUs appearance will not partition the network. The algorithms in [10] are compared with INSTC [50].

Some papers use primary user behavior models but predictability is not always possible. Like [10], our methods do not assume a predictable PU activity. But knowing the maximum number of channels that could be reclaimed simultaneously by PUs could help in choosing the best parameters for our algorithm.

### 3.2 Problem Definition

We consider a WSN consisting of  $n$  homogeneous sensor nodes  $s_1, s_2, \dots, s_n$  and a sink node  $S$ . We assume the nodes are densely deployed and the WSN is connected. The sink node  $S$  is used to collect data and is connected to the network of sensors. Data collection follows a convergecast communication model, where data flow from many nodes (e.g. the sensors) to one (the sink).

We model the network as an undirected graph  $G = (V, E)$ , with the set of vertices (or nodes) being the set of sensors and the sink. An edge exists between two nodes if they are within each other's communication range. We assume that each sensor node is equipped with  $Q$  radios and there are  $C$  channels available, where  $C \geq Q$ . The objective is to find a channel assignment  $A$  which assigns to each node's radio a channel such that the resulting topology is connected, robust to a PU, and has a reduced interference.

Let  $A(u)$  denote the set of channels assigned to the node  $u$ , where  $|A(u)| = Q$ . Based on the channels assigned to the radios at each node, a channel assignment  $A$  generates a new undirected graph  $G_A(V, E_A)$  where  $E_A = \{(u, v, c) : (u, v) \in E \text{ and } c \in A(u) \cap A(v)\}$ . Note that multiple edges may exist between two nodes if they share more than one channel, where one edge corresponds to a channel.

The *robustness constraint* requires that  $G_A$  is not partitioned in the presence of a PU which communicates on a channel  $c_p$ . In that case, all the edges in  $G_A$  assigned to  $c_p$  are removed. The resulting graph must be connected. In this chapter we assume that the PU can affect part of the network or the entire network (e.g. transmission of the TV tower), but only one channel is used by the PU at one time. We assume that the PU is using the reclaimed channel for some amount of time.

It is easy to observe that if each node has at least two radios, then there exists a channel assignment that satisfies the robustness constraint. Just consider the case when all nodes have assigned the two channels  $c_1$  and  $c_2$  to both radios. If the PU uses one of the channels, let's say channel  $c_1$ , then the topology remains connected using the channel  $c_2$ . The drawback of such an assignment is a high interference.

Transmissions on different channels can run in parallel with no interference. In order to reduce the network interference, the objective is to assign communication on nearby edges to different channels. If edges within interference range have assigned the same channel, then interference has to be addressed at the MAC level.

**Channel Assignment for a Robust Topology Control (CA-RTC) Problem:** Given a graph  $G$  find a channel assignment  $A$  such that  $G_A = \{(u, v, c) : (u, v, c) \in E_A\}$  is robustly connected for any channel  $c$  and the network interference is minimized.

According to the proof in [10], the CA-RTC problem is NP-complete. We consider that the WSN is homogeneous and all the sensor nodes have the same transmission range and the same interference range. The MAC protocol used is IEEE 802.11 DCF [51]. Each link is supporting communication in one direction at one time (half-duplex).

### 3.3 Solutions for the CA-RTC Problem

#### A. Grid-based Channel Assignment

In this section we propose the Grid-based Channel Assignment mechanism. The results of this study have been published in [52]. The monitored area is divided into grids, see Figure 3.1. Let  $r$  be the communication range of each sensor. It is assumed that sensors know their location information using GPS or other localization protocols [53]. In



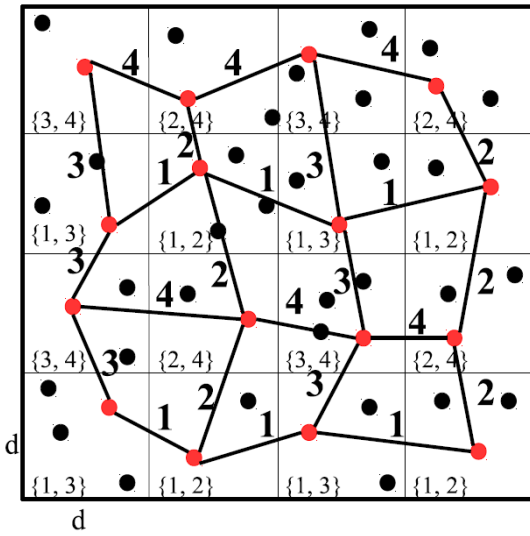
addition, since WSNs are densely deployed, it is assumed that each grid cell has at least one sensor. The *neighboring cells* of a certain cell are those placed above, below, left, and right. The grid size is  $d = r/\sqrt{5}$  so that any two sensors in neighboring cells can communicate directly.

Consider the case when each sensor node has  $Q = 2$  radios and there are  $C = 4$  channels available. Each sensor computes the grid cell that it belongs to based on its GPS coordinates. A static channel assignment can be allocated in this case, as illustrated in Figure 3.1. Figure 3.1a shows the channels used to communicate between neighboring cells, while Figure 3.1b shows the channels allocated for the communication inside a cell. For example, the representative of the bottom leftmost cell is assigning the channels  $\{1, 3\}$  to its radios, it uses channel 1 to communicate with the right representative and channel 3 to communicate with the representatives placed above. Also, the representative of this cell uses channel 3 for the intra-cell communication.

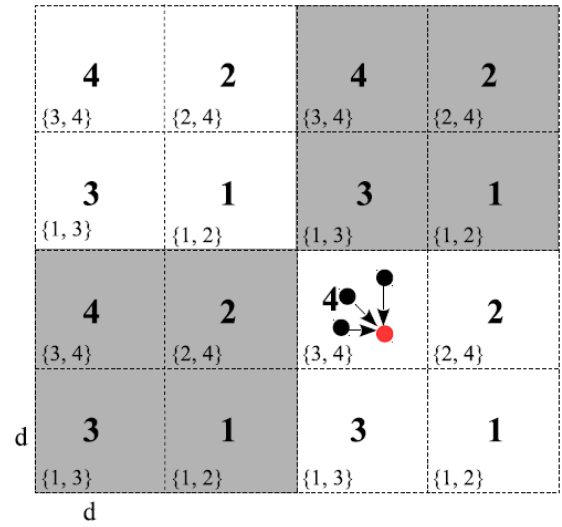
Communication between cells is accomplished through cell *representatives*. Each cell locally selects a representative, which can be for example the sensor node with the largest remaining energy. In case of a tie, the representative role is assumed by the sensor node with the largest ID. The cell representative is in charge with forwarding messages between cells and with transmission of messages from/to the nodes inside the cell. It has to be noted that the representative consumes the largest amount of energy in the cell. In order to avoid sensor energy depletion, representatives have to be re-elected periodically.

The grid-based channel assignment mechanism satisfies the robustness constraint; that is the topology remains connected when a PU reclaims any of the channels. Figure

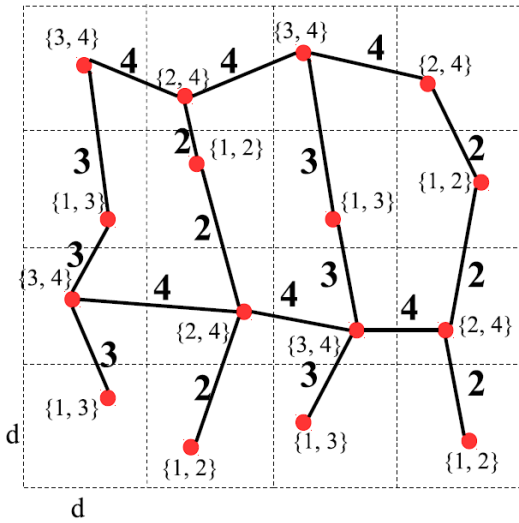
3.1c shows the resulting topology when channel 1 is reclaimed by a PU. The inter-cell communication topology still ensures communication between any cells.



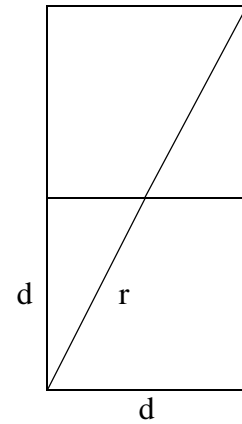
a. Channel allocation for each grid cell



b. Intra-cell communication



c. Channel assignment when channel 1 is reclaimed by PU



d. Sensor transmission range  $r$

Figure 3.1: Grid-based Channel Assignment

Intra-cell communications which originally were scheduled on the channel 1 now take place on the other assigned channel. For example, in Figure 3.1b grid cells with channel assignment  $\{1, 2\}$  are using channel 1 for intra-cell communication. If channel 1 is reclaimed by a PU, then intra-cell communication switches to channel 2. The proposed mechanism has a low overhead and can be easily extensible to the case with more radios and more channels, see the discussion in [52].

### *B. Distributed Channel Assignment*

In this section we present a distributed protocol for channel assignment. This protocol has been published in the article [54]. We assume that the sensor network is connected to the sink when nodes have communication range  $r$ . This mechanism has two phases:

- Phase 1: neighbor discovery and setting up the distance from the sink
- Phase 2: channel assignment

Below, we describe the mechanism for  $Q = 2$  radios and  $C = 4$  channels, and then we explain how it can be modified for a larger number of radios and/or channels.

#### *1) Neighbor Discovery and Setting up the Distance from the Sink:*

In this phase, all nodes use the same channel, let us say channel 1, to communicate. We assume that each node has a unique ID. Each node broadcasts a *Hello* message containing the node ID. To reduce the probability of interference, a node waits a random delay before sending the broadcast. The neighbor information is stored locally by each node. The sink node participates in this step as well.

The nodes that have the sink as a neighbor resend the *Hello* message containing their neighbor table. In this way, the sink collects information on its 2-hop neighborhood, while all other sensor nodes have 1-hop neighbor information.

In the second part of this phase, the sink broadcasts a message *Hops*, which contains a parameter hops - the number of hops from the sink. A node receiving a *Hops* message will retransmit the message in two cases: (i) if this is the first *Hops* message received, or (ii) if this message has a shorter distance to the sink. In both cases the node updates its shortest distance to the sink, increments the hops counter, and then retransmits the *Hops* message. At the end of this phase, the sink has 2-hop neighborhood information, and each node knows its number of hops from the sink.

## 2) Channel Assignment:

In this protocol, the nodes assign channels starting from the sink, in an incremental approach. The sink S has two radios, similar to all other nodes. S chooses two channels arbitrarily for its radios, let us say channels 1 and 2. Next, the sink S uses the 2-hop neighborhood information acquired in the previous step to assign channels to the sensor nodes on cycles of length 4 incident to S. The mechanism is shown in *SinkLocalNeighborhoodAssignChannels(S)* procedure where S is the sink.

### ***Algorithm 1 SinkLocalNeighborhoodAssignChannels(S)***

- 1: compute all cycles of length 4
- 2: **for** each cycle of length 4 **do**
  - 3: assign channels to all nodes on the cycle that have not assigned their channels yet, such that all 4 channels are used on the cycle
- 4: **end for**

- 5: broadcast *SinkLNChannelSet* containing all channel assignments to its 2-hop neighborhood (TTL = 2)
- 6: wait a random time and broadcast *ChannelSet*(S, 1, 2, TTL = 1)

**Algorithm 2 AssignChannels(*v*)**

- 1: **if** *v* receives channel assignments  $\{x, y\}$  in a *SinkLNChannelSet* message **then**
  - 2: node *v* assigns channels  $\{x, y\}$  to its radios
  - 3: node *v* waits a random time and broadcasts *ChannelSet*(*v*, *x*, *y*, TTL = 1)
  - 4: return
- 5: **end if**
- 6: set the waiting time  $t = \text{Time}(v\text{hops})$
- 7: record channels assigned by neighbor nodes based on *ChannelSet* messages received
- 8: when timer *t* expires, examine the recorded neighbor channels and compute the two least used channels *x* and *y*
- 9: node *v* assigns channels  $\{x, y\}$  to its radios
- 10: node *v* waits a random time and broadcasts *ChannelSet*(*v*, *x*, *y*, TTL = 1)

The sink assigns channels to cycles of length 4. The objective is that each pair of neighboring nodes communicates on a different channel, so that all four channels are used. The edges incident to the sink will have values 1 and 2 in arbitrary order. The other two edges on the cycle are assigned values 3 and 4 in arbitrary order. Figure 3.2a shows an example of channel assignments for the nodes on a cycle. An alternative assignment is when nodes are being assigned values  $\{1, 2\}$ ,  $\{1, 4\}$ ,  $\{3, 4\}$ , and  $\{2, 3\}$ , where  $\{1, 2\}$  is the assignment of the sink S. One of these two assignments is assigned to each cycle, arbitrarily.

If some of the nodes on a cycle have been already assigned channels, then the others are assigned channels following the same rule (e.g. each of the four channels on an edge), see Figure 3.2b.

The main goal in assigning channels to cycles of length 4 is to use all channels in order to increase diversity and to reduce communication interference. Cycles of length 4 can be computed using a depth-first search approach starting from S, and checking paths of length 4. The algorithm has to check adjacency lists of the first three nodes on the path, and if the fourth node is a sink neighbor. The complexity is  $O(\alpha^3)$ , where  $\alpha$  is the node degree.

The sink then broadcasts a message *SinkLNChannelSet* to its 2 hop-neighborhood, containing a list with channels assigned by the sink so far (e.g. sensor nodes on cycles of length 4). Nodes in the list receiving this message will assign their channels accordingly. In order to broadcast this message to the 2-hop neighborhood, the message will have a TTL = 2. The sink and the nodes whose channels have been set up at this step will then broadcast a *ChannelSet* message to their 1-hop neighbors (TTL = 1) to advertise their channels. These messages are sent with a small random delay to avoid interference.

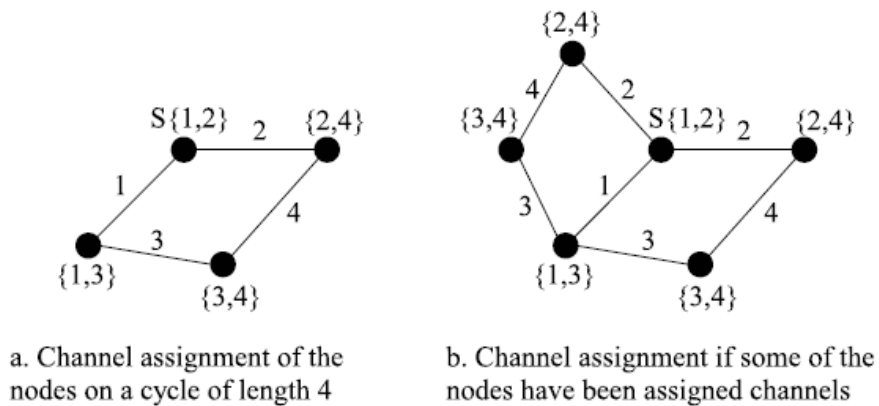


Figure 3.2: Channel assignment for cycles of length 4

Let us consider the example in Figure 3.3. The sink computes three cycles: (S, G, F, D), (S, G, K, J), and (S, B, A, C) and assigns channels to the nodes S, G, F, D, K, J, B, A, C. A list containing these channel assignments is sent using a *SinkLNChannelSet* message with  $TTL = 2$ . All these nodes in the list will assign their channels accordingly and then broadcast a *ChannelSet* message with  $TTL = 1$ . At the end of this step, all the nodes will have their channels assigned except the nodes E, H, I.

Sensor nodes wait for *SinkLNChannelSet* and *ChannelSet* messages in order to select their channels. The mechanism used by a sensor node  $v$  is shown in the *AssignChannels(v)* procedure. If a *SinkLNChannelSet* message is received and  $v$  has been assigned channels  $\{x, y\}$  by the sink, then  $v$  uses those channels and broadcasts *ChannelSet(v, x, y, TTL = 1)* after a random time to inform its neighbors about its channels selection.

Sensor nodes without a channel assignment wait a time proportional to the distance from the sink (e.g. number of hops from the sink  $vhops$ ). The waiting time is computed as  $Time(vhops) = vhops \times hopDelay$ , where  $hopDelay$  is the delay per hop and it must account for the propagation delay, algorithm execution time, and the maximum waiting time of a node before sending the *ChannelSet* message. In this way the nodes at distance 1 will set up their channels first, followed by the nodes at distance 2, then 3, and so on.

Considering the example in Figure 3.3a, nodes H and E are at distance 2 from the sink and they will establish their channels first. Node H knows the channels assigned by G and K, and will select the two least used channels  $\{2, 3\}$ . Node E knows the channels assigned by D and F and will assign channels  $\{1, 4\}$ . After nodes at distance 2 have set

up their channels, nodes at distance 3 will follow. In our example, the node I is at distance 3 from the sink and will select the least used channels  $\{2, 4\}$ .

The proposed channel assignment mechanism satisfies the robustness constraint; that is the topology remains connected when a PU reclaims any of the channels. Figure 3.3b and 3.3c show the resulting topology when channel 1 or 4 is reclaimed by a PU.

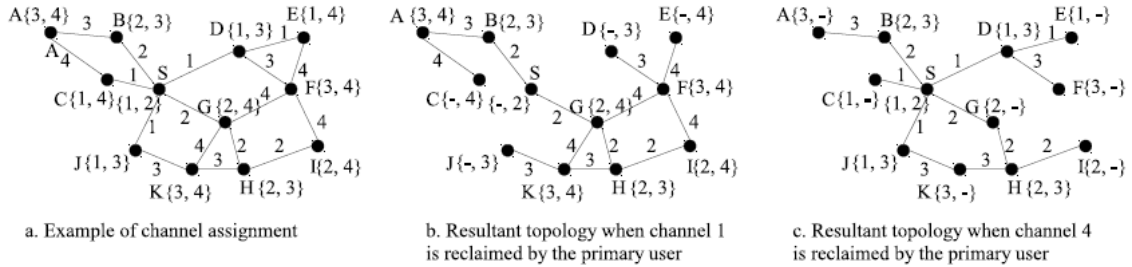


Figure 3.3: Example using the Distributed Channel Assignment Algorithm

**Theorem** Assuming that the starting wireless sensor topology is connected to the sink, the Distributed Channel Assignment algorithm terminates in finite steps and achieves robustness upon termination.

*Proof:* The proof is by induction. The starting point of the algorithm is the sink S, which assigns channels 1 and 2 to its radios. Let us assume that all nodes that have sent the *ChannelSet* message are robustly connected to the sink; that means that if one channel is reclaimed by a PU then the node is still connected through a path to the sink.

Let us take a node  $v$  which has just assigned its channels using the *AssignChannels* procedure. If  $v$  has received only one message of type *ChannelSet* from a node  $u$ , then  $v$  will assign the same two channels as  $u$ , so it will be robustly connected to the sink since  $u$  has this property based on the inductive assumption. If the node  $v$  has received two or more *ChannelSet* messages, then  $v$  selects two channels used by its neighbors. If one of



the channels is reclaimed by the PU, then  $v$  remains connected to at least one neighbor  $u$ . Since  $u$  is robustly connected to the sink based on the inductive assumption,  $v$  will remain connected to the sink after the PU's appearance.

The Distributed Channel Assignment protocol has a low overhead of  $O(n)$  messages, where  $n$  is the number of nodes. Each node is sending a *Hello* message, one (or few) *Hops* messages, and a *ChannelSet* message, while the nodes in the 2-hop neighborhood of the sink send two extra messages.

The method presented above for  $Q = 2$  radios and  $C = 4$  channels can be easily extended if there are more radios and/or channels available. The algorithm starts from the sink, which finds cycles of length at most  $C$ . For this the sink needs  $\lfloor C/2 \rfloor$ -hop neighborhood information. The sink then assigns channels to the nodes on the cycles and broadcasts this information using a *SinkLNChannelSet* message. Other nodes assign their channels incrementally, similar to the mechanism described above. The main objective is to have the nodes select the least used channels, in order to reduce interference.

### 3.4 Simulations

In this section we evaluate the performance of our Distributed Channel Assignment (Distributed-CA) mechanism using Ns-3 network simulator [55]. The performance of this algorithm is compared to the Grid-based Channel Assignment (Grid-CA) [3] mechanism.

#### A. Simulation Environment

In the simulations, we set the node communication range  $r = 100\text{m}$ . To be able to compare our protocol with the Grid-CA, we consider that the deployment area is a square divided into grids. We vary the number of rows (which is the same as the number of

columns) between 5 and 25 with increment of 4, see Table 3.1. The grid cell size is computed as  $d = r/\sqrt{5}$ . As a result the deployment area side varies between 223m and 1118m and the number of sensor nodes  $n$  varies between 75 and 1875. The sensors are deployed randomly in the monitoring area, and we place the sink  $S$  in the middle of the area.

The transmission rate for the wireless radio is 1Mbps. In our simulation, we consider that the nodes have  $Q = 2$  radios and  $C = 4$  channels. Once the sensors are deployed, they use a channel assignment mechanism to assign channels to their radios. To test the performance of the resulting topology, we employ the following shortest-path data gathering protocol. The sink  $S$  broadcasts a beacon message in the whole network. Each sensor node sets up its routing table with the next hop being the node from which the beacon with the smallest number of hops to the sink was received.

We use a convergecast communication model where traffic flows from the sensor nodes to the sink. At the MAC level we use CSMA for the wifi channels. Each sensor node has a parameter  $p$  - the probability that the node sends a message in each iteration (e.g. each second). Sensors send 656 bytes data packets every second. In the simulations, we represent two cases:  $p = 100\%$  and  $p = 30\%$ . We run each simulation scenario 5 times using different seed numbers and report the average values in the graphs. Each simulation scenario is run for 20 seconds. The simulation parameters are presented next.

Number of rows	Number of cells	Area side length, m	Number of sensors
5	25	223.607	75
9	81	402.4926	243

13	169	581.3782	507
17	289	760.2638	867
21	441	939.1494	1323
25	625	1118.035	1875

Table 3.1: Simulation parameters

### B. Simulation Results

In the first experiment the sensor nodes use the Distributed-CA protocol (denoted alg. 2 in the figures) to assign channels to their radios. We compare the performance of the network when the nodes are equipped with multi-radios multi-channels, versus the case when nodes have single-radio single-channel. Simulation results are presented in Figures 3.4-3.7. Two cases are considered, when nodes send data packets every second with probability  $p = 30\%$  and  $p = 100\%$ . Figure 3.4a shows both data transmitted by the sensors as well as data received by the sink. We can observe that in both cases a higher throughput is received for multi-radio WSNs. Some of the data packets are lost due to collisions. It is known that collisions increase as packets get closer to the sink.

Figure 3.4b compares the end-to-end delay. This metric is larger for  $p = 100\%$  since more packets are being transmitted. We also observe that the single-radio network has a slightly smaller delay. This is because in the multi-radio topology the shortest path to the sink may have a larger length than in the single-radio topology.

In Figure 3.4c we can observe that the delivery ratio decreases with an increase in the number of sensors, due to the collisions in the network. If nodes send with  $p = 30\%$ , then a higher delivery ratio is achieved due to a smaller number of collisions. This graph also

illustrates the advantage of a multi-radio network, which has an increased delivery ratio compared to single-radio networks.

In the second experiment we compare the performance of the two algorithms, Grid-CA (alg. 1) and Distributed-CA (alg. 2), in Figures 3.5 and 3.6. In Figure 3.5 nodes send data with probability  $p = 100\%$ , while in Figure 3.6 nodes send data with probability  $p = 30\%$ . The results in these figures are consistent.

Figures 3.5a and 3.6a show that the Distributed-CA has a higher throughput than the Grid-CA mechanism. This is due to a smaller number of collisions, since the nodes in the same grid cell compete when sending messages to the same representative. The figures also show a higher throughput when nodes are equipped with multi-radios, illustrating the benefit of using multiple transmissions without interference on different channels.

The end-to-end delay is analyzed in Figures 3.5b and 3.6b. Overall, the two algorithms have comparable end-to-end delays which increase with the size of the network. The Distributed-CA has a slightly smaller delay, since nodes may find a shorter path than in the case when data is forwarded through the cell representatives. The single-radio single-channel case has a slightly smaller delay than the multi-radio multi-channel one since nodes may find shorter paths to the sink.

Figures 3.5c and 3.6c present the delivery ratio. Distributed-CA has a higher delivery ratio due to a reduced number of collisions. Multi-radio networks have a higher delivery ratio due to channel diversity which decreases the number of collisions.

In the third experiment we test the behavior of the network in the presence of a PU. In Figure 3.7 the percentage of the area affected by the PU varies between 0 (no PU) to 1 (PU is affecting the whole area). The PU is using a single channel and two scenarios were

considered depending on whether the PU channel is identical to a sink channel or not. When the PU is on the same channel as the sink, the sink will use the other radio for wireless communication.

In the presence of a PU the topology is still connected, even though it is sparser. Note that in our simulations we consider that the sink is placed in the middle of the monitored area. We take the monitored area to be a square with side  $L$ . The area affected by the PU is taken to be the rectangle with height  $L$  and width  $PU_{\text{fraction}} \times L$ , starting from the origin.

In Figure 3.7a we can observe that a higher drop in data rate occurs when  $PU_{\text{fraction}} \geq 0.6$  and the PU is using one of the sink channels. In these cases the sink is in the area affected by the PU and it can communicate on a single radio only. This will reduce network capacity at the sink. If the sink is in the area affected by the PU we can also see an increase in the end-to-end delay, especially in the case when the PU is on a sink channel, see Figure 3.7b.

Figure 3.7c is consistent with the previous graphs and it shows a decrease in the delivery ratio as the sink is affected by the PU. Even though not represented in this graph, it is evident the drastic impact of a PU on a single-radio network. In such a case, for  $PU_{\text{fraction}} \geq 0.6$  the delivery ratio is 0.

In summary, the topology resulted by applying our Distributed Channel Assignment mechanism is robust to the presence of a PU. Simulation results show the benefit of using multi-radio networks and show the network performance in the presence of a PU. The protocol has better performance compared to the related algorithm Grid-based Channel Assignment, which requires traffic to flow through the cell representatives.

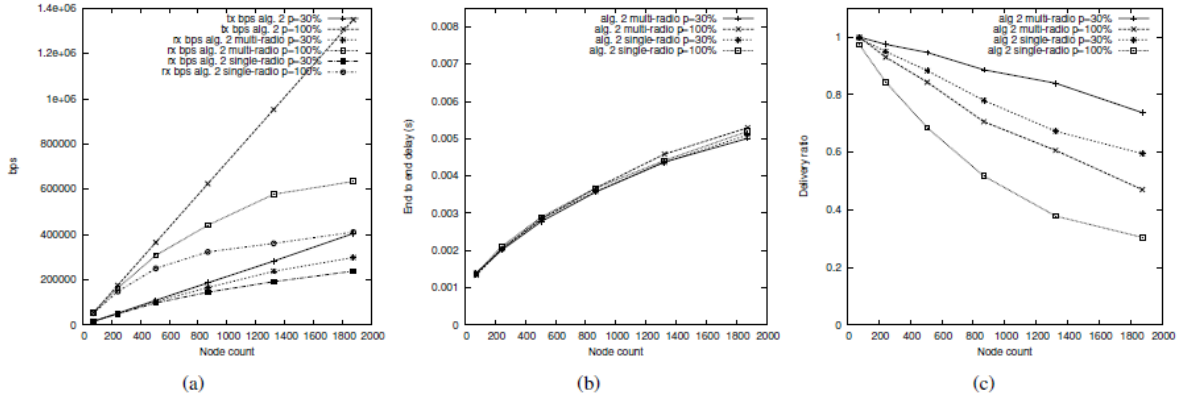


Figure 3.4: Distributed-CA - Comparisons between multi-radio and single-radio WSNs  
 (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio.

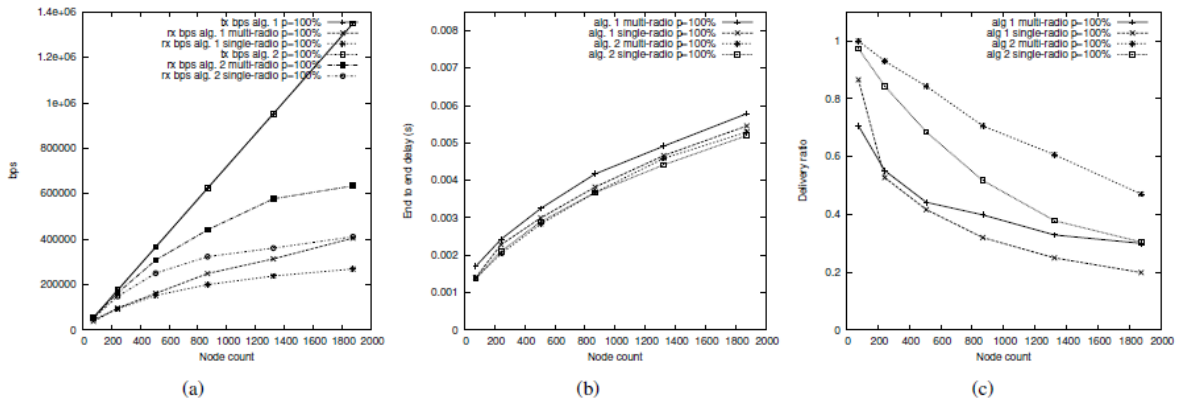


Figure 3.5: Comparisons between Distributed-CA (alg. 2) and Grid-CA (alg. 1). Multi-radio versus single-radio WSNs when  $p = 100\%$ . (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio.

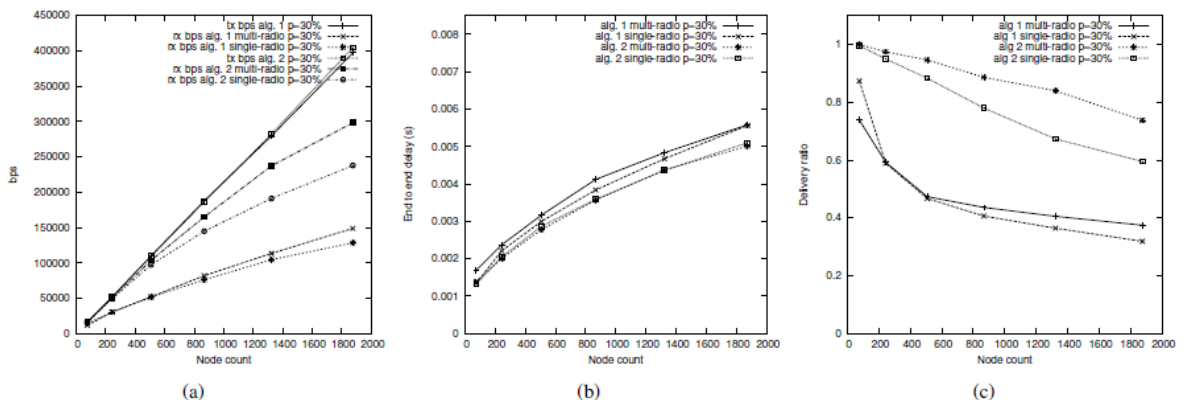


Figure 3.6: Comparisons between Distributed-CA (alg. 2) and Grid-CA (alg. 1). Multi-radio versus single-radio WSNs when  $p = 30\%$ . (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio.

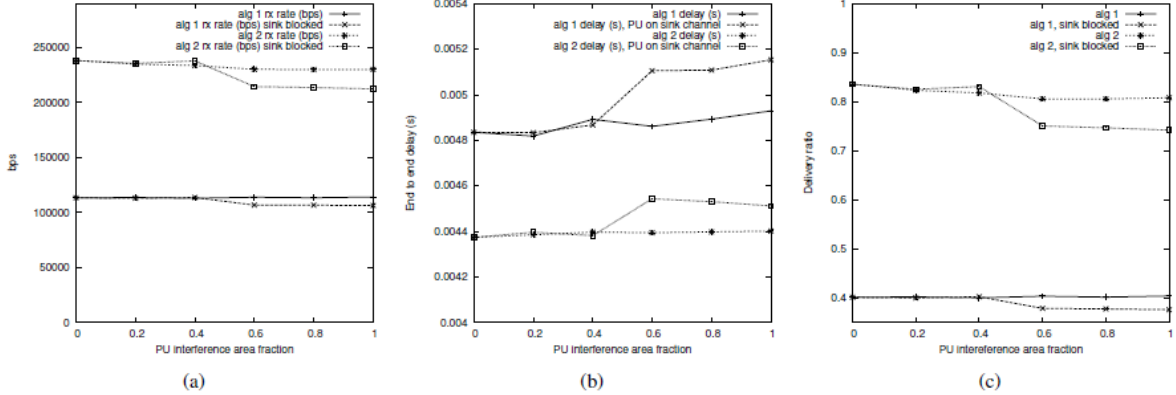


Figure 3.7: Comparisons between Distributed-CA (alg. 2) and Grid-CA (alg. 1). The impact of PU on WSN performance. (a)Data throughput. (b)End-to-end delay. (c)Delivery ratio.

### 3.5 Conclusions

We proposed the Grid-based Channel Assignment protocol in which traffic flows through cell representatives. In this algorithm devices have to be equipped with GPS or other localization protocols.

For the second protocol, Distributed Channel Assignment, the devices do not need GPS information. In the protocol, nodes assign their channels starting from the sink in an incremental approach, using channels less used by their neighbors, while maintaining the robustness constraint. Simulation results using Ns-3 show the benefit of using a multi-radio topology and the robustness of the proposed Distributed Channel Assignment protocol in the presence of a PU.

Both protocols are robust to the presence of a PU on a certain channel. In the presence of PUs that reclaim any of the assigned channels, the nodes are able to continue to deliver data to the sink following the same or a different path. The algorithms could be used in networks with limited resources like WSNs.

## 4. ROBUSTNESS TO PRIMARY USERS THAT RECLAIM MULTIPLE CHANNELS AT ONCE

### 4.1 Introduction and Problem Definition

In this chapter we propose an algorithm that is robust to PUs that could reclaim multiple channels at once. We consider a network where all the nodes have  $C$  channels and  $Q$  radios. We suppose that nodes are able to calculate their position using GPS or other localization protocols [53] and based on this information they can determine the grid cell they belong to. Each grid cell chooses a representative that will be used to communicate with the representatives of the neighboring cells (above, below, left and right). In Figure 4.1 the cell representatives are marked in red.

If the communication range of the sensors is  $r$ , then the grid size is chosen to be  $d=r/\sqrt{5}$  so that the representatives of any two neighboring cells can communicate directly. The algorithm is assigning channels only to the representatives and we suppose that any other node in a grid cell uses one of its representative's channels to communicate with it. Since our network is considered to be a dense network, we suppose that there is at least one node in each cell, therefore each cell has a representative.

We suppose that the PUs could reclaim up to  $k-1$  channels so in our channel assignment we need to have  $k$  common channels between the representatives of any two neighboring cells. We assume  $k < Q < C$ , otherwise the problem is trivial: if  $Q=C$  all the nodes are assigned all the channels, and if  $k=Q$  there have to be  $k$  common channels



between neighboring representatives and this means again that all of them are assigned the same channels.

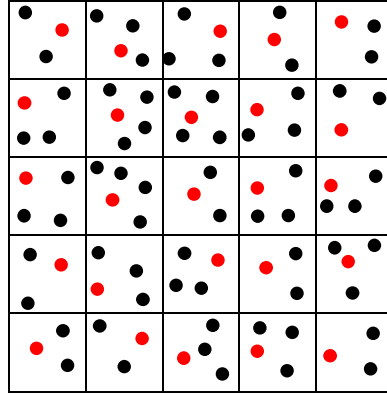


Figure 4.1: Monitored area divided into grids and cells' representatives (red)

For  $k=1$  the network might not be robust to PUs - reclaiming the channel used by a link might partition the network. In this case the network could become disconnected but we included this case in the analysis since our algorithm is a generalized one. This simple case helped us to find patterns for cases where  $k>1$ . We represent the channels assigned to a node as a vector of dimension  $Q$ . The construction of the first row of the grid/matrix of vectors is presented next. We consider that the representatives are indexed like in a matrix, in order from top to bottom and from left to right. The objective is to compute the channel assignment for each representative when  $C$ ,  $Q$ , and  $k$  are given, where  $k < Q < C$ .

## 4.2 Channel Assignment

We assign channels 1, 2, 3, ...,  $Q$  to the representative in cell (1,1) denoted by  $node_{11}$ . In the first table from Figure 4.2 the channels assigned to  $node_{11}$  are in the top leftmost cell. Then, suppose we want  $k$ -connectivity, to support  $k-1$  channels reclaimed by PUs,

we assign the last  $k$  channels of node<sub>11</sub> to its right neighbor, node<sub>12</sub>. So this neighbor will have the following channels assigned to its first  $k$  radios:  $Q-k+1, Q-k+2, \dots, Q$ .

We continue assigning channels to the next radios in order, starting from  $Q+1, Q+2, \dots$  and continuing until we exhaust all the channels or assign all the radios using the remaining channels, whichever comes first. If we exhaust all the channels and there are radios that are unassigned, after assigning the last channel  $C$  we go back to channel 1 and we continue to assign channels in increasing order, starting with 1.

The same method is applied for row 2, starting with node<sub>21</sub> that has the same assignment as node<sub>12</sub> so each element in row 2 will get the last  $k$  channels of the node above it and the next  $Q-k$  channels in increasing order, where channel  $C$  is followed by channel 1.

The first table in Figure 4.2 shows an example, for  $C=8, Q=5, k=3$ . We observe that starting from column 5 the elements start repeating. If we denote by  $A$  the  $6 \times 6$  matrix of vectors represented in the first table of Figure 4.2, we have  $A_{11}=A_{15}, A_{12}=A_{16}, A_{21}=A_{25}, A_{22}=A_{26}$  etc. Matrix  $A$  is symmetric so the first row is the same as the first column. Elements start to repeat from row 5.

A cell from the second table in Figure 4.2 represents the  $4 \times 4$  sub-matrix/block from the upper left corner of matrix  $A$ . Such blocks will repeat horizontally and vertically in a large network where we use this method to assign channels to representatives.

We study patterns in the channel assignments performed with our method. If the dimension of the grid is large enough, the vectors will repeat at a certain point. If we consider a square grid that is cut right before the assignments for the nodes start repeating

horizontally and vertically, the second row of the matrix of vectors is a cyclic permutation of one place to the left for the first row of the matrix.

1, 2, 3, 4, 5	3, 4, 5, 6, 7	5, 6, 7, 8, 1	7, 8, 1, 2, 3	1, 2, 3, 4, 5	3, 4, 5, 6, 7
3, 4, 5, 6, 7	5, 6, 7, 8, 1	7, 8, 1, 2, 3	1, 2, 3, 4, 5	3, 4, 5, 6, 7	5, 6, 7, 8, 1
5, 6, 7, 8, 1	7, 8, 1, 2, 3	1, 2, 3, 4, 5	3, 4, 5, 6, 7	5, 6, 7, 8, 1	7, 8, 1, 2, 3
7, 8, 1, 2, 3	1, 2, 3, 4, 5	3, 4, 5, 6, 7	5, 6, 7, 8, 1	7, 8, 1, 2, 3	1, 2, 3, 4, 5
1, 2, 3, 4, 5	3, 4, 5, 6, 7	5, 6, 7, 8, 1	7, 8, 1, 2, 3	1, 2, 3, 4, 5	3, 4, 5, 6, 7
3, 4, 5, 6, 7	5, 6, 7, 8, 1	7, 8, 1, 2, 3	1, 2, 3, 4, 5	3, 4, 5, 6, 7	5, 6, 7, 8, 1

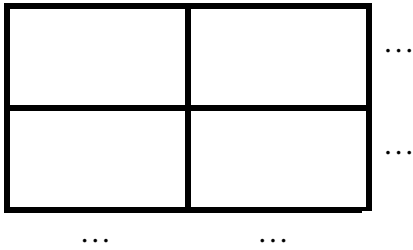


Figure 4.2: Channel assignment for  $C=8$ ,  $Q=5$ ,  $k=3$  and repeating blocks in a large network

This is true for columns also and it can be generalized for every row/column: each row/column is a cyclic permutation of one place to the left for the previous row/column of the matrix. The previous row/column of the first row/column is considered to be the last one.

First we want to find the positions in the first row of the grid where the configuration for the first node will repeat, i.e. will have the same channels assigned to the radios, in the same order, like the channels for node<sub>11</sub>. Suppose node<sub>11</sub> has channels 1,2,3,..., Q assigned to it. We denote by node<sub>1j</sub> the first node in the first row where channel C appears in position Q-k of that node. This means that the last k positions of that node will be filled with 1,2,3, ..., k and node<sub>1,j+1</sub> will start with 1, 2, 3, ... so it will have channels 1, 2, 3, ..., Q assigned to it. Therefore node<sub>1,j+1</sub> is the first node in the first row that has

the same channels as node<sub>11</sub>. We call the  $j \times j$  grid of vectors *basic grid*. Next we compute  $j$ , the size of the basic grid.

We denote by  $n_{\text{cycles}}$  the number of times that all channels from 1 to  $C$  appear in order in the first  $Q-k$  radios of the nodes from the first row, from node<sub>11</sub> to node<sub>1j</sub>. Since  $C$  is on the last position of the first  $Q-k$  radios of node<sub>1j</sub>, we have  $C \cdot n_{\text{cycles}} = 0 \pmod{(Q-k)}$ . We have to find the smallest value for  $n_{\text{cycles}}$  such that this equality takes place.  $n_{\text{cycles}}=1$  is equivalent to  $C$  a multiple of  $Q-k$ . Once we calculate  $n_{\text{cycles}}$  we could find  $j = C \cdot n_{\text{cycles}} / (Q-k)$  or  $j+1 = C \cdot n_{\text{cycles}} / (Q-k) + 1$ .

Figure 4.3 presents the construction of the first row/column of a grid for  $C=5$ ,  $Q=3$ ,  $k=1$  and  $C=5$ ,  $Q=3$ ,  $k=2$ . In both cases  $j=5$  and we observe that element  $j+1=6$  is the same as the first element. In the first case  $n_{\text{cycles}}=2$  because we cycled twice through the elements 1, 2, 3, 4, 5 before we got to our initial configuration, while in the second case  $n_{\text{cycles}}=1$ .

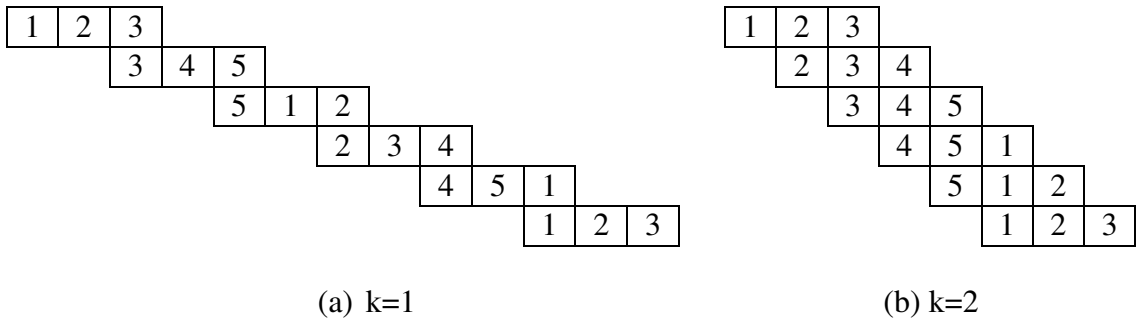


Figure 4.3: Construction of the first row/column of a grid for  $C=5$ ,  $Q=3$

The tables in Figure 4.4 display  $j$  and  $n_{\text{cycles}}$  between parentheses for some values of  $C$ ,  $Q$  and  $k$ . In a network with more than  $j$  columns, starting with column  $j+1$  the columns will repeat, i.e. column  $j+1$  will be the same as column 1, column  $j+2$  will be the same as

column 2 etc. The same statement is true for rows. For Figure 4.2, we have  $j=4$  in the cell corresponding to  $C=8, Q=5, k=3$  in Figure 4.4.

The results from Figure 4.4 help us to choose the grid size when we analyze the distribution of channels assigned to the nodes. We will choose the grid size equal to the numbers in the tables below, for the corresponding values of  $C, Q$  and  $k$ . If the distribution is balanced for these values, then it will be balanced in the whole network because we could consider the network as basic grids put together side by side. We took  $C$  from 3 to 11,  $Q$  from 2 to  $C-1$ , and  $k$  from 1 to  $Q-1$ . For example, for  $k=1$  corresponding to the first table, for  $C=4$  and  $Q=2$  the basic grid will have size  $4 \times 4$  and a bigger grid will have this basic structure repeated horizontally and vertically.

In Figure 4.4 the values in black correspond to balanced distributions while the colored ones correspond to imbalanced distributions. The red ones satisfy the conditions  $Q-2k > 0, n_{\text{cycles}}=1$  (which is equivalent to  $Q-k$  divides  $C$ ) and the green ones satisfy the conditions  $Q-2k < 0, n_{\text{cycles}}=1$  (which is equivalent to  $Q-k$  divides  $C$ ) and  $Q-k$  does not divide  $Q$ . The blue cells are those for which  $n_{\text{cycles}} > 1$  (which is equivalent to  $Q-k$  does not divide  $C$ ) and  $C$  does not divide  $j \cdot k$ .

We are interested in the values for  $C, Q$  and  $k$  that give a balanced distribution for the assigned channels, i.e. all the channels are assigned an equal number of times in a basic grid, which is a square grid with dimensions presented in Figure 4.4 for specific values of  $C, Q$  and  $k$ . The results for square grids with these dimensions can be generalized for bigger grids and will help us choose the best values for  $C, Q$  and  $k$ .

$C \backslash Q$	2	3	4	5	6	7	8	9	10
3	3 (1)								
4	4 (1)	2 (1)							
5	5 (1)	5 (2)	5 (3)						

6	6 (1)	3 (1)	2 (1)	3 (2)					
7	7 (1)	7 (2)	7 (3)	7 (4)	7 (5)				
8	8 (1)	4 (1)	8 (3)	2 (1)	8 (5)	4 (3)			
9	9 (1)	9 (2)	3 (1)	9 (4)	9 (5)	3 (2)	9 (7)		
10	10 (1)	5 (1)	10 (3)	5 (2)	2 (1)	5 (3)	10 (7)	5 (4)	
11	11 (1)	11 (2)	11 (3)	11 (4)	11 (5)	11 (6)	11 (7)	11 (8)	11 (9)

k=1

C\Q	3	4	5	6	7	8	9	10
4	4 (1)							
5	5 (1)	5 (2)						
6	6 (1)	3 (1)	2 (1)					
7	7 (1)	7 (2)	7 (3)	7 (4)				
8	8 (1)	4 (1)	8 (3)	2 (1)	8 (5)			
9	9 (1)	9 (2)	3 (1)	9 (4)	9 (5)	3 (2)		
10	10 (1)	5 (1)	10 (3)	5 (2)	2 (1)	5 (3)	10 (7)	
11	11 (1)	11 (2)	11 (3)	11 (4)	11 (5)	11 (6)	11 (7)	11 (8)

k=2

C\Q	4	5	6	7	8	9	10
5	5 (1)						
6	6 (1)	3 (1)					
7	7 (1)	7 (2)	7 (3)				
8	8 (1)	4 (1)	8 (3)	2 (1)			
9	9 (1)	9 (2)	3 (1)	9 (4)	9 (5)		
10	10 (1)	5 (1)	10 (3)	5 (2)	2 (1)	5 (3)	
11	11 (1)	11 (2)	11 (3)	11 (4)	11 (5)	11 (6)	11 (7)

k=3

C\Q	5	6	7	8	9	10
6	6 (1)					
7	7 (1)	7 (2)				
8	8 (1)	4 (1)	8 (3)			
9	9 (1)	9 (2)	3 (1)	9 (4)		
10	10 (1)	5 (1)	10 (3)	5 (2)	2 (1)	
11	11 (1)	11 (2)	11 (3)	11 (4)	11 (5)	11 (6)

k=4

C\Q	6	7	8	9	10
7	7 (1)				
8	8 (1)	4 (1)			
9	9 (1)	9 (2)	3 (1)		
10	10 (1)	5 (1)	10 (3)	5 (2)	
11	11 (1)	11 (2)	11 (3)	11 (4)	11 (5)

k=5

C\Q	7	8	9	10
8	8 (1)			
9	9 (1)	9 (2)		
10	10 (1)	5 (1)	10 (3)	
11	11 (1)	11 (2)	11 (3)	11 (4)

k=6

C\Q	8	9	10
9	9 (1)		
10	10 (1)	5 (1)	
11	11 (1)	11 (2)	11 (3)

k=7

C\Q	9	10
10	10 (1)	
11	11 (1)	11 (2)

k=8

C\Q	10
11	11 (1)

k=9

Figure 4.4: Tables with values of  $j$  and  $n_{\text{cycles}}$  for specific values of  $C$ ,  $Q$  and  $k$

Since these matrices of vectors have rows/columns that are cyclic permutations of the previous rows/columns, the distribution of the elements in the first row/column gives us an idea about the distribution for the whole grid. Therefore, when we are referring to “distribution” from now on, we consider the distribution of the elements in the first row/column. All the rows/columns will have a distribution equal to this in a basic grid of dimension  $j \times j$ .

Next, we do a theoretical analysis and we give more details related to the meaning of the colors in Figure 4.4, corresponding to imbalanced distributions. Then, we present an algorithm that helps us to observe patterns and to determine if the distribution of assigned channels is balanced.

### 4.3 Theoretical Analysis

Our theoretical analysis is divided into three cases:  $Q-2k>0$ ,  $Q-2k=0$  and  $Q-2k<0$ .

#### Case 1: $Q-2k>0$

We represent the channels assigned to a node as a vector of dimension  $Q$ . As we construct the first row or column of the grid/matrix of such vectors, the elements in the middle of this vector ( $Q-2k$  of them) will be *skipped* when creating the next/previous vector/node. From now on, by *skipping* an element we mean not *doubling* it in the next or previous vector. The next element of vector  $j$  is considered to be vector 1 and the previous element of vector 1 is considered to be vector  $j$ .

We analyze some distributions for  $Q-2k>0$  or  $Q>2k$ . For  $Q=3$ ,  $k=1$  we have balanced distributions for  $C$  odd and imbalanced distributions for  $C$  even. In Figure 4.5a, where  $j=3$ , we observe that the elements in the middle of the vector: 2, 4 and 6, are skipped while the elements 1, 3 and 5 are doubled.

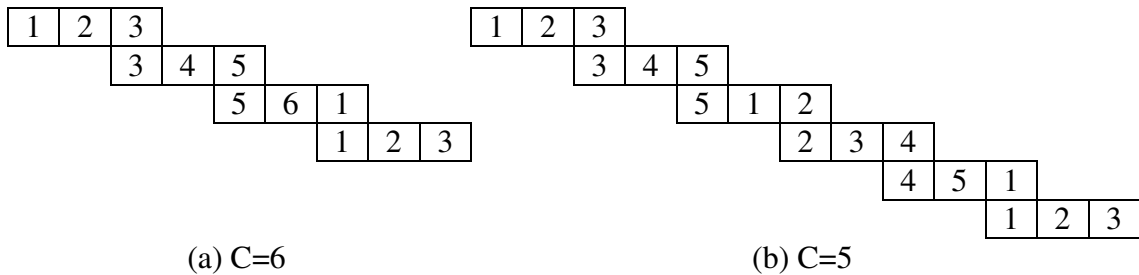


Figure 4.5: Construction of the first row/column of a grid for  $Q=3$ ,  $k=1$

For  $Q=3$ ,  $k=1$ , if  $C$  is even, see Figure 4.5a, the odd numbers are doubled and the even numbers are not, i.e. they are skipped when constructing subsequent channel



assignments. For  $C$  odd, see Figure 4.5b, all the channels are doubled and all the channels are skipped once and this does not imbalance the distribution.

For  $Q=4, k=1$  we obtain imbalanced distributions for  $C$  a multiple of 3, like 6 and 9. This happens because the two elements in the middle, in these cases 2, 3; 5, 6 and also 8, 9 for  $C=9$  get skipped continuously while the other elements get doubled.

As a general rule, we observe that if  $Q-2k>0$ , the  $Q-2k$  elements in the middle get skipped continuously while the other elements get doubled if  $n_{\text{cycles}}=1$  (which is equivalent to  $C$  a multiple of  $Q-k$ ). In this case channel  $C$  will be on the position  $Q-k$  of the node  $j$  in row 1 of the grid and the pattern will repeat starting at the following node, leading to the same elements being skipped again. For example, in Figure 4.6 all the  $Q-2k$  elements in the middle get skipped and the two shaded cells contain the same elements. The red elements in the tables from Figure 4.4 correspond to values for  $C, Q$  and  $k$  that satisfy the conditions  $Q-2k>0$  and  $n_{\text{cycles}}=1$  (which is equivalent to  $Q-k$  divides  $C$ ).

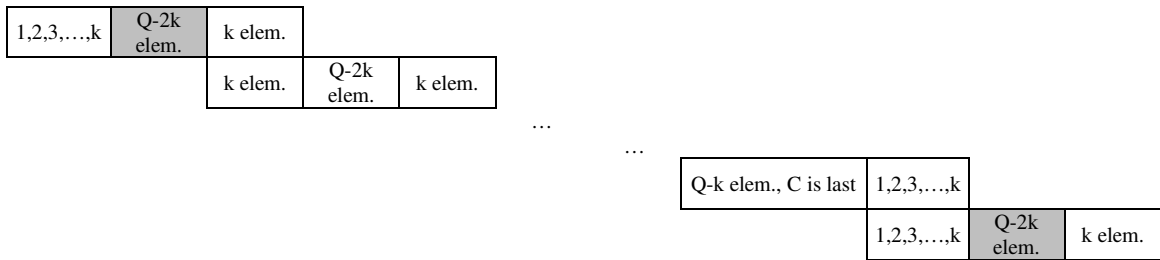


Figure 4.6: Patterns of skipped elements for  $C$  a multiple of  $Q-k$  and  $n_{\text{cycles}}=1$

In the stairs configuration from Figure 4.7 we call the shaded cells *borders* and the unshaded cells *centers*. The elements in the centers are displayed again in Table T from Figure 4.7, for a better understanding. In table T, of dimensions  $C \times n_{\text{cycles}}$  or  $9 \times 2$ , the elements in centers were added in order from left to right, top to bottom, by leaving blank/empty cells corresponding to the elements in borders.

From Figure 4.7, we observe that every time one border element is missing in table T, containing elements from centers, it is replaced by two copies of the same element from borders in our sought distribution, corresponding to the first row/column of vectors. This makes the elements in borders appear one more time in the distribution, in comparison to the other elements. It is easy to prove that the distribution of channels is balanced when all the columns of table T are covered with an equal number of empty/blank cells.

1	2	3	4	5	6	7	8												
						7	8	9	1	2	3	4	5						
												4	5	6	7	8	9	1	2

		3	4	5	6			9
1	2	3			6	7	8	9

Table T

Figure 4.7: Patterns of skipped elements for  $C=9$ ,  $Q=8$ ,  $k=2$

We try to find patterns for the blue cells. We observe that except for the case  $C=10$ ,  $Q=9$ ,  $k=5$ , all of them satisfy the condition  $Q-2k>0$ . For  $C=9$ ,  $Q=8$ ,  $k=2$  we have  $j=3$  and  $n_{cycles}=2$ . From table T in Figure 4.7, we observe that we have  $j=3$  blocks of  $k=2$  empty/blank cells. In total we have  $j \cdot k=6$  empty/blank cells and  $6 < C$  so the 6 blank cells cannot cover all  $C=9$  channels. Therefore for cases where  $j \cdot k < C$  we have imbalanced distributions. In Figure 4.4, all the blue cells in the tables corresponding to  $k=1$  and  $k=2$  have  $j \cdot k < C$ .

In addition to this, in order to have an equal number of empty/blank cells in each of the  $C$  columns of the table T,  $C$  must divide the total number of blank cells, which equals  $j \cdot k$ . We observe that all the blue cells in the tables corresponding to  $k>2$  have  $j \cdot k$  not divisible by  $C$ . So we found a pattern for the blue cells:  $n_{cycles}>1$  (which is equivalent to

$Q-k$  does not divide  $C$ ) and  $C$  does not divide  $j \cdot k$ . This includes the condition  $j \cdot k < C$ . We observe that for all the black cells from Figure 4.4,  $C$  divides  $j \cdot k$ .

**Case 2:  $Q-2k=0$**

Next we analyze the case  $Q-2k=0$  or  $Q=2k$  and show that we always obtain a balanced distribution of channels for this case. Figure 4.8 represents the construction of the first row/column of the basic grid, which contains  $j$  vectors. The first  $k$  elements of the first vector, marked in grey (upper left) are the same as the last  $k$  elements of the last vector, also marked in grey (bottom right) so if we bring the last cell under the first cell we observe that all the cells, i.e. all the elements in them get doubled. This means that the distribution is balanced for  $Q=2k$ .

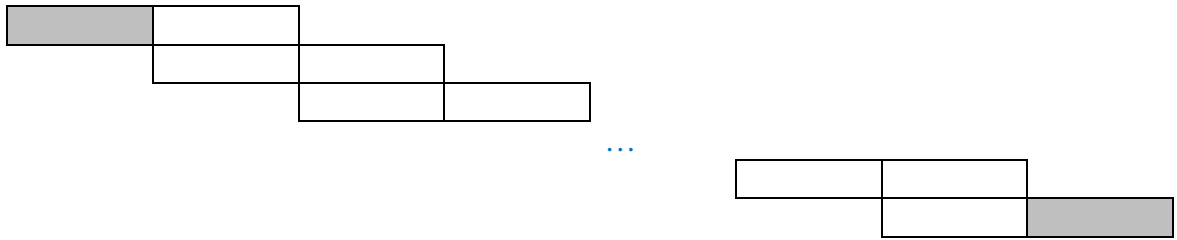


Figure 4.8: Blocks for  $Q-2k=0$

**Case 3:  $Q-2k < 0$**

All the cells colored green in Figure 4.4, representing imbalanced distributions, have  $Q-2k < 0$  and  $n_{\text{cycles}}=1$ , which is equivalent to  $Q-k$  divides  $C$ , more precisely  $j=C/(Q-k)$ . In addition to these, there is only one case for which  $Q < 2k$  and the distribution is imbalanced:  $C=10$ ,  $Q=9$ ,  $k=5$ . In Figure 4.9 we assumed that  $Q-k$  divides  $Q$ , for a nicer representation of the configurations.

We call the grey parts of the vectors *heads* and the white parts *tails*. The heads have length  $Q-k$  and the tails have length  $k$ . If  $n_{\text{cycles}}=1$  or  $j=C/(Q-k)$ , it is obvious that in such a representation, by concatenating all the heads in order from left to right (or top to bottom) we get the elements  $1, 2, 3, \dots, C$  in this order. We have  $j$  vectors/rows in such a representation.

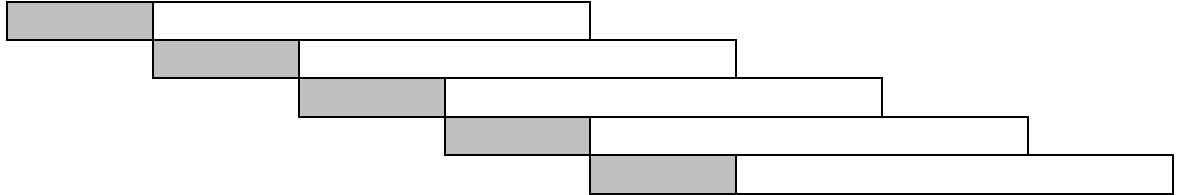


Figure 4.9: Blocks for  $Q-2k < 0$

In Figure 4.10 we numbered the blocks of  $Q-k$  elements from 1 to  $j$ . From this figure it is easy to see that in cases where  $n_{\text{cycles}}=1$  and  $Q-k$  divides  $Q$  we get balanced distributions: the blocks numbered  $1, 2, 3, \dots, j$  appear an equal number of times (four times) in Figure 4.10, representing the vectors from the first row/column of a basic grid of dimension  $j \times j$ .

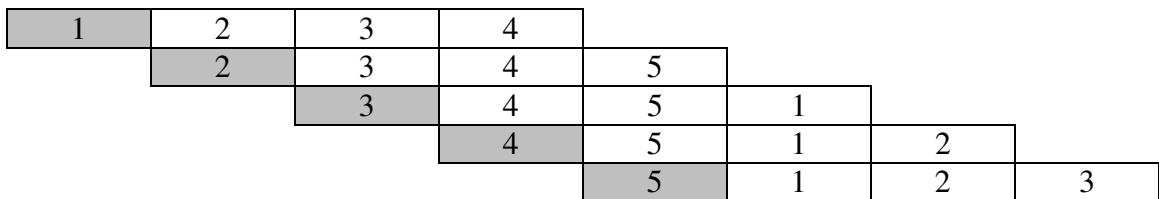


Figure 4.10: Blocks for  $Q-2k < 0$  and  $Q-k$  divides  $Q$

We analyze cases for which  $Q-k$  does not divide  $Q$  and show that they give imbalanced distributions. Suppose  $Q-2k < 0$  and  $n_{\text{cycles}}=1$  (which is equivalent to  $C$  a multiple of  $Q-k$ ). In Figure 4.11 we divided each vector in parts that have  $Q-k$  elements. The portions left at the end, colored in dark grey, have each  $r=Q \bmod (Q-k)$  elements and

contain elements  $e_i$ ,  $i$  from 1 to  $r$ ,  $r < Q-k$ , for which  $e_i \bmod (Q-k) = i$ . But  $e_i$ 's for different levels are not equal so the numbers in the dark grey cells are  $j \cdot r$  unique elements.

We numbered the blocks for an easier understanding. In some blocks we also enumerated some elements within parentheses. If we consider just the white blocks, the distribution is balanced, as we showed for Figure 4.10. Each block is repeated  $Q \text{ div } (Q-k)$  times, in our case 4 times. But the elements in the dark grey cells make the distribution imbalanced since these unique elements appear one more time in the distribution that we seek.

1 (1,2, ...)	2	3	4					
	2	3	4	5				
		3	4	5	1			
			4	5	1	2		
				5 (... C-1,C)	1 (1,2, ...)	2	3	

Figure 4.11: Blocks for  $Q-2k < 0$  and  $Q-k$  does not divide  $Q$

So the distribution for a subcase where  $Q-k$  does not divide  $Q$  is imbalanced. Therefore we found a rule that is followed by all the green cells corresponding to imbalanced distributions:  $Q-2k < 0$ ,  $n_{\text{cycles}}=1$  (which is equivalent to  $Q-k$  does not divide  $C$ ) and  $Q-k$  does not divide  $Q$ .

#### 4.4 Finding the Distribution of the Assigned Channels

Next we develop an algorithm to find the distribution of the assigned channels. In Figure 4.12 we moved the numbers on the right to the next row, so that each column contains the same numbers.

We can see that our problem can be transformed into a similar one: In a matrix  $P$  with  $C$  columns, going from left to right, top to bottom we fill  $Q=9$  cells with the numbers corresponding to their columns then we leave 3 blank cells, then we fill other  $Q=9$  cells with the numbers corresponding to their columns then we leave 3 blank cells etc. When we get to a row that has the first  $k$  cells, starting from column 1, filled with elements 1, 2, 3, ...,  $k$  we stop. This matrix  $P$  tells us if the distribution is balanced: we count how many empty/blank cells we have in each column and if these numbers are equal for all the columns then our distribution is balanced. Otherwise it is not balanced.

1	2	3	4	5	6	7	8	9		
		3	4	5	6	7	8	9	10	1
1				5	6	7	8	9	10	1 2 3
1	2	3				7	8	9	10	1 2 3 4 5
1	2	3	4	5				9	10	1 2 3 4 5 6 7
1	2	3	4	5	6	7				

Figure 4.12: Configuration for  $C=10$ ,  $Q=9$ ,  $k=7$

We can do the same thing for the non-empty cells: we count how many non-empty/non-blank cells we have in each column and if these numbers are equal for all the columns then our distribution is balanced. Otherwise it is not balanced. The number of non-empty/non-blank cells for each column  $ch$  of  $P$ ,  $ch$  from 1 to  $C$ , represents the number of times channel  $ch$  appears in the final distribution.

The matrix we obtained in this case has  $j+1$  rows and  $C$  columns. We could also concatenate the rows of this matrix and form an array of length  $(j+1) \cdot C$ . Then we could find the distribution of the non-empty/non-blank elements in this array.

We find a general formula for the number of cells that we should leave blank between the groups of  $Q$  elements that are filled. To obtain Figure 4.13, we deleted from Figure 4.12 the elements that were moved.

1	2	3	4	5	6	7	8	9	
		3	4	5	6	7	8	9	10
1				5	6	7	8	9	10
1	2	3				7	8	9	10
1	2	3	4	5				9	10
1	2	3	4	5	6	7			

Figure 4.13: Table P for  $C=10$ ,  $Q=9$ ,  $k=7$

At the end of the first row we have  $C-Q$  empty/blank cells and at the beginning of the second row we have  $Q-k$  empty/blank cells so, in total, we have  $C-Q+Q-k=C-k$  empty/blank cells in a block. In our case  $C-k=10-7=3$ . We want to find formulas for the positions of the blank cells in the table/matrix from Figure 4.13, which we denoted by  $P$ .

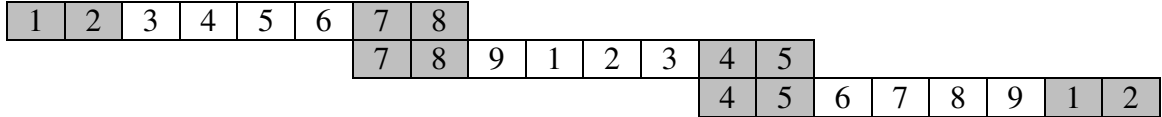
A row that contains a whole block of empty/blank cells contains  $C-k$  empty/blank cells and  $C-(C-k)=k$  non-empty/non-blank cells. A block of empty/blank cells could be split on two rows and a row could contain pieces from two blocks of empty/blank cells like row 2 from Figure 4.14.

We have  $j$  blocks of adjacent empty cells, some of which could be split on two rows. Here the word “adjacent” has a “circular” meaning: a cell from the last column is adjacent to the cell in the first column from the next row.

1	2	3	4	5	6	7			
		3	4	5	6	7	8	9	
				5	6	7	8	9	10
1						7	8	9	10
1	2	3						9	10
1	2	3	4	5					

Figure 4.14: Table P for  $C=10$ ,  $Q=7$ ,  $k=5$

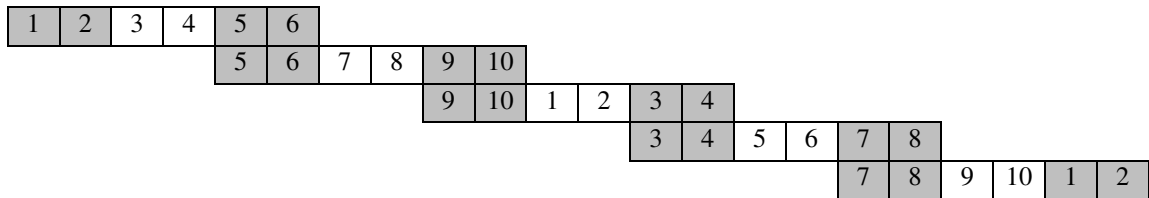
In Figure 4.15a, except for the first one, each channel assignment vector is split on 2 rows in table P, so table P contains  $1+2(j-1)=2j-1=5$  rows. In Figure 4.15b, the third and the fifth channel assignment vector is split on 2 rows in table P, so table P contains  $3+2\cdot 2=7$  rows. The vectors that are split are those that contain channels C and 1 as adjacent channels, in this order.



1	2	3	4	5	6	7	8	
						7	8	9
1	2	3	4	5				
			4	5	6	7	8	9
1	2							

Table P

(a)  $C=9, Q=8, k=2, j=3$



1	2	3	4	5	6				
				5	6	7	8	9	10
								9	10
1	2	3	4						
		3	4	5	6	7	8		
						7	8	9	10
1	2								

Table P

(b)  $C=10, Q=6, k=2, j=5$

Figure 4.15: Constructing tables P



Next we give an algorithm for constructing the tables P. Suppose we know C, Q, k,  $n_{\text{cycles}}$  and j. We define matrix P with C columns and we calculate the row and column in matrix P for each element of the j blocks of C-k empty/blank cells. We use i to index the blocks and c to index elements within a block, with both i and c starting from 1.

If the empty/blank cells span all the columns of P an equal number of times, i.e. if there is an equal number of blank cells in all the columns of P then our distribution is balanced. In other words, if the distribution of the columns of all the blank cells is balanced, then our distribution is balanced. The same statement is true for the non-blank cells, which are the elements from our channel assignment: if the distribution of the columns (or values) of all the non-blank cells is balanced, then our distribution is balanced.

For an easier understanding of the formulas, in Figure 4.16 we represent the structure of an array constructed by concatenating the rows of table P in order from top to bottom.

non-blank elem.	blank elem.	non-blank elem.	blank elem.	non-blank elem.	...
Q elem.	C-k elem.	Q elem.	C-k elem.	Q elem.	

Figure 4.16: The structure of an array constructed by concatenating the rows of table P

We want to find formulas for the columns of the blank cells. This is recommended in cases where the number of empty/blank cells (C-k) is smaller than the number of non-empty/non-blank cells (Q). The index of the empty/blank cells  $E_{ic}$  in the array can be calculated like this:  $\text{index}(E_{ic})=Q+c+(Q+C-k)(i-1)$ .

The columns for the empty/blank cells  $E_{ic}$  can be calculated like this:

$$\text{col}(E_{i1}) = 1+((\text{index}(E_{i1})-1) \bmod C)$$

$$\text{col}(E_{i2}) = 1+((\text{index}(E_{i2})-1) \bmod C)$$

...

$$\text{col}(E_{i,C-k}) = 1 + ((\text{index}(E_{i,C-k}) - 1) \bmod C),$$

where  $i$  takes values from 1 to  $j$ .

The rows for the empty/blank cells  $E_{ic}$  can be calculated like this:

$$\text{row}(E_{i1}) = 1 + ((\text{index}(E_{i1}) - 1) \text{div } C)$$

$$\text{row}(E_{i2}) = 1 + ((\text{index}(E_{i2}) - 1) \text{div } C)$$

...

$$\text{row}(E_{i,C-k}) = 1 + ((\text{index}(E_{i,C-k}) - 1) \text{div } C),$$

where  $i$  takes values from 1 to  $j$ .

We also find formulas for the columns of the non-blank cells. This is recommended in cases where the number of empty/blank cells ( $C-k$ ) is greater than the number of non-empty/non-blank cells ( $Q$ ). The index of the non-empty/non-blank cells  $N_{ic}$  in the array can be calculated like this:  $\text{index}(N_{ic}) = c + (Q + C - k)(i - 1)$ .

The columns for the non-empty/non-blank cells  $N_{ic}$  can be calculated like this:

$$\text{col}(N_{i1}) = 1 + ((\text{index}(N_{i1}) - 1) \bmod C)$$

$$\text{col}(N_{i2}) = 1 + ((\text{index}(N_{i2}) - 1) \bmod C)$$

...

$$\text{col}(N_{iQ}) = 1 + ((\text{index}(N_{iQ}) - 1) \bmod C),$$

where  $i$  takes values from 1 to  $j$ .

The rows for the non-empty/non-blank cells  $N_{ic}$  can be calculated like this:

$$\text{row}(N_{i1}) = 1 + ((\text{index}(N_{i1}) - 1) \text{div } C)$$

$$\text{row}(N_{i2}) = 1 + ((\text{index}(N_{i2}) - 1) \text{div } C)$$

...

$$\text{row}(N_{iQ}) = 1 + ((\text{index}(N_{iQ}) - 1) \text{div } C),$$

where  $i$  takes values from 1 to  $j$ .

The table/matrix  $P$  has  $C$  columns. The total number of rows for table  $P$ ,  $n_{\text{rows}}$ , can be found using the following formula:  $(Q+C-k) \cdot j = n_{\text{rows}} \cdot C$ . The left side of the equality was obtained by adding the number of cells in each of the two categories of blocks:  $j$  non-blank blocks, containing  $Q$  elements each, and  $j$  blank blocks, containing  $C-k$  elements each. The right side was obtained by multiplying the number of rows and the number of columns in table  $P$ . So table  $P$  has  $(Q+C-k) \cdot j / C$  rows. We have  $j = C \cdot n_{\text{cycles}} / (Q-k)$  and by replacing this in the previous formula, the number of rows can be written like this:  $(Q+C-k) \cdot (C \cdot n_{\text{cycles}} / (Q-k)) / C = (Q+C-k) \cdot n_{\text{cycles}} / (Q-k)$ .

For each  $E_{ic}$ ,  $i$  from 1 to  $j$  and  $c$  from 1 to  $C-k$ , we could modify matrix  $P$  like this: we set  $P[\text{row}(E_{ic}), \text{col}(E_{ic})] = 0$  (corresponding to the empty/blank elements) and we set the remaining elements  $P[\text{row}(N_{ic}), \text{col}(N_{ic})] = 1$  (corresponding to the non-empty/non-blank elements). We calculate the sums of all the elements for all the columns of  $P$  and if the sums are equal then our sought distribution is balanced, i.e. all elements 1, 2, ...,  $C$

appear an equal number of times in the first row/column of the basic grid, therefore they appear an equal number of times in the whole basic grid. The sums of the elements in column  $c$  give the number of times channel  $ch=c$  appears in the final distribution.

#### 4.5 Algorithm for Channel Assignment in a Grid

The channels assigned to the nodes from the first row/column in our channel assignment algorithm are given by the columns for the non-empty/non-blank cells  $N_{ic}$  which can be calculated like this:

$$\text{col}(N_{i1}) = 1 + ((\text{index}(N_{i1}) - 1) \bmod C)$$

$$\text{col}(N_{i2}) = 1 + ((\text{index}(N_{i2}) - 1) \bmod C)$$

...

$$\text{col}(N_{iQ}) = 1 + ((\text{index}(N_{iQ}) - 1) \bmod C),$$

where  $i$  takes values from 1 to  $j$  and  $\text{index}(N_{ic}) = c + (Q + C - k)(i - 1)$ .

Consider the grid where the assignment starts from node<sub>11</sub>. To find the channels assigned to an arbitrary node node<sub>xy</sub> we first find the indices  $ro, co$  in the basic grid it belongs to:  $ro = 1 + ((x - 1) \bmod j)$  and  $co = 1 + ((y - 1) \bmod j)$ . Then, we find the assignment for node<sub>ro, co</sub> in the basic grid like this:

1. We find  $w = 1 + ((ro + co - 1 - 1) \bmod j) = 1 + ((ro + co - 2) \bmod j)$ , where  $ro + co - 2$  represents the Manhattan distance between the top leftmost node of our current basic grid and node<sub>ro, co</sub> in our current basic grid, which is the basic grid corresponding to our initial node, node<sub>xy</sub>.

The indices  $ro$ ,  $co$  are considered in this current basic grid while the indices  $x$ ,  $y$  are considered in the initial (bigger) grid.

2. We first find the basic grid assignment for node $_{ro, co}$  using the formulas for the column index in the matrix P:

$$\text{col}(N_{w1}) = 1 + ((\text{index}(N_{w1}) - 1) \bmod C)$$

$$\text{col}(N_{w2}) = 1 + ((\text{index}(N_{w2}) - 1) \bmod C)$$

...

$$\text{col}(N_{wQ}) = 1 + ((\text{index}(N_{wQ}) - 1) \bmod C),$$

where  $w$  takes values from 1 to  $j$  and  $\text{index}(N_{wc}) = c + (Q + C - k)(w - 1)$ .

The channels assigned to an edge incident to this node, node $_{ro, co}$ , that has this node on the left or above, are the last  $k$  channels of this assignment, i.e. the second index takes the values  $Q - k + 1$ ,  $Q - k + 2$ , ...,  $Q$ :

$$\text{col}(N_{w, Q-k+1}) = 1 + ((\text{index}(N_{w, Q-k+1}) - 1) \bmod C)$$

$$\text{col}(N_{w, Q-k+2}) = 1 + ((\text{index}(N_{w, Q-k+2}) - 1) \bmod C)$$

...

$$\text{col}(N_{w, Q}) = 1 + ((\text{index}(N_{w, Q}) - 1) \bmod C),$$

where  $w$  takes values from 1 to  $j$  and  $\text{index}(N_{wc}) = c + (Q + C - k)(w - 1)$ .

For the horizontal edges we consider that each edge takes the last  $k$  channels of the node to its left. For the vertical edges we consider that each edge takes the last  $k$  channels of the node above it.

From Figure 4.17 we observe some patterns in our channel assignment.

Q-1, Q	x, y	z, t
Q-1, Q	x, y	z, t
x, y	z, t	
x, y	z, t	
z, t		
z, t		

Figure 4.17: Patterns of assigned channels for  $k=2$

The tables that give the values for  $j$  and  $n_{\text{cycles}}$  could be used to choose the best values for  $C$ ,  $Q$  and  $k$ . The values that are displayed in black in the tables from Figure 4.4 correspond to  $C$ ,  $Q$  and  $k$  that give perfectly balanced distributions for channels 1, 2, ...,  $C$  in a grid with dimension  $j \times j$ , where  $j$  is the first element displayed in the tables. The second element is  $n_{\text{cycles}}$ , i.e. the number of times we cycle through all the elements from 1 to  $C$  when constructing the first row/column of vectors of the basic grid, each vector representing the channels assigned to a node/device.

The colored values correspond to imbalanced distributions. The red ones are those that satisfy the conditions  $Q-2k > 0$ ,  $n_{\text{cycles}}=1$  (which is equivalent to  $Q-k$  divides  $C$ ) while the green ones satisfy the conditions  $Q-2k < 0$ ,  $n_{\text{cycles}}=1$  (which is equivalent to  $Q-k$  divides

C) and  $Q-k$  does not divide  $Q$ . The blue cells are those for which  $n_{\text{cycles}} > 1$  (which is equivalent to  $Q-k$  does not divide  $C$ ) and  $C$  does not divide  $j \cdot k$ .

We gave a method that uses matrix  $P$  to determine cases that give balanced distributions. This method helps us to better understand rules and patterns related to our channel assignment algorithm. Due to the high number of cases obtained from varying  $C$ ,  $Q$  and  $k$ , a theoretical analysis of the algorithm was necessary.

#### 4.6 Analysis of Interference

Next we will add a second criterion that could be used to make decisions regarding the best values for  $C$ ,  $Q$  and  $k$  for our algorithm that provides robustness to PUs which could reclaim up to  $k-1$  channels simultaneously.

In Figure 4.18, we suppose that the node in the upper left corner of the grid, marked with an empty circle, has channel assignment  $(1, 2, 3, \dots, Q)$  and the rest of the nodes are assigned using our algorithm. The sensors that are in the interference range of node  $u$  are marked in red. Let us consider that  $u$  communicates with  $v$  (bidirectional communication). We add the edge  $e = (u, v)$  and as a result we add additional nodes, marked in blue, that are in the interference range of  $q$ .

In [10] the authors used the following formula to calculate the potential interference index for an edge  $e$ :  $p(e) = |\{(u', v') : (u', v') \in E, u' \text{ or } v' \in D(e)\}|$ .  $D(e)$  represents the set of nodes in the interference range of link  $e = (u, v)$ , i.e.  $D(e) = D(u) \cup D(v)$ .

We analyze interference just for horizontal edges. Similar results could be obtained for vertical edges since our assignment matrix  $A$  is symmetric, see Figure 4.2. Suppose that the assignment for the edge  $e$  is  $e_0 = (a_1, a_2, a_3, \dots, a_k)$ . In Figure 4.19 we marked

with labels the horizontal edges that are in the interference range of  $e$ . All the edges marked with label 0 have the same assignment as edge  $e$ . Next we use an index for the vector of ones, denoted by  $J$ , index which represents its length, i.e.  $J_k$  has length  $k$ .

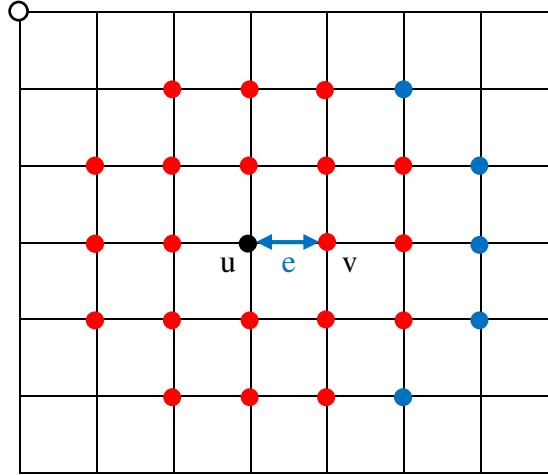


Figure 4.18: Sensors within interference range of node  $e$

In general, the edges marked with label  $i$  have the assignment  $e_i = J_k + ((e_0 + (i(Q-k) - 1)J_k) \bmod C)$ , where  $J_k = (1,1,1,\dots,1)$  and  $i$  can take any integer value from  $-4$  to  $4$ .

For example, in Figure 4.20 we marked in red the channels assigned to edges in a representation using matrix  $P$ . We colored the last assignment too since we consider our configuration as part of a larger network. We have  $e_0 = (5,6)$ ,  $e_1 = J_2 + ((e_0 + (1(Q-k) - 1)J_2) \bmod C) = J_2 + (((5,6) + 3 \cdot J_2) \bmod 10) = J_2 + ((8,9) \bmod 10) = (9,10)$ ,  $e_2 = J_2 + ((e_0 + (2(Q-k) - 1)J_2) \bmod C) = J_2 + (((5,6) + 7 \cdot J_2) \bmod 10) = (3,4)$ . In Figure 4.19, edges marked with 0 have the assignment  $e_0 = (5,6)$ , edges marked with 1 have the assignment  $e_1 = (9,10)$  etc.

Similar calculations can be applied to the algorithm that assigns channels to nodes. In the next section we present an alternative version of this algorithm using such calculations.



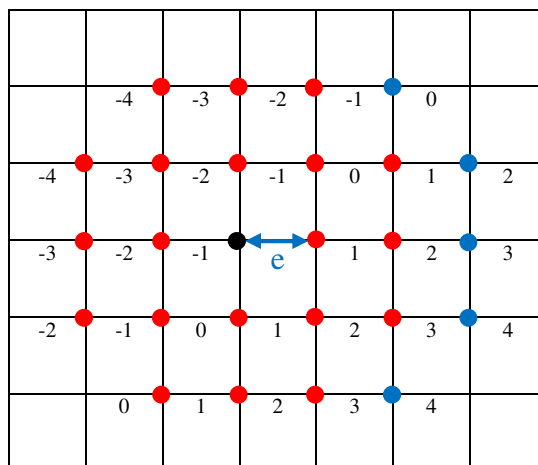


Figure 4.19: Horizontal edges within interference range of edge e

1	2	3	4	5	6				
				5	6	7	8	9	10
								9	10
1	2	3	4						
		3	4	5	6	7	8		
						7	8	9	10
1	2								

Figure 4.20: Table P for  $Q-2k < 0$ ;  $C=10$ ,  $Q=6$ ,  $k=2$ ,  $j=5$

We calculate the potential interference index of the edge e in the center for several values of C, Q and k, first by considering an edge as one link with k channels assigned to it, and second, by considering that an edge consists of k links, each link having one of the k channels (common to the end nodes) assigned to it. In the first case, if an edge that contains a node marked in red or blue has at least one channel in common with the edge e, we add 1 to the potential interference index of edge e. In the second case, if an edge that contains a node marked in red or blue has t links in common with the edge e, we add t (instead of 1) to the potential interference index of edge e.

The tables in Figure 4.21 give the values of  $p(e)$  (first case) and  $p'(e)$  (second case) for several choices of  $C$ ,  $Q$  and  $k$ . The value of  $p'(e)$  is included in parentheses. The colored results correspond to imbalanced distributions of channels while the black ones represent balanced distributions. The values for  $C$ ,  $Q$  and  $k$  that correspond to the smallest black values are the best ones to choose as parameters for the algorithm.

C\Q	2	3	4	5	6	7	8	9	10
3	10 (10)								
4	8 (8)	16 (16)							
5	4 (4)	4 (4)	4 (4)						
6	4 (4)	10 (10)	16 (16)	10 (10)					
7	4 (4)	4 (4)	4 (4)	4 (4)	4 (4)				
8	4 (4)	8 (8)	4 (4)	16 (16)	4 (4)	8 (8)			
9	4 (4)	4 (4)	10 (10)	4 (4)	4 (4)	10 (10)	4 (4)		
10	4 (4)	4 (4)	4 (4)	4 (4)	16 (16)	4 (4)	4 (4)	4 (4)	
11	4 (4)	4 (4)	4 (4)	4 (4)	4 (4)	4 (4)	4 (4)	4 (4)	4 (4)

k=1

C\Q	3	4	5	6	7	8	9	10
4	22 (30)							
5	16 (20)	18 (22)						
6	12 (16)	10 (20)	16 (32)					
7	12 (16)	14 (18)	12 (16)	12 (16)				
8	12 (16)	8 (16)	10 (14)	16 (32)	10 (14)			
9	12 (16)	8 (12)	10 (20)	12 (16)	12 (16)	10 (20)		
10	12 (16)	4 (8)	10 (14)	4 (8)	16 (32)	4 (8)	10 (14)	
11	12 (16)	4 (8)	8 (12)	10 (14)	12 (16)	12 (16)	10 (14)	8 (12)

k=2

C\Q	4	5	6	7	8	9	10
5	30 (50)						
6	24 (40)	30 (50)					
7	20 (36)	22 (40)	22 (38)				
8	20 (36)	22 (38)	18 (32)	16 (48)			
9	20 (36)	16 (28)	10 (30)	16 (32)	16 (32)		
10	20 (36)	16 (24)	14 (28)	18 (26)	16 (48)	18 (26)	
11	20 (36)	12 (20)	14 (26)	10 (24)	16 (32)	16 (32)	10 (24)

k=3

C\Q	5	6	7	8	9	10
6	30 (76)					
7	30 (66)	30 (70)				
8	26 (62)	22 (60)	26 (58)			
9	26 (62)	22 (50)	30 (60)	22 (54)		
10	26 (62)	16 (40)	22 (50)	18 (44)	16 (64)	
11	26 (62)	16 (36)	22 (48)	18 (42)	16 (48)	16 (48)

k=4

C\Q	6	7	8	9	10
7	30 (106)				
8	30 (96)	30 (98)			
9	30 (92)	30 (80)	30 (90)		
10	30 (92)	30 (70)	30 (80)	30 (74)	
11	30 (92)	24 (60)	22 (70)	26 (68)	22 (70)

k=5

C\Q	7	8	9	10
8	30 (136)			
9	30 (126)	30 (118)		
10	30 (122)	30 (100)	30 (110)	
11	30 (122)	30 (90)	30 (100)	30 (98)

k=6

C\Q	8	9	10
9	30 (166)		
10	30 (156)	30 (144)	
11	30 (152)	30 (126)	30 (138)

k=7

C\Q	9	10
10	30 (196)	
11	30 (186)	30 (170)

k=8

C\Q	10
11	30 (226)

k=9

Figure 4.21: Tables with values of  $p(e)$  and  $p'(e)$  for specific values of C, Q and k

#### 4.7 Another Version for Channel Assignment Algorithm in a Grid

Suppose that the vertex/node from the upper left corner of the grid, denoted by  $v_0$ , has channel assignment  $v_0 = (1, 2, 3, \dots, Q)$ . Then the edges adjacent to it (right and bottom) have channel assignment  $e_0 = (Q-k+1, Q-k+2, \dots, Q)$ .

Based on GPS information, each representative assigns channels to its  $Q$  radios and then each horizontal edge gets the last  $k$  channels of the node to its left, and each vertical edge gets the last  $k$  channels of the node above it. The assignments will repeat since we can consider the network as basic grids put together side by side, horizontally and vertically.

We first consider channel assignment for the nodes of a basic grid of dimension  $j \times j$ . The channels assigned to the nodes from the first row/column with our channel assignment algorithm are given by  $v_i = J_Q + ((v_0 + (i(Q-k)-1)J_Q) \bmod C)$ , where  $J_Q = (1,1,1,\dots,1)$  and  $i$  takes integer values from 0 to  $j-1$ , in increasing order. The channels assigned to the edges from the first row/column in our channel assignment algorithm are given by  $e_i = J_k + ((e_0 + (i(Q-k)-1)J_k) \bmod C)$ , where  $J_k = (1,1,1,\dots,1)$  and  $i$  takes integer values starting from 0.

In what follows, the indices  $row$ ,  $col$  are considered in the current basic grid while the indices  $x$ ,  $y$  are considered in the whole network, which we suppose is a bigger grid. All indices start from 1 and they represent the coordinates of the representatives' cells in the matrix/grid of cells. Since the devices have GPS or other localization protocols [53], they are able to find  $x$  and  $y$ . Based on these, they can calculate indices  $row$  and  $col$ , and then assign their channels accordingly. Each node assigns its channels using the following algorithm.

Suppose that the left corner of each basic grid has assignment  $v_0$ . In previous examples we considered  $v_0 = (1,2,3,\dots,Q)$  but  $v_0$  could be any assignment that contains  $k$  different channels.

To find its assignment, an arbitrary node denoted by  $\text{node}_{xy}$  calculates the indices  $\text{row}$ ,  $\text{col}$  in the basic grid it belongs to:  $\text{row} = 1 + ((x-1) \bmod j)$  and  $\text{col} = 1 + ((y-1) \bmod j)$ . Then, it finds the assignment for  $\text{node}_{\text{row}, \text{col}}$  in the basic grid like this:

1. Calculates  $w = 1 + ((\text{row} + \text{col} - 2) \bmod j)$ , where  $\text{row} + \text{col} - 2$  represents the Manhattan distance between the top leftmost node of its basic grid (or  $\text{node}_{11}$ ) and  $\text{node}_{\text{row}, \text{col}}$  in its basic grid, which is the basic grid corresponding to the node initially denoted by  $\text{node}_{xy}$ .

2. Finds the basic grid channel assignment for  $\text{node}_{\text{row}, \text{col}}$  using the formula  $v_i = J_Q + ((v_0 + (w(Q-k)-1)J_Q) \bmod C)$ , where  $J_Q = (1,1,1,\dots,1)$ .

Suppose that each node  $\text{node}_{xy}$  in the network assigns its  $Q$  channels using the previous algorithm. Then an edge incident to  $\text{node}_{xy}$  that has this node on the left or above it, uses for communication the last  $k$  channels of this assignment. In other words, each horizontal edge uses the last  $k$  channels of the node to its left and each vertical edge uses the last  $k$  channels of the node above it.

## 4.8 Conclusions

We introduced and analyzed a channel assignment algorithm which is robust to PUs that could reclaim multiple channels at once and which can provide a balanced distribution of channels for certain parameters. We calculated the potential interference

index for different values of C, Q and k by considering that cell representatives form perfect grids, i.e. they are placed on a grid at distance d, and their range is  $r=d\sqrt{5}$ .

When making decisions about choosing the best values for C, Q and k, we should take into account these results related to our channel assignment method. If we have flexibility in choosing the values for C, Q and k, we can avoid the cases which give imbalanced distributions and the ones that have a high probability of interference.

We could use Figure 4.21 to choose the best values for C, Q and k, by giving priority to black values (balanced distribution) that have small numbers (small interference). Depending on our goals and on the flexibility in choosing the values for C, Q and k, we could take into account both criteria (balanced distribution and small interference) or just one of them, in order to maximize the performance of the algorithm.

This channel assignment algorithm is relevant because it could be applied to different kinds of networks, like mesh networks or WSNs with limited resources. In comparison to other algorithms, each device can assign channels to its radios based on simple calculations and this provides energy efficiency.

## 5. EFFICIENT WIRELESS COMMUNICATION IN GRID NETWORKS

### 5.1 Introduction and Problem Definition

With a single radio and a single channel, it is difficult to provide reliable and timely communication in case of high data rate requirements because of collisions and limited bandwidth. Therefore, designing multi-channel based communication protocols is essential for improving the network throughput and providing quality communication services.

In this chapter we give lower and upper bounds for capacity in grid networks when using one channel and one radio per node. We also analyze capacity for multiple channels and show how it improves compared to the case with one channel. The bounds obtained depend on the range of communication, the distance between nodes and the size of the grid network. Using our communication pattern, we can schedule node communication to ensure end-to-end communication between any two nodes in a grid network while maximizing capacity.

### 5.2 Capacity for Unidirectional Communication

#### *Capacity for One Channel*

We consider an  $s \times s$  grid with  $s^2$  nodes (or  $(s-1) \times (s-1)$  cells). Suppose  $d$  is the size of a grid cell. By an *active edge* we mean a communication pair sender-receiver. A *blank edge* refers to a pair of adjacent nodes that do not currently communicate. We consider

just horizontal edges in the analysis. We could repeat the same pattern with vertical edges; see the last two sections for details.

The formulas for capacity were obtained by placing non-interfering horizontal active edges of length  $d$ , starting with the upper border of the grid, with the first horizontal active edge starting at the upper left corner. We suppose that the communication range is  $r$  and  $r \geq d$ .

In Figure 5.1 we represented a portion of the top border of a grid, starting from the left and continuing further to the right. The edges marked with  $d$  represent non-interfering active edges that could use the same channel. In this chapter, by row/column we mean a row/column of cells (small squares) and by line we mean a line of horizontal edges.

For unidirectional communication, in order to avoid collisions, the distance between the sender from an active edge and the receiver from another active edge has to be greater than  $r$ , for any such pair of nodes. As illustrated in Figure 5.3, to maximize capacity for unidirectional communication, we consider a scheduling with an alternating communication, e.g. nodes in the first marked column send from left to right, nodes in the second marked column send from right to left, and so on. In Figure 5.1 the horizontal distance between two adjacent horizontal active edges must be greater than  $r-d$ .

The number of blank columns/edges between two horizontally adjacent horizontal active edges, which we denote by  $nBC$ , is the smallest integer greater than  $\frac{r-d}{d} = \frac{r}{d} - 1$ , which is  $\left\lceil \frac{r}{d} - 1 + 1 \right\rceil = \left\lceil \frac{r}{d} \right\rceil$ , see Figure 5.1. So  $nBC = \left\lceil \frac{r}{d} \right\rceil$ .

We have  $s-1$  edges on the upper border so the length of the upper border is  $(s-1)d$ . We take out the first active edge from the left and the length becomes  $(s-1)d-d = (s-2)d$ . The number of additional active edges on the same channel that we can fit in the remaining



portion of the upper border can be obtained by dividing  $(s-2)d$  to the length of the segment AB from Figure 5.1, which is  $\lfloor \frac{r}{d} \rfloor d + d = \lfloor \frac{r}{d} + 1 \rfloor d$ , and taking the lower bound. We add 1 to account for the first active edge of length  $d$  that we removed, so the number of horizontal active edges in one line is  $1 + \lfloor \frac{s-2}{\lfloor \frac{r}{d} + 1 \rfloor} \rfloor$ .

In Figure 5.3 we marked with solid lines two vertically adjacent non-interfering horizontal active edges, of length  $d$ , that are on the same channel. We denote by  $y$  the number of rows (of cells) between them so the length of the vertical side is  $y \cdot d$ .

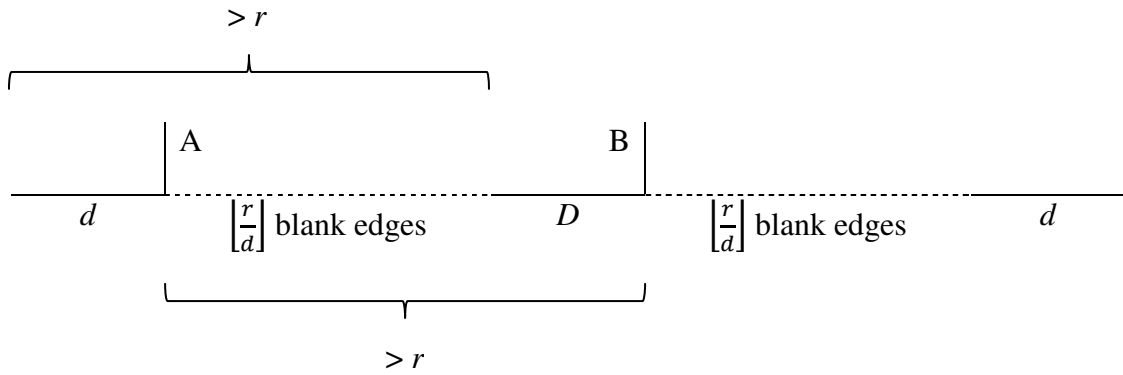


Figure 5.1: Top border of a grid (unidirectional communication)

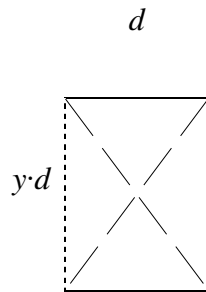


Figure 5.2: Non-interfering horizontal active edges

Since we consider the two horizontal active edges from Figure 5.2 to have the same direction of communication (see Figure 5.3), any of the two diagonals from Figure 5.2

has to be greater than  $r$  to avoid interference. The length of the diagonal is

$$\sqrt{(y \cdot d)^2 + d^2} > r \text{ or } y \cdot d > \sqrt{r^2 - d^2} \text{ or } y > \sqrt{\left(\frac{r}{d}\right)^2 - 1}.$$

So the vertical distance between two vertically adjacent horizontal active edges is greater than  $d\sqrt{\left(\frac{r}{d}\right)^2 - 1}$ . The number of rows between two vertically adjacent horizontal

active edges is the smallest integer greater than  $\sqrt{\left(\frac{r}{d}\right)^2 - 1}$ , which is  $\left\lceil \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rceil$ .

The number of blank horizontal lines between two vertically adjacent horizontal active

edges is one less:  $nBL = \left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} \right\rfloor$ .

We have  $s-1$  vertical edges on the left border so the length of the left border is  $(s-1)d$ .

We take out the first horizontal active edge from the top. The number of additional non-interfering horizontal active edges that we can fit in the remaining portion of the left border can be obtained by dividing its length to the vertical distance between two

vertically adjacent non-interfering horizontal active edges, which is  $\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor d$ ,

and taking the lower bound. We add 1 to account for the first active edge of length  $d$  that

we removed, so we obtain  $1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor$ . Therefore the number of non-interfering

horizontal active edges in one column is  $1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor$ .

Using our construction, the total number of active edges is:

$$\left(1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor\right) \left(1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor\right).$$

If we divide this by  $s^2$  (the number of nodes), we get the capacity:

$$\text{Capacity}_{lc} = \frac{\left(1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d+1} \right\rfloor} \right\rfloor\right) \left(1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor\right)}{s^2}.$$

In Table 5.1 we show the values of  $nBC$  and  $nBL$  for some values of  $r/d$ . In Figure 5.3 we present our construction for four cases. We placed non-interfering active edges on four grids that correspond to the first four cases from Table 5.1. On the same page, some formulas are given for each case.

	$r/d$	Number of blank columns $nBC = \left\lfloor \frac{r}{d} \right\rfloor$	Number of blank lines $nBL = \left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} \right\rfloor$
$r=d$	1	1	0
$r=d\sqrt{2}$	1.414214	1	1
$r=2d$	2	2	1
$r=2d\sqrt{2}$	2.828427	2	2
$r=3d$	3	3	2
$r=4d$	4	4	3
$r=3d\sqrt{2}$	4.242641	4	4
$r=5d$	5	5	4
$r=4d\sqrt{2}$	5.656854	5	5
$r=6d$	6	6	5
$r=7d$	7	7	6
$r=5d\sqrt{2}$	7.071068	7	7

Table 5.1: Values of  $nBC$  and  $nBL$  for some values of  $r/d$

### Examples

1) For  $r = d$ ,  $\frac{r}{d} = 1$

The number of horizontal active edges in one line:

$$1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor = 1 + \left\lfloor \frac{s-2}{2} \right\rfloor$$

The number of horizontal active edges in one column:

$$1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor = 1 + \left\lfloor \frac{s-1}{1} \right\rfloor = s$$

2) For  $r = d\sqrt{2}$ ,  $\frac{r}{d} = \sqrt{2} = 1.4142$

The number of horizontal active edges in one line:

$$1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor = 1 + \left\lfloor \frac{s-2}{2} \right\rfloor$$

The number of horizontal active edges in one column:

$$1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor = 1 + \left\lfloor \frac{s-1}{2} \right\rfloor$$

3) For  $r = 2d$ ,  $\frac{r}{d} = 2$

The number of horizontal active edges in one line:

$$1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor = 1 + \left\lfloor \frac{s-2}{3} \right\rfloor$$

The number of horizontal active edges in one column:

$$1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor = 1 + \left\lfloor \frac{s-1}{2} \right\rfloor$$

4) For  $r = 2d\sqrt{2}$ ,  $\frac{r}{d} = 2\sqrt{2} = 2.8284$

The number of horizontal active edges in one line:

$$\left(1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor\right) = \left(1 + \left\lfloor \frac{s-2}{3} \right\rfloor\right)$$

The number of horizontal active edges in one column:

$$1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor = 1 + \left\lfloor \frac{s-1}{3} \right\rfloor$$

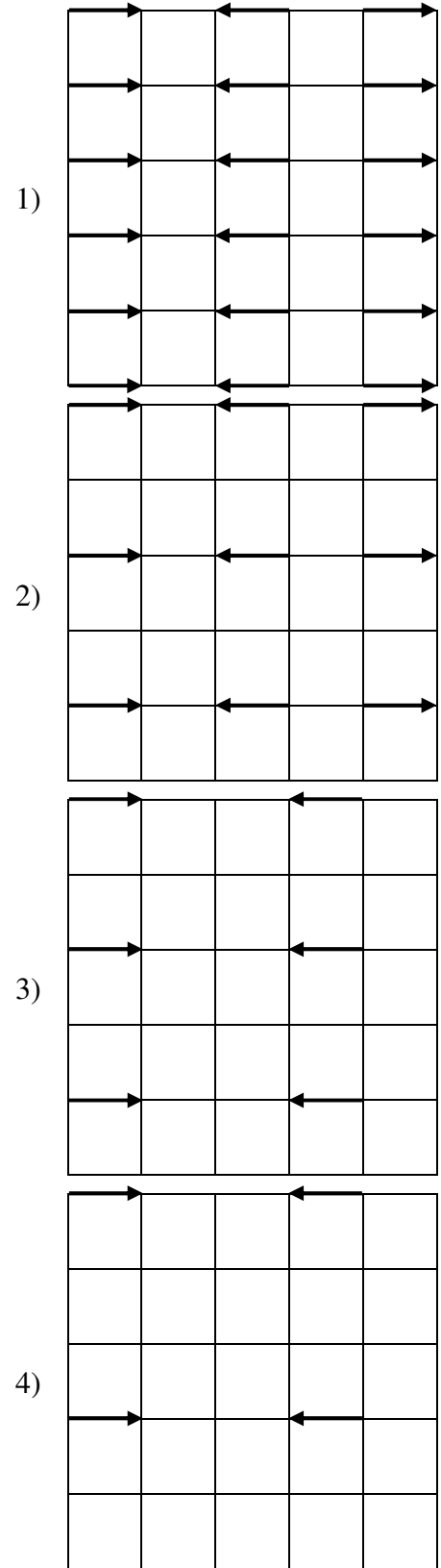


Figure 5.3: Non-interfering active edges for four cases

For our algorithm presented in [52], where  $d=r/\sqrt{5}$ , we use the fourth case above and we get the following approximation for capacity when using one radio and one channel:

$$\text{Capacity}_{1c} = \frac{\left(1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor\right) \left(1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor\right)}{s^2} = \frac{\left(1 + \left\lfloor \frac{s-2}{3} \right\rfloor\right) \left(1 + \left\lfloor \frac{s-1}{3} \right\rfloor\right)}{s^2}.$$

### *Capacity for Multiple Channels*

For more channels, we use the blank lines and blank columns between the active edges from the case  $m=1$  (one channel). We could fill the blank lines and blank columns in order to increase capacity, when multiple channels are available.

We first focus on filling the blank lines for the columns that we filled in our construction for  $m=1$ . For one radio, if the number of additional channels that we want to add to the case for one channel is smaller or equal to the number of blank lines ( $nBL$ ), we could fill the blank lines with the same pattern using the additional channels, and the capacity will increase by the total number of assigned channels, on average:

$$\text{Capacity}_{mc} = m \text{ Capacity}_{1c} = m \frac{\left(1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor\right) \left(1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor\right)}{s^2}.$$

The maximum value for  $\text{Capacity}_{mc}$  is obtained when  $m=1+nBL$ , i.e. when there is a different channel available for each line.

Next we focus on filling the blank columns:

- For the cases where the number of blank columns ( $nBC$ ) is 3 or 4, we could intercalate additional channels on columns, up to  $1+nBC$ , so the capacity of the network could double:  $2 \text{ Capacity}_{mc}$ .

- For the cases where the number of blank columns ( $nBC$ ) is 5 or 6, we could intercalate additional channels on columns, up to  $1+nBL$ , so the capacity of the network could triple:  $3 \text{ Capacity}_{mc}$ .

...

- For the cases where the number of blank columns ( $nBC$ ) is  $2p-1$  or  $2p$ , we could intercalate additional channels on columns, up to  $1+nBL$ , so the capacity of the network could increase  $p$  times:  $p \text{ Capacity}_{mc}$ .

Some special cases are shown in Figure 5.4. In the first grid,  $r=d\sqrt{2}$  and  $nBL=1$ . Figure 5.4a shows how we can intercalate on lines active edges that use a new channel, marked with dotted lines. In the second grid,  $r=3d$  and  $nBC=3$ . Figure 5.4b shows how we can intercalate on columns active edges that use a new channel, marked with dashed lines.

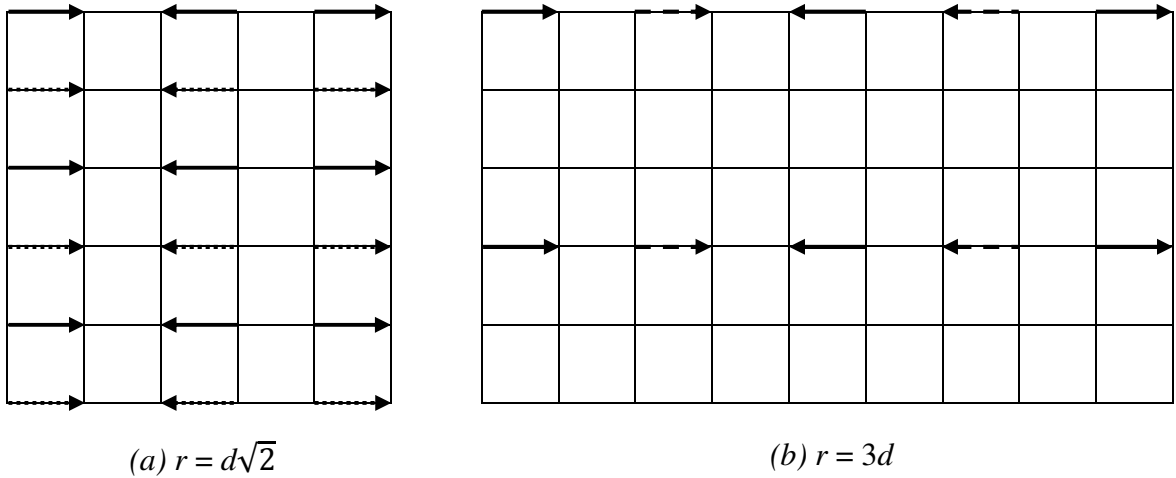


Figure 5.4: Intercalating active edges for a new channel

If we combine both methods, in the second grid we can triple the capacity by filling the blank lines for the columns marked in solid black with 2 extra channels and we can

double this capacity or get a capacity 6 times the initial one (for one channel) if, in addition, we fill the columns marked with dashed lines with three channels that are different from the ones used for the columns marked in solid black.

For this, we need to have at least 6 different channels available. We cannot have two arrows that start or end at the same node since a radio can either transmit or receive at one point in time.

Therefore, if enough channels are available, the capacity could increase by

$$n = (1+nBL) \left(1 + \left\lfloor \frac{nBC-1}{2} \right\rfloor\right) = \left(1 + \left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} \right\rfloor\right) \left(1 + \left\lfloor \frac{\lfloor r \rfloor - 1}{2} \right\rfloor\right)$$

when this quantity is less than or equal to  $m$ , the number of different channels that are available. In our previous example  $n=6$  and we assumed  $n \leq m$ . Other formulas could be derived for cases where  $n > m$  or when there are multiple radios per node.

The formula we found with our construction is a lower bound on capacity. For one channel we have the following lower bound:

$$\text{Capacity}_{\text{LB, 1}} = \frac{\left(1 + \left\lfloor \frac{s-2}{\lfloor \frac{r}{d+1} \rfloor} \right\rfloor\right) \left(1 + \left\lfloor \frac{s-1}{\left\lfloor \sqrt{\left(\frac{r}{d}\right)^2 - 1} + 1 \right\rfloor} \right\rfloor\right)}{s^2}.$$

An upper bound for one channel could be found by counting the maximum number of points that can be placed in an  $s \times s$  grid with  $s^2$  nodes (or  $(s-1) \times (s-1)$  cells), such that the distance between any point and its reachable neighboring points is  $r$ . We consider an active edge reduced to a node with negligible size or point. We also eliminate the restriction that the points (former active edges) are on the grid or at the intersection of the grid lines.

We construct equilateral triangles drawn downward, with side  $r$ , with the bases on the upper side of the grid, starting at the upper left corner. Between these triangles we construct other equilateral triangles etc. The vertices of the triangles are our points. A line of points is shifted horizontally by  $r/2$  compared to the line of points above it so we obtain a tiling (tessellation) of the square grid using equilateral triangles.

The maximum number of points that can be placed on the upper side of the grid is  $\left\lfloor \frac{(s-1)d}{r} \right\rfloor = \left\lfloor \frac{s-1}{\frac{r}{d}} \right\rfloor$ . On the next line we could have at most the same number of points,  $\left\lfloor \frac{s-1}{\frac{r}{d}} \right\rfloor$ . The distance between two horizontal lines that contain the points is equal to the height of the equilateral triangles, which is  $\frac{r\sqrt{3}}{2}$ . The maximum number of lines that could

$$\text{be formed is } 1 + \left\lfloor \frac{(s-1)d}{\frac{r\sqrt{3}}{2}} \right\rfloor = 1 + \left\lfloor \frac{s-1}{\frac{\sqrt{3}}{2} \cdot \frac{r}{d}} \right\rfloor.$$

So, by reducing the active edges to points, without the restriction that these points are on the grid or at the intersection of the grid lines, we get an upper bound for capacity:

$$\text{Capacity}_{\text{UB}, 1} = \min \left( 1, \frac{\left( \left\lfloor \frac{s-1}{\frac{r}{d}} \right\rfloor \right) \left( 1 + \left\lfloor \frac{s-1}{\frac{\sqrt{3}}{2} \cdot \frac{r}{d}} \right\rfloor \right)}{s^2} \right).$$

If we have  $m$  channels and at least  $m$  radios, then this upper bound can be multiplied by  $m$ :

$$\text{Capacity}_{\text{UB}, m} = m \cdot \min \left( 1, \frac{\left( \left\lfloor \frac{s-1}{\frac{r}{d}} \right\rfloor \right) \left( 1 + \left\lfloor \frac{s-1}{\frac{\sqrt{3}}{2} \cdot \frac{r}{d}} \right\rfloor \right)}{s^2} \right).$$



Tables 2-4 give some values for  $\text{Capacity}_{\text{LB}, 1}$  and  $\text{Capacity}_{\text{UB}, 1}$  for  $s = 100, 1000, 10000$ . We added a column for  $1/(r/d)^2$ , which can be used to approximate the capacity when  $s$  goes to infinity. We observe that the value for this approximation is between the lower and the upper bound.

Given a network where the nodes are in an infinite grid (corners of squares with sides  $d$ ) and each node has range  $r \geq d$ , we find a formula that gives  $x=r/d$  as a function of  $N =$  the number of reachable neighbors. The inverse function is:

$$N(x) = 4(\lfloor x \rfloor + \lfloor \sqrt{x^2 - 1} \rfloor + \lfloor \sqrt{x^2 - 4} \rfloor + \lfloor \sqrt{x^2 - 9} \rfloor + \dots + \lfloor \sqrt{x^2 - \lfloor x \rfloor^2} \rfloor).$$

For example:  $N(1) = 4, N(\sqrt{2}) = 8, N(2) = 12, N(2\sqrt{2}) = 24$ .

To invert this function  $N(x)$  and get  $x$  as a function of  $N$ , we can just say that the sides are 1 and the range is  $x$ . So we can imagine a fixed node at the origin and  $N(x)$  is the number of nodes at a distance between 1 and  $x$  from it. If we imagine the picture of these nodes, it will look like a circle. The number of nodes is approximately  $N(x) \sim \pi x^2$  (the area of the circle) so we can invert to  $x \sim \sqrt{\frac{N}{\pi}}$ . This is an approximate formula.

We compare the values obtained using this approximation with the original values. For the values of  $x=r/d$  given in Tables 2-4, we get the following approximation errors:

-0.12837917, -0.18155512, 0.04558995, 0.30529448, 0.01458934, 0.09117990,

-0.12755272, -0.04626504, 0.12894761, 0.02917868, 0.13633748, -0.06542846,

or in percentages: -12.8379167, -12.8378818, 2.2794976, 10.7937903, 0.4863113,

2.2794976, -3.0064463, -0.9253009, 2.2794933, 0.4863113, 1.9476783, -0.9252982.

If  $r/d$  is not known, it could be approximated with  $\sqrt{\frac{N}{\pi}}$ , where  $N$  is the number of reachable neighbors. But this approximation is not always precise.

<i>Case</i>	$r/d$	Capacity <sub>LB, 1</sub>	Capacity <sub>UB, 1</sub>	$1/(r/d)^2$
$r=d$	1	0.5	1	1
$r=d\sqrt{2}$	1.414214	0.25	0.567	0.5
$r=2d$	2	0.165	0.2842	0.25
$r=2d\sqrt{2}$	2.828427	0.1122	0.1435	0.125
$r=3d$	3	0.085	0.1287	0.11111111
$r=4d$	4	0.05	0.0696	0.0625
$r=3d\sqrt{2}$	4.242641	0.04	0.0621	0.05555556
$r=5d$	5	0.034	0.0437	0.04
$r=4d\sqrt{2}$	5.656854	0.0289	0.0357	0.03125
$r=6d$	6	0.0255	0.032	0.02777778
$r=7d$	7	0.0195	0.0238	0.02040816
$r=5d\sqrt{2}$	7.071068	0.0169	0.0238	0.02

Table 5.2: Bounds for capacity when  $s = 100$

<i>Case</i>	$r/d$	Capacity <sub>LB, 1</sub>	Capacity <sub>UB, 1</sub>	$1/(r/d)^2$
$r=d$	1	0.5	1	1
$r=d\sqrt{2}$	1.414214	0.25	0.576096	0.5
$r=2d$	2	0.1665	0.287923	0.25
$r=2d\sqrt{2}$	2.828427	0.111222	0.144024	0.125
$r=3d$	3	0.0835	0.128205	0.11111111
$r=4d$	4	0.05	0.071961	0.0625
$r=3d\sqrt{2}$	4.242641	0.04	0.06392	0.05555556
$r=5d$	5	0.0334	0.045969	0.04
$r=4d\sqrt{2}$	5.656854	0.027889	0.035904	0.03125
$r=6d$	6	0.023881	0.032038	0.02777778
$r=7d$	7	0.017875	0.02343	0.02040816
$r=5d\sqrt{2}$	7.071068	0.015625	0.023124	0.02

Table 5.3: Bounds for capacity when  $s = 1000$

<i>Case</i>	$r/d$	Capacity <sub>LB, 1</sub>	Capacity <sub>UB, 1</sub>	$1/(r/d)^2$
$r=d$	1	0.5	1	1
$r=d\sqrt{2}$	1.414214	0.25	0.577266	0.5
$r=2d$	2	0.16665	0.288592	0.25
$r=2d\sqrt{2}$	2.828427	0.111122	0.144334	0.125
$r=3d$	3	0.08335	0.128287	0.11111111
$r=4d$	4	0.05	0.072146	0.0625
$r=3d\sqrt{2}$	4.242641	0.04	0.06413	0.05555556
$r=5d$	5	0.03334	0.046177	0.04
$r=4d\sqrt{2}$	5.656854	0.027789	0.036082	0.03125
$r=6d$	6	0.023821	0.032071	0.02777778
$r=7d$	7	0.017863	0.023562	0.02040816
$r=5d\sqrt{2}$	7.071068	0.015625	0.023091	0.02

Table 5.4: Bounds for capacity when  $s = 10000$

### 5.3 Capacity for Bidirectional Communication

We consider an  $s \times s$  grid with  $s^2$  nodes (or  $(s-1) \times (s-1)$  cells). Suppose  $d$  is the size of the grid. In Figure 5.5 we represented a portion of the top border of a grid, starting from the left and continuing further to the right. The edges marked with  $d$  represent non-interfering active edges that could use the same channel.

For bidirectional communication, in order to avoid collisions, the distance between a node from an active edge and a node from another active edge has to be greater than  $r$ , for any such pair of nodes. So the number of blank columns/edges between two horizontally adjacent horizontal active edges is the smallest integer greater than  $\frac{r}{d}$ , which is  $\left\lfloor \frac{r}{d} + 1 \right\rfloor$ , see Figure 5.5. So  $nBC = \left\lfloor \frac{r}{d} + 1 \right\rfloor$ .

We have  $s-1$  edges on the upper border so the length of the upper border is  $(s-1)d$ . We take out the first active edge from the left and the length becomes  $(s-1)d - d = (s-2)d$ . The number of additional active edges on the same channel that we can fit in the remaining portion of the upper border can be obtained by dividing  $(s-2)d$  to the length of the segment AB from Figure 5.5, which is  $\left\lfloor \frac{r}{d} + 1 \right\rfloor d + d = \left\lfloor \frac{r}{d} + 2 \right\rfloor d$ , and taking the lower bound. We add 1 to account for the first active edge of length  $d$  that we removed, so the number of horizontal active edges in one line is  $1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 2 \right\rfloor} \right\rfloor$ .

In Figure 5.6 we marked with solid lines two vertically adjacent non-interfering horizontal active edges, of length  $d$ , that are on the same channel. We denote by  $y$  the number of rows (of cells) between them so the length of the vertical side is  $y \cdot d$ . To avoid interference, we need  $y \cdot d > r$  or  $y > \frac{r}{d}$ . This ensures that the two diagonals are also greater than  $r$ .

So the vertical distance between two vertically adjacent non-interfering horizontal active edges is greater than  $r$ . The number of rows between them is the smallest integer greater than  $\frac{r}{d}$ , which is  $\left\lfloor \frac{r}{d} + 1 \right\rfloor$ . The number of blank horizontal lines between two vertically adjacent non-interfering horizontal active edges is one less:  $nBL = \left\lfloor \frac{r}{d} \right\rfloor$ .

We have  $s-1$  vertical edges on the left border so the length of the left border is  $(s-1)d$ . We take out the first horizontal active edge from the top. The number of additional non-interfering horizontal active edges that we can fit in the remaining portion of the left border can be obtained by dividing its length to the vertical distance between two vertically adjacent non-interfering horizontal active edges, which is  $\left\lfloor \frac{r}{d} + 1 \right\rfloor d$ , and taking the lower bound. We add 1 to account for the first active edge of length  $d$  that we removed, so we obtain  $1 + \left\lfloor \frac{s-1}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor$ . Therefore, the number of non-interfering horizontal

active edges in one column is  $1 + \left\lfloor \frac{s-1}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor$ .

Using our construction, the total number of active edges is:

$$\left( 1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 2 \right\rfloor} \right\rfloor \right) \left( 1 + \left\lfloor \frac{s-1}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor \right).$$

If we divide this by  $s^2$  (the number of nodes), we get the capacity:

$$\text{Capacity}_{1c} = \frac{\left( 1 + \left\lfloor \frac{s-2}{\left\lfloor \frac{r}{d} + 2 \right\rfloor} \right\rfloor \right) \left( 1 + \left\lfloor \frac{s-1}{\left\lfloor \frac{r}{d} + 1 \right\rfloor} \right\rfloor \right)}{s^2}.$$

This capacity is smaller than the capacity for the unidirectional case since we increased the denominators of the two ratios in the numerator.

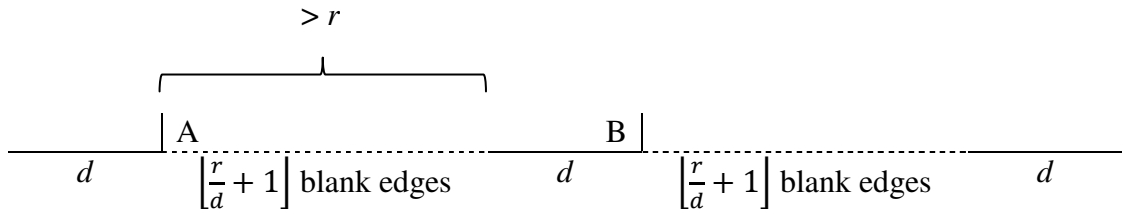


Figure 5.5: Top border of a grid (bidirectional communication)

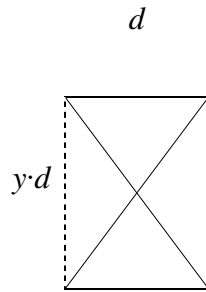


Figure 5.6: Non-interfering horizontal active edges

We could use the same method as for unidirectional communication, to fill the blank lines and blank columns, in order to increase capacity when multiple channels are available. Since in this case  $nBL = \lfloor \frac{r}{d} \rfloor$  and  $nBC = \lfloor \frac{r}{d} + 1 \rfloor$ , the capacity could increase by

$$n = (1+nBL) \left( 1 + \left\lfloor \frac{nBC-1}{2} \right\rfloor \right) = \left( 1 + \left\lfloor \frac{r}{d} \right\rfloor \right) \left( 1 + \left\lfloor \frac{\lfloor \frac{r}{d} + 1 \rfloor - 1}{2} \right\rfloor \right) = \left( 1 + \left\lfloor \frac{r}{d} \right\rfloor \right) \left( 1 + \left\lfloor \frac{\lfloor \frac{r}{d} \rfloor}{2} \right\rfloor \right)$$

when this quantity is less than or equal to  $m$ , the number of different channels that are available. Other formulas could be derived for cases where  $n > m$  or when there are multiple radios per node.

### Example of communication scheduling

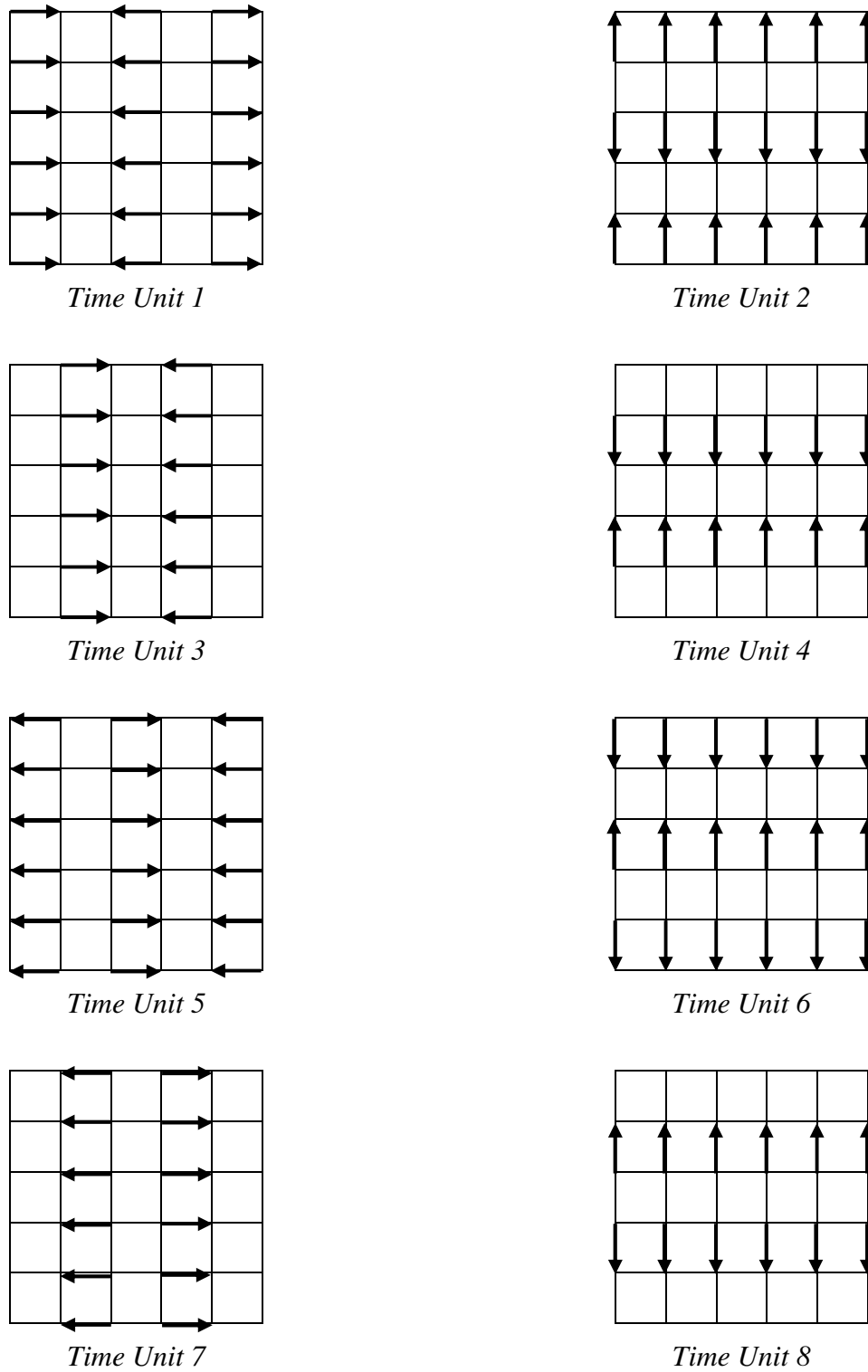


Figure 5.7: Time units for unidirectional communication, one radio and one channel

The previous formulas for the upper bound work for bidirectional communication too. They are given next. The second one requires  $m$  channels and at least  $m$  radios.

$$\text{Capacity}_{\text{UB}, 1} = \min \left( 1, \frac{\left( \left\lfloor \frac{s-1}{r} \right\rfloor \right) \left( 1 + \left\lfloor \frac{s-1}{\frac{\sqrt{3}}{2} \cdot \frac{r}{d}} \right\rfloor \right)}{s^2} \right),$$

$$\text{Capacity}_{\text{UB}, m} = m \cdot \min \left( 1, \frac{\left( \left\lfloor \frac{s-1}{r} \right\rfloor \right) \left( 1 + \left\lfloor \frac{s-1}{\frac{\sqrt{3}}{2} \cdot \frac{r}{d}} \right\rfloor \right)}{s^2} \right).$$

Using our communication pattern, we can schedule node communication to ensure end-to-end communication between any two nodes. The scheduling illustrated in Figure 5.7 corresponds to the communication pattern in Figure 5.3.1, where  $r=d$ . In this case of unidirectional communication with one radio and one channel, periods with 8 time units are needed. In Figure 5.7 we presented the patterns used for communication in each of these 8 time units. We alternated horizontal and vertical communication such that data can travel between any two nodes, in all directions.

#### 5.4 Formula for the Number of Time Units

The number of time units in a period, which we denote by  $nTU$ , can be calculated for both unidirectional communication and bidirectional communication, with formulas that use  $nBC$  and  $nBL$ . To derive the formula for unidirectional communication, we consider Figure 5.8 where  $r=3d$ . In Figure 5.8a we divided the grid into *slices* made of  $1+nBC=4$  columns, of width  $d(1+nBC)$ . We observe that if we start the column index from 1, the arrows are in the columns with indices  $1+k(1+nBC)$  where  $k \geq 0$ ,  $k$  integer. The arrows pointing to the right are in columns  $1+k(1+nBC)$  with  $k$  even and the arrows pointing to the left are in columns  $1+k(1+nBC)$  with  $k$  odd.

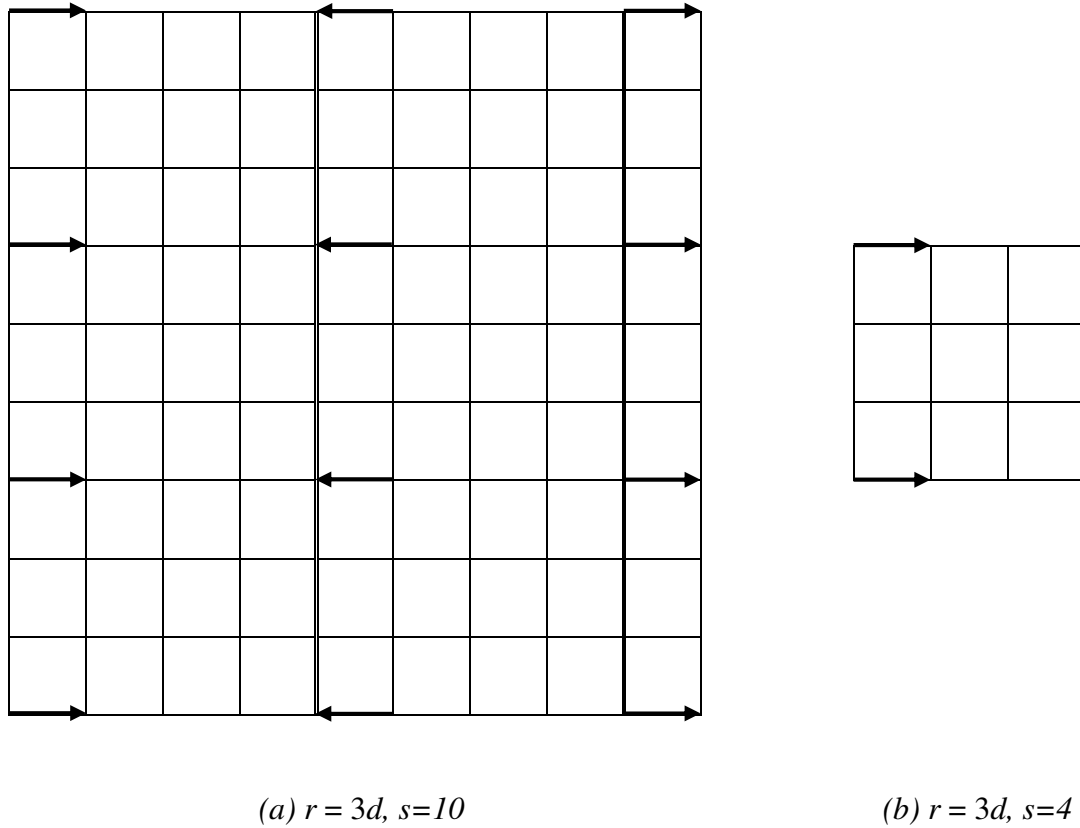


Figure 5.8: Non-interfering active edges for one channel

In what follows, we use *marked lines* to refer to the horizontal lines that have arrows on them. We start the line index from 1 so in Figure 5.8a the marked lines are the lines with indices 1, 4, 7 and 10. If we shift the pattern of arrows one column to the right (by  $d$ ), we need  $1+nBC$  time units to fill the marked horizontal lines with arrows, including the initial position. The odd numbered slices will be covered with arrows that point to the right and the even numbered slices will be covered with arrows that point to the left. If in addition we reverse the direction of each arrow in these  $1+nBC$  patterns (corresponding to the  $1+nBC$  time units), we have covered every edge from the marked lines with a right arrow and with a left arrow, using  $2(1+nBC)$  time units.



But we also have to cover the blank lines between the marked (already covered) horizontal lines, so each of the  $2(1+nBC)$  horizontal patterns has to be shifted down to cover  $nBL$  blank lines. If we consider the initial position also, we have  $1+nBL$  vertical shifts. Therefore the total number of time units for horizontal patterns is  $(1+nBL) \cdot 2(1+nBC) = 2(1+nBL)(1+nBC)$ .

We need the same number of time units to rotate these patterns by 90 degrees in order to cover all the vertical edges in the same way, with arrows pointing upward and downward, so in total we need  $nTU = 2 \cdot 2(1+nBL)(1+nBC) = 4(1+nBL)(1+nBC)$  time units.

For example, in Figure 5.7, corresponding to the case from Figure 5.3.1, we have  $nBL=0$ ,  $nBC=1$  so  $nTU = 4(1+nBL)(1+nBC) = 4(1+0)(1+1) = 8$ . For the case in Figure 5.3.2, we have  $nBL=1$ ,  $nBC=1$  so  $nTU = 4(1+nBL)(1+nBC) = 4(1+1)(1+1) = 16$ .

In order to get the formula for  $nTU$  we supposed that  $s-1 \geq 1+nBC$  and  $s \geq 1+nBL$ . In the first inequality  $s-1$  refers to the number of columns of edges in the grid while in the second inequality  $s$  refers to the number of horizontal lines in the grid. Recall that we use an  $s \times s$  grid with  $s^2$  nodes (or  $(s-1) \times (s-1)$  cells).

If one of the two inequalities is not true, i.e. if  $s-1 < 1+nBC$  or  $s < 1+nBL$ ,  $nTU$  is smaller:  $nTU = 4 \cdot \min(1+nBL, s) \cdot \min(1+nBC, s-1) < 4(1+nBL)(1+nBC)$ . Next, we explain this generalized formula by giving an example.

If we cut the upper left corner of Figure 5.8a, we obtain the grid in Figure 5.8b for which  $s=4$ . If we use the formula for  $nBC$  that we calculated in the section Capacity for Unidirectional Communication, we get  $nBC = \left\lfloor \frac{r}{d} \right\rfloor = 3$ . But since  $s-1 < 1+nBC$  we have just

2 blank columns that have to be filled with horizontal arrows as we can see in Figure 5.8b.

The number of horizontal shifts needed to fill the marked lines in the grid from Figure 5.8b is 2 times the number of columns, or  $2(s-1) = 6$ . Multiplication by 2 is needed because each edge in the marked lines has to be filled with a right arrow and with a left arrow in the corresponding number of time units, which in this case is  $2(s-1)$ . We have

$$nBL = \left\lceil \sqrt{\left(\frac{r}{d}\right)^2 - 1} \right\rceil = 2 \text{ and } s \geq 1+nBL \text{ so the number of vertical shifts for horizontal}$$

patterns does not change in Figure 5.8b compared to Figure 5.8a (it would change for  $s < 3$ ). We need  $1+nBL$  vertical shifts for each of the  $2(s-1)$  horizontal patterns. Therefore the total number of time units for horizontal patterns is  $(1+nBL) \cdot 2(s-1) = 2(1+nBL)(s-1)$ .

We need the same number of time units to rotate these patterns by 90 degrees in order to cover all the vertical edges in the same way, with arrows pointing upward and downward, so in total we need  $nTU = 2 \cdot 2(1+nBL)(s-1) = 4(1+nBL)(s-1)$  time units. The generalized formula gives us the same result:  $nTU = 4 \cdot \min(1+nBL, s) \cdot \min(1+nBC, s-1) = 4(1+nBL)(s-1) = 4(1+2)(4-1) = 36$ .

The formula is the same for bidirectional communication but the values of  $nBL$  and  $nBC$  are different and they can be calculated using the formulas from the section Capacity for Bidirectional Communication. Because we consider that the transmission of payload data is done in one direction at a time (even if exchanged messages go in both directions), we could imagine that we have arrows pointing in one direction but the distances between them are higher than for unidirectional communication. So the formula for bidirectional communication is also  $nTU = 4 \cdot \min(1+nBL, s) \cdot \min(1+nBC, s-1)$ . It is more likely that in practice the grids will be large, so  $s-1 \geq 1+nBC$  and  $s \geq 1+nBL$  will be both true. In such

cases  $nTU = 4(1+nBL)(1+nBC)$ . The values for  $nBL$  and  $nBC$  depend on the ratio  $r/d$  and they are different for unidirectional and bidirectional communication.

## 5.5 Conclusions

We established lower and upper bounds for capacity in grid networks and we showed how these bounds change when increasing the number of channels. The results are used to estimate the capacity in grid networks and to approximate the capacity of the representatives' communication in our algorithms that use cell representatives. These bounds depend on the range of communication, the distance between nodes and the size of the grid network. We also calculated the number of time units needed to ensure end-to-end communication between any two nodes while maximizing capacity. Formulas are given for both unidirectional and bidirectional communication. This analysis could be further extended for multi-radio grid networks.

## 6. CONCLUSIONS

Chapter 2 is a comprehensive survey focused on theoretical aspects and algorithms related to channel assignment in multi-radio multi-channel networks in general networks.

In Chapter 3 we introduced algorithms that are robust to the presence of PUs that could reclaim any of the assigned channels. In Chapter 4 we introduced a channel assignment algorithm that is robust to PUs that could reclaim more channels at once and can provide a balanced distribution of channels for some input parameters. We did a theoretical analysis in order to find reasons why specific cases give balanced distributions and we also analyzed interference.

When there is flexibility in choosing the network parameters, like number of channels, number of radios per node, and number of channels that could be reclaimed simultaneously by PUs, we could set them up such as to optimize the performance of the network, using one or both of these criteria: balanced distribution and minimum interference.

Due to the large number of cases obtained by varying  $C$ ,  $Q$  and  $k$ , results from simulations could not be generalized, so a theoretical analysis was necessary for our algorithm which provides robustness to PUs that could reclaim more than one channel at once.

Our scheduling method that maximizes capacity provides efficient communication in a grid network where data can travel in any direction with equal probability. The bounds

on capacity we established, presented in Chapter 5, depend on the size of the network and on  $r/d$ , the ratio between the range of the nodes and the distance between them. The number of time units needed to cover all edges of communication (both directions), also depending on  $r/d$ , helps to predict delays and to estimate the average time needed for data to arrive at the destination.

Our scheduling method, when applied to the generalized grid algorithm, could help in choosing input parameters  $C$ ,  $Q$  and  $k$  that would give the minimum number of time units necessary to cover the whole network. Optimizing some network parameters like the number of radios per device could help in designing efficient hardware, in order to improve the network performance when running algorithms.

The complexity of our generalized grid algorithm is much lower than that of other algorithms and, in addition, it provides robustness to PUs that could reclaim multiple channels at once. Existing algorithms do not address this last aspect. For certain input parameters our channel assignment is balanced and this is important in cases where there are limitations in channels' capacity.

As the creation of smaller, battery-powered devices is pursued, energy efficiency is of major importance. The development of new technologies and the increase in the amount of data that needs to be transmitted should be accompanied by the development of simpler algorithms that could be run on devices with limited computational capabilities. The advantages of such algorithms are observable especially in real-time environments.

Algorithms that work for networks with less restricted devices might not work in limited scenarios, so reducing complexity and providing energy efficiency are important goals when designing and analyzing algorithms oriented towards networks with limited

resources. These translate into lower delays when running applications or transmitting data and in a longer battery life, qualities that are directly related to customer satisfaction.

We provide efficiency by obtaining faster, simpler, robust, energy-efficient algorithms that could be used in real-time environments and for various types of networks, including networks with limited resources like WSNs.

### **Future Work**

The algorithm from Chapter 3 could be generalized to provide robustness to PUs that reclaim more than one channel at once. In this case the theorem could also be generalized. If we suppose that each device has  $Q$  radios, a node that receives two messages with assignments for  $2Q$  radios could choose the least used  $Q$  channels from among the received  $2Q$  channels. A node that receives one such message will assign all the received  $Q$  channels to its radios. In this case we have robustness to PUs that reclaim any  $Q-1$  channels.

The grid algorithm presented in Chapter 4 could be further explored in order to find mathematical formulas for interference, which would allow us to theoretically predict interference, without running algorithms.

Another direction for future work would be a scheduling algorithm for the channel assignment from Chapter 4. Communication patterns like those from Figure 5.7 could be derived for this channel assignment for specific sets of values for  $C$ ,  $Q$ ,  $k$ . The analysis of interference would play an important role in finding such patterns.

Other aspects that could be taken into account in future research are: variable number of radios for devices, multiple ranges of communication, uneven channel availability etc.

## BIBLIOGRAPHY

- [1] "ET Docket No 03-237 Notice of Proposed Rule Making and Order," 2003.
- [2] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White Space Networking with Wi-Fi like Connectivity," in *SIGCOMM '09: Proceedings of the ACM SIGCOMM Conference on Data Communication*, Barcelona, Spain, 2009, pp. 27-38.
- [3] K.-H. Phung, B. Lemmens, M. Mihaylov, L. Tran, and K. Steenhaut, "Adaptive Learning Based Scheduling in Multichannel Protocol for Energy-Efficient Data-Gathering Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, 2013.
- [4] J. Wu, Y. Dai, and Y. Zhao, "Effective Channel Assignments in Cognitive Radio Networks," *Computer Communications*, vol. 36, no. 4, pp. 411-420, 2013.
- [5] O. D. Incel, "A Survey on Multi-Channel Communication in Wireless Sensor Networks," *Computer Networks*, vol. 55, no. 13, pp. 3081-3099, 2011.
- [6] J. So and N. H. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver," in *MobiHoc '04: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Roppongi Hills, Tokyo, Japan, 2004, pp. 222-233.
- [7] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, "Component Based Channel Assignment in Single Radio, Multi-Channel Ad Hoc Networks," in *MobiCom '06: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, Los Angeles, CA, USA, 2006, pp. 378-389.
- [8] Y. F. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks," in *INFOCOM '08: Proceedings of the 27th IEEE International Conference on Computer Communications*, Phoenix, AZ, USA, 2008, pp. 1193-1201.

- [9] G. Xing et al., "Multi-Channel Interference Measurement and Modeling in Low-Power Wireless Networks," in *RTSS '09: Proceedings of the 30th IEEE Real-Time Systems Symposium*, Washington, DC, USA, 2009, pp. 248-257.
- [10] J. Zhao and G. Cao, "Robust Topology Control in Multi-Hop Cognitive Radio Networks," in *INFOCOM '12: Proceedings of the 31st IEEE International Conference on Computer Communications*, Orlando, FL, USA, 2012, pp. 2032-2040.
- [11] Q. Yu, J. Chen, Y. Fan, X. Shen, and Y. Sun, "Multi-Channel Assignment in Wireless Sensor Networks: A Game Theoretic Approach," in *INFOCOM '10: Proceedings of the 29th IEEE International Conference on Computer Communications*, San Diego, CA, USA, 2010, pp. 1-9.
- [12] G. Zhou et al., "MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks," in *INFOCOM '06: Proceedings of the 25th IEEE International Conference on Computer Communications*, Barcelona, Spain, 2006, pp. 1-13.
- [13] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An Energy-Efficient Multi-Channel MAC Protocol for Dense Wireless Sensor Networks," in *IPSN '08: Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, St. Louis, MO, USA, 2008, pp. 53-63.
- [14] L. Zhang, Q. Zhu, and A. Chen, "Fast Message Dissemination Tree and Balanced Data Collection Tree for Wireless Sensor Network," *Journal of Software*, vol. 8, no. 6, pp. 1346-1352, June 2013.
- [15] A. Ghosh, O. D. Incel, V. A. Kumar, and B. Krishnamachari, "Multi-Channel Scheduling for Fast Aggregated Convergecast in Wireless Sensor Networks," in *MASS '09: Proceedings of the 6th IEEE International Conference on Mobile Adhoc and Sensor Systems*, Macao, China, 2009, pp. 363-372.
- [16] O. D. Incel and B. Krishnamachari, "Enhancing the Data Collection Rate of Tree-Based Aggregation in Wireless Sensor Networks," in *SECON '08: Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Francisco, CA, USA, 2008, pp. 569-577.
- [17] S. Ji, Y. Li, and X. Jia, "Capacity of Dual-Radio Multi-Channel Wireless Sensor Networks for Continuous Data Collection," in *INFOCOM '11: Proceedings of the 30th IEEE International Conference on Computer Communications*, Shanghai, China, 2011, pp. 1062-1070.



- [18] S. Chen, S. Tang, M. Huang, and Y. Wang, "Capacity of Data Collection in Arbitrary Wireless Sensor Networks," in *INFOCOM '10: Proceedings of the 29th IEEE International Conference on Computer Communications*, San Diego, CA, USA, 2010, pp. 1-5.
- [19] X.-Y. Li et al., "Broadcast Capacity for Wireless Ad Hoc Networks," in *MASS '08: Proceedings of the 5th IEEE International Conference on Mobile Adhoc and Sensor Systems*, Atlanta, GA, USA, 2008, pp. 114-123.
- [20] P. G. Namboothiri and K. M. Sivalingam, "Throughput Analysis of Multiple Channel Based Wireless Sensor Networks," *Wireless Networks*, vol. 19, no. 4, pp. 461-476, 2013.
- [21] H.-S. W. So, J. Walrand, and J. Mo, "McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol," in *WCNC '07: Proceedings of the IEEE Wireless Communications and Networking Conference*, Hong Kong, China, 2007, pp. 334-339.
- [22] L. van Hoesel and P. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks," in *INSS '04: Proceedings of the 1st International Conference on Networked Sensing Systems*, Tokyo, Japan, 2004, pp. 205-208.
- [23] G. Halkes and K. Langendoen, "Crankshaft: An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks," in *EWSN '07: Proceedings of the 4th European Conference on Wireless Sensor Networks*, Delft, The Netherlands, 2007, pp. 228-244.
- [24] O. D. Incel, S. Dulman, and P. Jansen, "Multi-Channel Support for Dense Wireless Sensor Networking," *EUROSSC '06: Proceedings of the 1st European Conference on Smart Sensing and Context*, vol. 4272, pp. 1-14, 2006.
- [25] C. Xun, H. Peng, H. Qiu-sheng, T. Shi-liang, and C. Zhang-long, "A Multi-Channel MAC Protocol for Wireless Sensor Networks," in *CIT '06: Proceedings of the 6th IEEE International Conference on Computer and Information Technology*, Seoul, Korea, 2006.
- [26] J. Ansari, X. Zhang, and P. Mahonen, "Demo Abstract: Multi-Radio Medium Access Control Protocol for Wireless Sensor Networks," in *SenSys '07: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, Sydney, Australia, 2007, pp. 403-404.

- [27] X. D. Wang, X. R. Wang, X. Fu, G. L. Xing, and N. Jha, "Flow-Based Real-Time Communication in Multi-Channel Wireless Sensor Networks," in *EWSN '09: Proceedings of the 6th European Conference on Wireless Sensor Networks*, Cork, Ireland, 2009, pp. 33–52.
- [28] O. Chipara et al., "Real-Time Power-Aware Routing in Sensor Networks," in *IWQoS '06: Proceedings of the 14th IEEE International Workshop on Quality of Service*, New Haven, CT, USA, 2006, pp. 83-92.
- [29] S. Moad et al., "On Balancing between Minimum Energy and Minimum Delay with Radio Diversity for Wireless Sensor Networks," in *Proceedings Wireless Days*, Dublin, Ireland, 2012, pp. 1-6.
- [30] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks," in *Proceedings of ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, New York, NY, USA, 2001, pp. 11-25.
- [31] A. Pal and A. Nasipuri, "DRCS: A Distributed Routing and Channel Selection Scheme for Multi-Channel Wireless Sensor Networks," in *PerSeNS '13: Proceedings of the 9th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing*, San Diego, CA, USA, 2013, pp. 602-608.
- [32] Y. Wu, M. Keally, G. Zhou, and W. Mao, "Traffic-Aware Channel Assignment in Wireless Sensor Networks," in *WASA '09: Proceedings of the 4th International Conference on Wireless Algorithms, Systems, and Applications*, Boston, MA, USA, 2009, pp. 479-488.
- [33] H. Le, D. Henriksson, and T. Abdelzaher, "A Practical Multi-Channel Media Access Control Protocol for Wireless Sensor Networks," in *IPSN '08: Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, St. Louis, MO, USA, 2008, pp. 70-81.
- [34] Z. Yuanyuan, N. Xiong, J. H. Park, and L. T. Yang, "An Interference-Aware Multichannel Media Access Control Protocol for Wireless Sensor Networks," *The Journal of Supercomputing*, vol. 60, no. 3, pp. 437-460, 2012.
- [35] C. E.-A. Campbell, S. Khan, D. Singh, and K.-K. Loo, "Multi-Channel Multi-Radio Using 802.11 Based Media Access for Sink Nodes in Wireless Sensor Networks," *Sensors*, vol. 11, no. 5, pp. 4917-4942, 2011.

- [36] A. M. Canthadai, S. Radhakrishnan, and V. Sarangan, "Multi-Radio Wireless Sensor Networks: Energy Efficient Solutions for Radio Activation," in *GLOBECOM '10: IEEE Global Telecommunications Conference*, Miami, FL, USA, 2010, pp. 1-5.
- [37] W. Cheng, X. Chen, T. Znati, X. Lu, and Z. Lu, "The Complexity of Channel Scheduling in Multi-Radio Multi-Channel Wireless Networks," in *INFOCOM '09: Proceedings of the 28th IEEE International Conference on Computer Communications*, Rio de Janeiro, Brazil, 2009, pp. 1512-1520.
- [38] D. Lymberopoulos, N. B. Priyantha, M. Goraczko, and F. Zhao, "Towards Energy Efficient Design of Multi-Radio Platforms for Wireless Sensor Networks," in *IPSN '08: Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, St. Louis, MO, USA, 2008, pp. 257-268.
- [39] T. Stathopoulos et al., "End-to-End Routing for Dual-Radio Sensor Networks," in *INFOCOM '07: Proceedings of the 26th IEEE International Conference on Computer Communications*, Anchorage, AK, USA, 2007, pp. 2252-2260.
- [40] J. Gummesson, D. Ganesan, M. D. Corner, and P. Shenoy, "An Adaptive Link Layer for Range Diversity in Multi-Radio Mobile Sensor Networks," in *INFOCOM '09: Proceedings of the 28th IEEE International Conference on Computer Communications*, Rio de Janeiro, Brazil, 2009, pp. 154-162.
- [41] R. E. Irwin, Traffic-Aware Channel Assignment for Multi-Transceiver Wireless Networks (PhD Dissertation), 2012.
- [42] S. Avallone and I. F. Akyildiz, "A channel assignment algorithm for multi-radio wireless mesh networks," in *Proceedings of 16th International Conference on Computer Communications and Networks ((ICCCN)*, vol. 31(7), 2008, pp. 1343-1353.
- [43] M. A. Nezhad, L. Cerda-Alabern, and M. G. Zapata, "UBCA: A centralized utility based channel assignment mechanism for multi-radio mesh networks," *Scientific Research and Essays*, 2011.
- [44] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *ACM MobiCom*, 2004.
- [45] M. K. Marina, S. R. Das, and A. P. Subramanian, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," *Computer Networks*, vol. 54, no. 2, pp. 241-256, 2010.

- [46] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian, "Practical, distributed channel assignment and routing in dual-radio mesh networks," in *ACM SIGCOMM*, 2009.
- [47] A. Subramanian, H. Gupta, S. Das, and J. Cao, "Minimum interference channel assignment in multi-radio wireless mesh networks," *IEEE Transactions on Mobile Computing*, pp. 1459-1473, 2008.
- [48] M. A. Nezhad and L. Cerda-Alabern, "Adaptive channel assignment for wireless mesh networks using game theory," in *Proceeding of the IEEE 8th Int Mobile Adhoc and Sensor Systems (MASS)*, 2011, pp. 746-751.
- [49] W. Kim et al., "Towards distributed channel assignment in cognitive multi-radio mesh networks," in *IFIP Wireless Days*, Venice, Italy, 2010.
- [50] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in *ACM MobiHoc*, 2005.
- [51] IEEE 802.11 Working Group, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1997.
- [52] M. Cardei and A. Mihnea, "Channel assignment in cognitive wireless sensor networks," in *International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, 2014.
- [53] H. Karl and A. Willig, "Chapter 9: Localization and position," in *Protocols and Architectures for Wireless Sensor Networks.*: John Wiley & Sons, 2005.
- [54] M. Cardei and A. Mihnea, "Distributed protocol for channel assignment in cognitive wireless sensor networks," in *IEEE International Performance, Computing and Communications Conference (IPCCC)*, San Diego, CA, 2013.
- [55] (2013) Ns-3 network simulator. [Online]. <http://www.nsnam.org>
- [56] Wikipedia. [Online]. <http://www.wikipedia.org/>
- [57] A. Mihnea and M. Cardei, "Bounds on capacity in multi-channel grid networks," in *Wireless Telecommunications Symposium*, Washington, DC, 2014.
- [58] P. Carmi and M. J. Katz, "Power assignment in radio networks with two power levels," *Algorithmica*, vol. 47, no. 2, pp. 183-201, 2007.
- [59] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to*

*algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.

[60] D. Knuth, *The art of computer programming*. Reading, MA: Addison-Wesley, 1981, vol. 3.

[61] J. H. Maindonald and J. Braun, *Data analysis and graphics using R*. Cambridge, UK: Cambridge University Press, 2007.

[62] G. E. Martin, *The art of enumerative combinatorics*. New York, NY: Springer, 2001.

[63] J. H. Van Lint and R. M. Wilson, *A course in combinatorics*. Cambridge, UK: Cambridge University Press, 1992.