On the performance of STDMA Link Scheduling and Switched Beamforming Antennas in Wireless Mesh Networks

Sergio Gómez Fernández

Master's thesis
King's College London
London, United Kingdom
September 2, 2009

Abstract

Wireless Mesh Networks (WMNs) aim to revolutionize Internet connectivity due to its high throughput, cost-effectiveness and ease deployment by providing last mile connectivity and/or backhaul support to different cellular networks. In order not to jeopardize their successful deployment, several key issues must be investigated and overcome to fully realize its potential. For WMNs that utilize Spatial Reuse TDMA as the medium access control, link scheduling still requires further enhancements.

The first main contribution of this thesis is a fast randomized parallel link swap based packing (RSP) algorithm for timeslot allocation in a spatial time division multiple access (STDMA) wireless mesh network. The proposed randomized algorithm extends several greedy scheduling algorithms that utilize the physical interference model by applying a local search that leads to a substantial improvement in the spatial timeslot reuse. Numerical simulations reveal that compared to previously scheduling schemes the proposed randomized algorithm can achieve a performance gain of up to 11%. A significant benefit of the proposed scheme is that the computations can be parallelized and therefore can efficiently utilize commoditized and emerging multi-core and/or multi-CPU processors.

Furthermore, the use of selectable multi-beam directional antennas in WMNs, such as beam switched phase array antennas, can assist to significantly enhance the overall reuse of timeslots by reducing interference levels across the network and thereby increasing the spectral efficiency of the system. To perform though a switch on the antenna beam it may require up to 0.25 ms in practical deployed networks, while at the same time very frequent beam switchings can affect frame acquisition and overall reliability of the deployed mesh network.

The second key contribution of this thesis is a set of algorithms that minimize the overall number of required beam switchings in the mesh network without penalizing the spatial reuse of timeslots, i.e., keeping the same overall frame length in the network. Numerical investigations reveal that the proposed set of algorithms can reduce the number of beam switchings by almost 90% without affecting the frame length of the network.

Acknowledgments

I would like to express my gratitude to many people without their support I could have not carried this thesis to completion. First of all, I would like to express my gratitude to my Professor, Vasilis Friderikos whose excellent guidance and encouragement have helped me to seek for the best outcome. Many ideas have matured during our fruitful discussions regarding research problems.

I am particularly thankful to Oriol Gras who I had the pleasure to work with during this last year. Apart from that working together, he and my other colleges offered me a pleasant work environment; thanks for all those necessary moments of distraction. In addition, I am grateful to all my "English" friends for supporting me and helping me in a city far away from my family and friends.

Finally, I want to express my immense and never-ending gratefulness to my family, who always supported me and encouraged me during my entire life. This thesis and all my achievements are dedicated to everyone who have always believed in me.

Sergio Gómez Fernández
London, United Kingdom
September 2, 2009

Table of Contents

Introduction

Over the last few years, the rapid growth of high-speed multimedia services for residential and small business customers has created an increasing demand for last mile broadband access. Wireless Mesh Networks (WMNs) have recently emerged to provide last mile broadband wireless Internet connectivity and/or backhaul support for 3G cells and IEEE 802.11x hot spots where wired infrastructure is difficult or economically infeasible to deploy. WMNs are expected to become an important wireless technology in the near future due to its high throughput, cost-effectiveness (no need of a costly wired network infrastructure) and ease deployment.

WMNs consist of mesh routers and possibly of mesh clients connected to a predefined set of gateway nodes by establishing multi hop wireless links. In that respect each node in the WMN can operate not only as a host but also, and more importantly, as a wireless router, able to provide multiple hop connectivity to other nodes in the network. WMNs are anticipated to resolve the limitations and to significantly improve the performance of ad hoc networks, wireless local area networks (WLANs), wireless personal area networks (WPANs), and wireless metropolitan area networks (WMANs). In addition, the integration of WMNs with other networks such as the Internet, cellular, IEEE 802.11, IEEE 802.15, IEEE 802.16, sensor networks, etc., can be accomplished through the gateway and bridging functions in the mesh routers. Consequently, through an integrated WMN, the users of existing network can be provided with otherwise impossible services of these networks. WMNs will deliver wireless services for a large variety of applications in personal, local, campus, and metropolitan areas such as broadband home networking, community and neighborhood networks, enter-prise networking, building automation.

In order not to jeopardize their successful deployment, significant research factors that critically influence its performance still requires further investigation. One of the most important building blocks of WMNs is how to perform efficient scheduling so that high levels of throughput can be achieved. Thus, the medium-access control scheme considered hereafter is based on Spatial Time Division Multiple Access (STDMA), which has been proposed as an access scheme for multi-hop radio net-

works where real-time service guarantees are important. In this access scheme, time is divided into timeslots and each node can transmit only at predefined timeslots; thus, collisions can be avoided and, since nodes are spatially distributed, timeslots can be potentially reused by nodes that are sufficiently far apart. Thereby, for collision-free WMNs that support STDMA the critical aims is to increase the spectral efficiency by enhancing the spatial reuse so a predefined number of transmitting and receiving pairs of nodes can successfully transmit. The transmission rights of the different users are described with a schedule.

Finding the optimal reuse of timeslots has been shown to be an NP-complete optimization problem. In this work, we will analyze various sub-optimal STDMA link scheduling algorithms using both interference-based network model and graph-based network model that have been previously proposed in the literature.

The first important contribution of this work is a very fast randomized link scheduling algorithm for STDMA wireless mesh networks that is build upon previously proposed greedy scheduling schemes. As will become evident in the sequel, the proposed scheme can significantly decrease the frame length by up to 11 %, providing in that respect better spatial reuse of timeslots in the mesh network compared to previous well known greedy scheduling algorithms. Another key benefit of the proposed scheduling scheme is that the computations can be parallelized. Clearly, among the applications that can significantly gain from multi-core and multi-CPU enabled network elements are the scheduling algorithms. To this end, the proposed fast scheduling algorithm falls within the family of the so-called embarrassingly parallel problems since different iterations of the algorithm can be executed without requiring any communication between them. To the best of authors' knowledge this is the first work that explores the potential benefits of randomizing link scheduling order in greedy STDMA algorithms.

Many approaches have been proposed to increase capacity and flexibility of wireless systems over the last few years. The employment of advance radio wireless techniques to enhance the spatial reuse of resources in WMNs, such as directional antennas, has been widely studied in the literature. The application of directional antennas in conjunction with collision free scheduling algorithms has shown to significantly reduce the overall frame length in STDMA, compared to the scenario where nodes are equipped with omnidirectional antennas. The problem of using fixed directional antennas is the so-called scalability problem. If the routing algorithm changes due to a link failure or a new node added to the network, the antennas have to be redirected manually. Switched beam systems, formed by several available fixed beam patterns (each beam can be formed by a directional antenna), avoids the scalability problem. This type of antenna uses different patterns depending on the direction towards a communication need to be established. The change of radiation

pattern in an antenna is commonly called in the literature as a *switch*. Each time a switch is performed, the beam switched antenna consumes energy (≈ 40 μW) and requires an amount of fixed time (between 5 μs to 0.25 ms) to stabilize the new radiation pattern.

The second key objective of this work is to provide efficient algorithms for reducing the number of beam switchings to minimize the total time required for altering the transmission antenna pattern in all nodes in the network and also to reduce the energy consumed by the antennas without deteriorating the timeslot allocation. Simulation results suggest that the proposed algorithms can reduce the number of beam switches by almost 90% without affecting the spatial reuse of timeslots. To the best of authors' knowledge this is the first work that analyzes this cross issue between link scheduling and beam switching in directional antennas and develop polynomial time algorithms to minimize unnecessary beam switchings without affecting the spatial reuse of timeslots.

This thesis is written as a monograph composed by six chapters. In chapter 1 some background and current work regarding Wireless Mesh Networks, STDMA link scheduling and directional antennas are provided. The system model upon where the link scheduling and beam switching algorithm for wireless mesh networks are developed is detailed in chapter 2. In chapter 3 some STDMA link scheduling algorithms are described and analyzed and in chapter 4 enhanced versions of them are developed. Algorithms to minimize the number of required beam switches in a WMN are presented in chapter 5. Finally, the thesis concludes in chapter 6 and some interesting futures avenues of research are highlighted.

Chapter 1

Background and Related Work

In this chapter we introduce the Wireless Mesh Networks (WMNs), a promising new technology expected to become an important wireless technology in the near future. We briefly describe its main characteristics and we outline the main challenging research topics which remain still open. Since the election of the MAC protocol is essential for an efficient utilization of the wireless resources, we explain in section 1.2 the reasonings of using Spatial Reuse TDMA (STDMA) as the medium access scheme throughout this work and we sum up some of the related work on this issue. Finally, in section 1.3 we describe the different types of antennas that can be utilized in WMNs.

1.1 Wireless Mesh Networks (WMNs)

1.1.1 Introduction

The number of wireless-enabled devices such as laptops, PDAs and mobile phones has increased considerably in the last years. At the same time, the Internet has become into an essential and necessary work tool and continuous access, anywhere and anytime, is highly demanded from end users. With current wireless technology, the gap between a set of wireless devices and the Internet is bridged by fixed wireless stations, named access points. Nevertheless, the communication range of current wireless LAN technology is limited to a few hundred meters and a large coverage would require an excessive number of access points. Wireless Mesh Networks have recently emerged as a solution to extend the coverage of the existing access points, letting wireless devices to relay packets towards access points on behalf of their neighbors.

WMNs are a promising wireless technology which can provide Internet services without the need of a costly wired network infrastructure [1]. A WMN is dynamically self-organized, self-configured and self-healed, with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves (creating, in effect, an ad hoc network). With these capabilities, WMNs can be deployed incrementally, one node at a time, as needed. As more nodes are installed, the reliability and connectivity for the users increase accordingly. These features brings many advantages to WMNs such as low up-front cost, easy network maintenance, robustness, and reliable service coverage.

Additionally, WMNs have emerged as a key technology to fulfil a diverse set of applications due to its special characteristics. The envisioned applications for WMNs range from being a viable alternative to wire line last mile broadband Internet service delivery at home or offices to backhaul support for wireless local area networks to different cellular networks such as for example LTE [1] [2]. For instance, we can refer to some of them such as broadband home networking, community and neighborhood networks, enter-prise networking, building automation, intelligent transportation systems, public safety, public Internet access. Figure 1.1 shows an example of a Wireless Mesh Network.

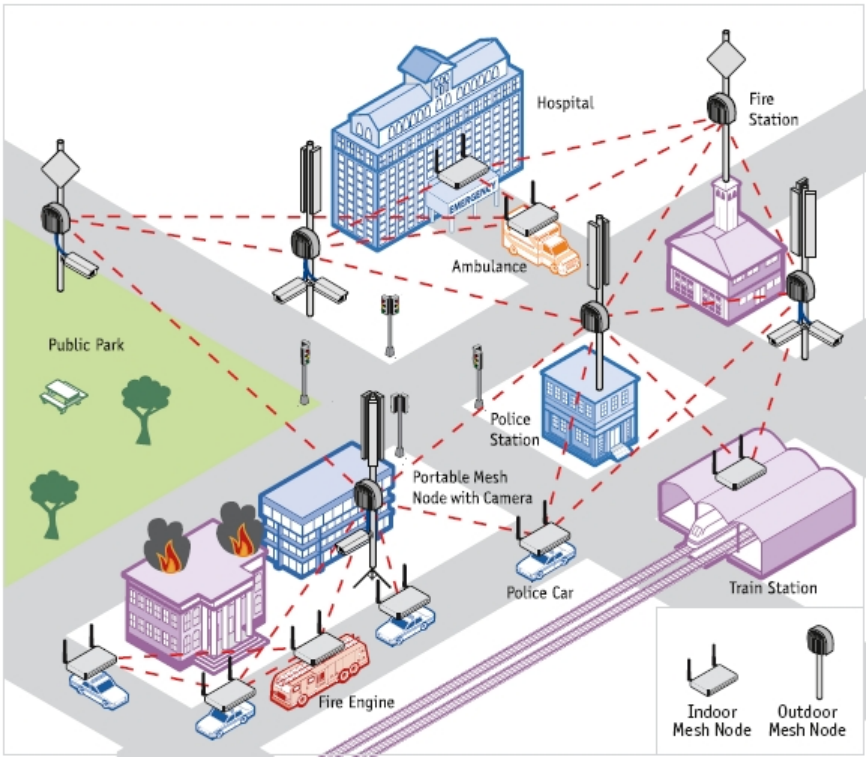

Figure 1.1: Example of a Wireless Mesh Network.

1.1.2 WMN Architecture

A wireless network is a local area network in which computers communicate through radio frequency signals instead of electrical signals transmitted over cables. In a typical configuration, connectivity for client devices such as laptops is provided through a device known as a wireless access point which serves as a gateway to a wired network and to the Internet. A wireless mesh network is a network which relies on its nodes to propagate signals. Although the wireless signal may start at some Base Station (BS) or Access Point (AP) attached to a wired network, a wireless mesh network extends the transmission distance by relaying the signal from one node to another.

WMNs consist of mesh routers and possibly of mesh clients connected to a predefined set of gateway nodes by establishing multi hop wireless links. In that respect each node in the WMN can operate not only as a host but also, and more importantly, as a wireless router, able to provide multiple hop connectivity to other nodes in the network that may not be within direct wireless transmission range of their destinations. Mesh clients can be either stationary or mobile, and can form a client mesh network among themselves and with mesh routers.

Mesh routers have minimal mobility and form the backbone of WMNs. Other than the routing capability for gateway/repeater functions as in a conventional wireless router, a wireless mesh router contains additional routing functions to support mesh networking, providing network access for both mesh and conventional clients. A mesh router is a combination of an access point and a router in one device. Like any Wi-Fi access point, the access point in the mesh router communicates with the mobile users in the area. The backhaul side of the device wirelessly relays the traffic from router to router until it reaches a gateway that connects to the Internet or other private network via a wired or wireless connection. Furthermore, compared with a conventional wireless router, a wireless mesh router can achieve the same coverage with much lower transmission power through multi-hop communications. To further improve the flexibility of mesh networking, a mesh router is usually equipped with multiple wireless interfaces built on either the same or different wireless access technologies.

In spite of all these differences, mesh and conventional wireless routers are usually built based on a similar hardware platform. Mesh routers can be built based on dedicated computer systems (e.g., embedded systems) and look compact, as shown in figure 1.2. They can also be built based on general-purpose computer systems (e.g., laptop/ desktop PC).

Mesh clients also have necessary functions for mesh networking, and thus, can also work as a router. However, gateway or bridge functions do not exist in these

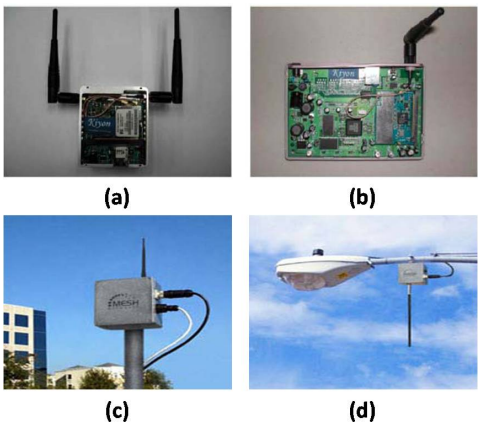

Figure 1.2: Examples of mesh routers based on different embedded systems: (a) PowerPC, (b) Advanced Risc Machines (ARM) and (c,d) Placement on the streets.

Figure 1.3: Examples of mesh clients: (a) Laptop, (b) PDA, (c) Wi-Fi IP Phone and (d) Wi-Fi RFID Reader

nodes. In addition, mesh clients usually have only one wireless interface. As a consequence, the hardware platform and the software for mesh clients can be much simpler than those for mesh routers. Mesh clients have a higher variety of devices compared to mesh routers. They can be a laptop/desktop PC, pocket PC, PDA, IP phone, RFID reader, BACnet (building automation and control networks) controller, and many other devices, as shown in figure 1.3.

1.1.3 Network Architecture Classification

The architecture of WMNs can be classified into three main groups based on the functionality of the nodes:

Infrastructure/Backbone WMNs.

This architecture is shown in figure 1.4, where dash and solid lines indicate wireless and wired links, respectively. In this type of WMNs, mesh routers form an infrastructure for clients that connect to them. The WMN infrastructure/backbone can be built using various types of radio technologies, in addition to the mostly used IEEE 802.11 technologies. The mesh routers form a mesh of self-configuring and self-healing links among themselves. With gateway functionality, mesh routers can be connected to the Internet. This approach, also referred to as infrastructure meshing, provides a backbone for

conventional clients and enables integration of WMNs with existing wireless networks, through gateway/bridge functionalities in mesh routers.

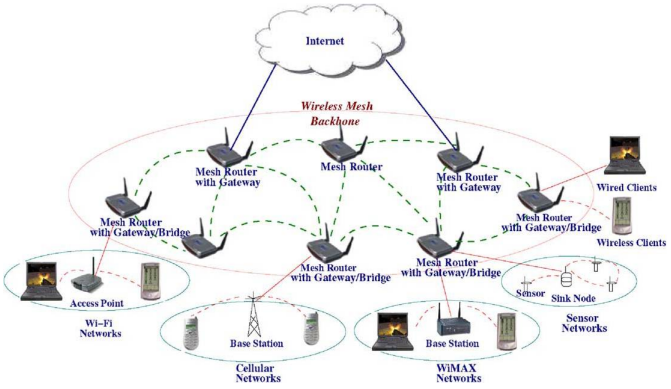

Figure 1.4: Infrastructure/Backbone WMNs.

Client WMNs.

Client meshing constitutes peer-to-peer networks among client devices. Client nodes perform routing and configuration functionalities as well as providing end-user applications to customers. Hence, a mesh router is not required for these types of networks. The basic architecture is shown in Fig. 1.5.

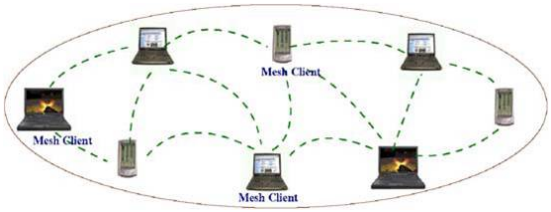

Figure 1.5: Client WMNs.

Hybrid WMNs.

This architecture is the combination of infrastructure and client meshing as can be seen in Fig. 1.6. Mesh clients can access the network through mesh routers as well as directly meshing with other mesh clients. While the infrastructure provides connectivity to other networks such as the Internet, Wi-Fi, WiMAX, cellular, and sensor networks; the routing capabilities of clients provide improved connectivity and coverage inside the WMN. The hybrid architecture is expected to be the most applicable case.

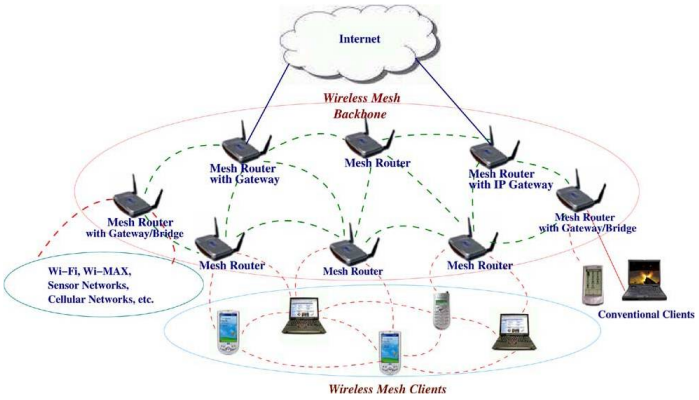

Figure 1.6: Hybrid WMNs.

1.1.4 Characteristics

The characteristics of WMNs are described next.

- **Multi-hop wireless network.** An objective of WMNs is to extend the coverage range of current wireless networks without diminishing the channel capacity. Another objective is to provide non-line-of-sight (NLOS) connectivity among the users without direct line-of-sight (LOS) links. The mesh-style multi-hopping is required to accomplish this features, which achieves higher throughput without sacrificing effective radio range via shorter link distances, less interference between the nodes and more efficient frequency re-use.

- **Support for ad hoc networking and capability of self-forming, self-healing, and self-organization.** WMNs enhance network performance, because of flexible network architecture, easy deployment and configuration, fault tolerance, and mesh connectivity, i.e., multipoint-to-multipoint communications. Due to these characteristics, WMNs have low upfront investment requirement, and the network can grow gradually as needed.

- **Multiple radios and multiple channel systems.** Since multiple channels are usually available in the frequency band of a wireless radio, they can be used to enhance the capacity of the network. A single-transceiver radio can use different channels by channel switching on the time axis according to the needs of higher layer protocols. For a multi-transceiver radio, simultaneous transmissions in different channels can be supported.

- **Mobility dependence on the type of mesh nodes.** Mesh routers usually have minimal mobility, while mesh clients can be stationary or mobile nodes.

- **Multiple types of network access.** In WMNs, both backhaul access to the Internet and peer-to-peer (P2P) communications are supported. Moreover, the gateway/bridge functionalities in mesh routers enable the integration of WMNs with existing wireless networks as the Internet, cellular, IEEE 802.11, IEEE 802.15, IEEE 802.16, sensor networks, etc. Consequently, users of existing networks can utilize services which were unavailable without the integration of WMN's.

- **Dependence of power-consumption constraints on the type of mesh nodes.** Mesh routers usually do not have strict constraints on power consumption though mesh clients may require power efficient protocols. In consequence, the MAC or routing protocols optimized for mesh routers may not be appropriate for mesh clients, as might occur in wireless sensor networks [3].

- **Compatibility and inter-operability with existing wireless networks.** WMNs need to be inter-operable with other wireless networks such as Wi-Fi, WiMAX, Zig-Bee, and cellular networks.

1.1.5 Challenging Research Issues

Wireless Mesh Networks are expected to achieve a great success in the near future. However, in order not to jeopardize their successful deployment, considerable research factors that critically influence its performance still need to be enhanced. Not only some technical issues are not solved but also it is imperative to take advantage of all favorable characteristics that a WMN can provide. Some of the open research issues and requirements of a WMN are summarized next:

- **Scalability.** It is well known that multi-hop networking communication protocols suffer from scalability issues , i.e., when the size of network increases, the network performance degrades significantly. Routing protocols may not be able to find a reliable routing path, transport protocols may loose connections, and MAC protocols may experience significant throughput reduction. In order to improve the scalability of WMN's, resolving the issue of low end-to-end throughput, requires the implementation of innovative and efficient solutions. Thus, the election of the the medium access control to the shared channel is of fundamental importance for efficient utilization of the scarce wireless resources.

- **Radio techniques.** Physical layer techniques evolves fast as RF technologies and circuit design for wireless communications progress. These advance

wireless techniques require a revolutionary design in higher layer protocols, especially MAC and routing protocols.

- **Mesh Connectivity.** Several advantages of WMNs originate from mesh connectivity which is a critical requirement on protocol design, especially for MAC and routing protocols. Network self organization and topology control algorithms are generally needed. Topology-aware MAC and routing protocols can significantly improve the performance of WMNs.
- **QoS.** Efficient resource management need to be performed to ensure the QoS of the final users.
- **Fault tolerance.** A robust and reliable network is imperative for the successful deployment of WMNs.
- **Path Redundancy.** Diversity appears to be necessary for increased performance.
- **Compatibility and inter-operability.** It is a desired feature for WMNs to support network access for both conventional and mesh clients. Thus, WMNs need to be backward compatible with conventional client nodes; otherwise, the motivation of deploying WMNs will be significantly compromised.
- **Autonomy.** Protocols must be designed to enable the network to be as autonomous as possible, in the sense of power management, self-organization, dynamic topology control, robust to temporary link failure, and fast network-subscription/user-authentication procedure.

1.2 Medium Access Control Layer: STDMA

Traditionally, MAC protocols for ad hoc networks are based on contention based access methods, for instance, a user attempts to access the channel only when it has packets to send. The user has no specific reservation of a channel and only tries to contend for or reserve the channel when it has packets to transmit. More specifically, the most commonly used protocols are based on carrier sense multiple access (CSMA), i.e. each user monitors the channel to see if it is used and the user only transmits if it is not busy. Nevertheless, this is performed in the transmitter whereas collisions appear in the receiver. This can lead to the common called hidden terminal problem. An approach to solve this problem, is to first transmit a short request-to-send (RTS) and then only send the message if a clear-to-send (CTS) is received, which is called the CSMA collision avoidance (CSMA/CA). This

is the main principle of the IEEE 802.11 standard, which at present is the most investigated and used MAC protocol. However, several RTS can be lost in a row which makes delay guarantees difficult.

Classical MAC protocols can not be implemented directly in WMNs since they are limited to one-hop communication. Therefore, currently several MAC protocols have been proposed for multi-hop ad hoc networks by enhancing classical protocols, i.e., improving the CSMA/CA protocol. These schemes usually adjust parameters of CSMA/CA such as contention window size and modify backoff procedures. They may improve throughput for one-hop communications. However, recent research indicates that CSMA/CA is not suitable to meet the high traffic demand for multi-hop cases such as in WMNs. These solutions still reach a low end-to-end throughput, because they cannot significantly reduce the probability of contentions among neighboring nodes. Therefore, despite several efforts have been made to guarantee QoS in CSMA/CA, contention-based medium access methods are inherently inappropriate for providing QoS guarantees.

In order to fundamentally resolve the issue of low end-to-end throughput in a multi-hop ad hoc environment such as WMNs, innovative solutions are required. Determined by their poor scalability in an ad hoc multi-hop network, random access protocols such as CSMA/CA are not an efficient solution. Thus, improving the design of MAC protocols based on TDMA or CDMA could improve the capacity problem.

On the other hand, one of the most important QoS parameters in many applications that are specifically sensitive to the MAC is the delay guarantees. One approach where delay bounds can be guaranteed is time division multiple access (TDMA), i.e. the time is divided into time slots and each user communicates in its own time slot. Unfortunately, in sparsely connected networks this is usually inefficient. But, due to the multi-hop properties, the time slots can often be shared by more than one user without conflicts.

Therefore, the medium-access control scheme considered hereafter to achieve both high capacity and delay guarantees, is based on spatial time division multiple access (STDMA), an extension of TDMA. Note that collision-free access techniques are the basis for constructing wireless mesh networks via the mesh mode of the IEEE 802.16 standard [4] [5]. Since nodes are spatially distributed, timeslots can be potentially reused by nodes that are sufficiently far apart, as have been defined and analyzed in the seminal work of Kleinrock [6]. In that respect, STDMA applied to multi hop wireless networks has been intensively studied previously in the literature [7] [8] [9].

1.2.1 STDMA Scheduling

One of the most important building blocks of wireless mesh networks is how to perform efficient scheduling so that high levels of throughput can be attained. For collision-free WMNs that support Spatial Time Division Multiple Access (STDMA) the critical aim is to increase the spectral efficiency by minimizing the frame length (i.e., number of timeslots) that a predefined number of transmitting and receiving pairs of nodes can successfully transmit. Note that due to spatial separation, several nodes can make wireless transmissions simultaneously; sharing the communication medium, provided there is no destructive interference of a transmission by others. However, finding the optimal reuse of timeslots, i.e., the shortest frame length, has been shown to be an NP-complete optimization problem [10]. To provide a feasible STDMA timeslot allocation a number of sub-optimal algorithms with polynomial time complexity have been previously proposed [11] [12] [13].

In a WMN, nodes might change their location and thereby nodes that can transmit simultaneously without conflict at one moment, might not be able to do so later. Thus, the STDMA schedule must be updated whenever something changes in the network. The aim of the heuristic STDMA schedulers is to design fast algorithms which minimize the frame length of the network without requiring a high computational complexity. An scheduling algorithm may reduce considerably the number of timeslots, but if it requires to much time to be performed, by the time the new schedule has been propagated it is already obsolete, due to node movements.

This can be done in a centralized manner, i.e., all information is collected into a central node, which is responsible to calculate collision-free schedules to maximize the spatial reuse. This schedule is then propagated throughout the network. This type of schedulers can be very efficient because the central node has all information about the network. On the other hand, another way to create STDMA schedules is to do it in a distributed manner, i.e., when something changes in the network, only the nodes in the local neighborhood of the change will act on it and update their schedules without the need to collect information into a central unit.

Moreover, all transmission rights may be assigned to the links, i.e., both transmitting and receiving nodes are determined before the schedule is performed by the routing algorithm. This is called link assignment or link activation and is normally employed for unicast traffic. An alternative option may be to assign transmission rights to the nodes instead. In this case only the node is scheduled to transmit in the time slot. Any of its neighbors, or all, can be chosen to be the receiving node. This is called node assignment or node activation and is generally used for broadcast traffic.

Hence, in this work we focus our attention on the centralized STDMA link

scheduling scheme, as will be shown in chapter 3. Thereafter, in chapter 4 we aim to enhance the proposed scheduling techniques maximizing the spatial reuse of timeslots.

1.2.2 Previous Work

The concept of Spatial-TDMA has first been presented in the seminal work of Kleinrock [6]. A significant part of previous research in the area of STDMA scheduling has been concentrating on graph based representation of the STDMA scheduling problem and associated graph theoretic tools; conceiving in that respect the STDMA scheduling as a graph coloring problem [14] [15] [16]. Despite their attractiveness, graph coloring based algorithms can resolve only the problems of primary and secondary conflicts between the links that need to be scheduled [12]. Hence, their drawback is that they do not consider the effect of aggregate interference, as reflected at the Signal to Interference Noise Ratio constraint for successful packet transmission and, therefore they may lead to schedules which are infeasible [12], [17]. To resolve this issue a number of previous works have explicitly taken into consideration the constraints (the so-called physical interference model) together with power control for constructing minimum frame length schedules [13] [18], [19].

Over the years research on both node assignment and link assignment strategies for STDMA has been done, since both broadcast traffic and unicast traffic have been considered important in multihop ad hoc networks. Examples of algorithms that generate node assignment schedules can be found in [10] [20] and examples of link assignment algorithms can be found in [7] [18] [21].

1.3 Physical Layer: Advanced Radio Techniques

Currently many approaches have been proposed to increase capacity and flexibility of wireless systems. Typical examples include directional and smart antennas [22] [23], MIMO systems [24], and multi-radio/multi-channel systems [25] [26]. In addition, to further improve the performance of a wireless radio and control by higher layer protocols, more advanced radio technologies such as reconfigurable radios, frequency agile/cognitive radios [27] [28], and even software radios [29] have been employed in wireless communication.

1.3.1 Switched Directional Antennas

Traditionally, omnidirectional antennas have been deployed in several types of networks due to its simplicity and low cost. However, as the RF techniques have been evolving, new possibilities to enhance the performance of the networks by employing sophisticated antennas have been appearing. Directional antennas and smart antennas are the antenna systems that are currently being studied to improve the capacity of WMNs and other ad hoc networks.

Fixed directional antennas focus its energy towards an specific area and are formed by just one directional fixed pattern, as can be seen on the left side of figure 1.7. Directional antennas concentrate the transmitted energy into a limited region avoiding the inefficiencies of omnidirectional antennas caused by spreading the energy in all directions and therefore increasing significantly the interference power level to other concurrent transmitting links. Undoubtedly, the significant level of interference reduction that can be achieved by directional antennas permits for a higher spatial reuse of resources (i.e., timeslots) compared to omnidirectional antennas, leading in that respect to better exploitation of the scarce wireless resources and potentially to better overall performance of the network. The problem of using directional antennas is that we do not have scalability. If a new node is added to the network, the routing algorithm might change and thus the antennas have to be redirected manually.

An alternative solution to avoid the scalability problem is the employment of smart antennas. Smart antennas are classified in two types, switched beam antennas and adaptative arrays. The first type are formed by several available fixed beam patterns, where each beam can be a directional antenna. This kind of antennas solves the problem of scalability, since when a new node is added to the network, it is only necessary to activate another beam of the antenna if the current beams do not cover the region of the new node. Its problem is that as the radiation patterns are fixed, it is possible that the nodes do not communicate through its maximum gain directions. Therefore, the power required to establish a communication between two nodes would be higher than if they were communicating through its maximum gain directions. This might produce an increase of the interference generated by this communication to other nodes communicating at the same time. However, the simplicity and low cost of this kind of antennas, make them an efficient alternative to omnidirectional antennas despite this problem. The other type of antennas, the adaptative arrays, are able to create a wide variety of radiation patterns pointing towards any direction. Despite being the most efficient antennas, these type are complex and expensive. Hence, the huge investment necessary becomes other low-cost antennas into a more realistic implementation in the near future.

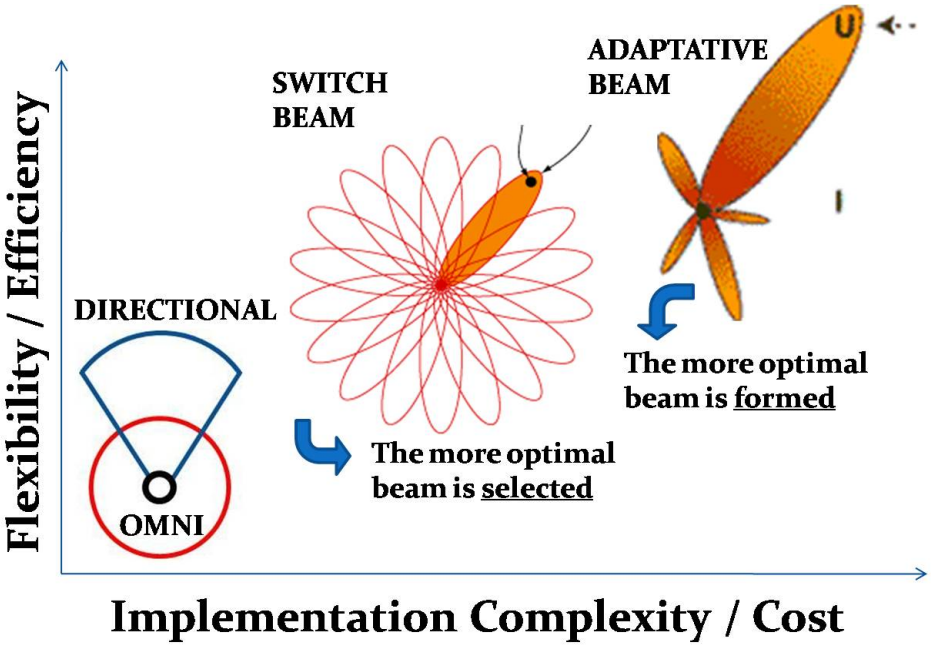

Figure 1.7: Left: Directional antenna. Center: Switched Beamforming Antenna. Right: Adaptative array antenna.

In chapter 5, we will focus on the performance of switched beam antennas formed by directional antennas. The election of this type of smart antennas is due to its simplicity, low cost and proven performance. As previously explained, this antenna uses different patterns depending on the direction towards a communication need to be established. The change of radiation pattern in an antenna is commonly called in the literature as a *switch* [22]. Therein lies the main side effect of employing this type of antennas, each time a switch is performed, the beam switched antenna consumes energy ($\approx$ 40 μW) and requires an amount of fixed time (between 5 μs to 0.25 ms) to stabilize the new radiation pattern [30] [31] [32] [33]. Thereby, the aim of chapter 5 is to minimize the overall number of required beam switches in a wireless mesh network with switched beam antennas without deteriorating the spatial reuse of timeslots.

1.3.2 Previous Work

The employment of directional antennas for WMNs, which is the basis upon which the switched beam antennas used in this work are formed, has been proposed and widely studied [22] [23] [34] [35] over the last few years. In addition to directional antennas, as mentioned above, a collision-free spatial time division multiple access (STDMA) medium-access control scheme is considered. The application of

directional antennas in conjunction with collision free scheduling algorithms has shown to significantly reduce the overall frame length in STDMA, compared to the scenario where nodes are equipped with omnidirectional antennas [36] (and references therein). In addition, the deployment of switched beam antennas has been previously studied in conjunction with routing decisions to minimize the energy consumption in wireless multi hop networks [37].

Chapter 2

System Model

In order to determine which links can transmit at the same time when we do the scheduling, we need a description of the network. The network models used by STDMA algorithms varies in complexity depending on the requirements of the system. In addition, to apply the scheduling algorithms to the network, a preliminary set up need to be performed such as the calculation of the required power to establish a communication or the computation of the routing protocol. Besides, a description of the different antenna models employed throughout this work is presented. Finally, the interference models considered hereafter are analyzed.

2.1 Set Up of the Network

2.1.1 Network Model

We consider a Wireless Mesh Network, which can be modeled by a network graph $G^N = (V^N, E^N)$, where V^N expresses the set of nodes (mesh routers and clients) and E^N denotes the set of wireless links. Each node $i \epsilon V$ is equipped with one wireless interface card, and with D_i we express the degree of node i. We further assume that all nodes in the mesh network operate at the same frequency band (hence frequency reuse factor is one) and we do not consider spurious or other inter channel interference. The packet length is normalized and occupies a single timeslot.

The wireless mesh network is deployed in a square area AxA Km^2 containing N wireless nodes that are randomly and uniformly distributed over the squared area. A special node in the topology acts as the gateway node for providing Internet connectivity; throughout the numerical investigations and without loss of generality

a single gateway node is considered. In addition, it should be noted that only unidirectional links in the downlink scenario (from the gateway to the nodes) are considered. Similar results are expected to hold also for the uplink scenario but are not considered in this work.

2.1.2 Power Required

The power transmitted by a node in the network is determined by the minimum power required to establish a communication with a receiver node in absence of interference as long as it does not exceed the maximum power transmitted (P_{max}), in other words, it is the minimum power to accomplish a Signal to Noise Ratio threshold (SNR). In case that the power required exceeds P_{max}, a link can not be established between this pair of nodes.

It is important to remark that independently of the antenna employed the power required by a node i to establish a link with node j must be calculated considering omnidirectional antennas, where the antenna gain is 1. It is quite obvious to use omnidirectional antennas whether the network employs this kind of antennas since we want to minimize the power transmitted, though it can be more confusing whether the network uses directional antennas. The aim of not using directional antennas to calculate the power requirements for establishing a link is to ensure that irrespectively if the antenna patterns of the nodes has changed, their SNR thresholds will still be accomplished. Note that the minimum antenna gain for a directional antenna is 1 (when all patterns of the antenna are active at the same time), as can be deducted from expression 2.5. Therefore, the communication between two nodes is ensured irrespectively of the type or configuration of the antenna employed.

To calculate the required transmission power level for link (i, j) the following simple path loss model has been considered hereafter,

$$PL(d(i,j)) = PL(d_0) + 10\eta log\left(\frac{d(i,j)}{d_0}\right) \tag{2.1}$$

where $d(i, j)$ expresses the Euclidean distance of link (i, j), $PL(d_0)$ is the close-in reference distance loss, which is assumed to be equal to 78 dB for distance d_0 equal to 50 meters, and finally η denotes the path loss exponent, which in general take values between 2 (free-space path loss) to 6 depending on the environment under consideration [38].

Hence, the power required P_{ij} for link (i, j) is obtained using the following expression,

$$P_{ij} = SNR + PL(d(i,j)) + W \quad | \quad P_{ij} < P_{max} \tag{2.2}$$

where SNR denotes the Signal to Noise Ratio threshold and W expresses the total sum power of background and thermal noise.

2.1.3 Routing algorithm

Based on all feasible links that can be constructed when no co-channel interference is considered, a shortest path spanning tree is constructed rooted at the gateway node spanning all other nodes in the network. The spanning tree is based on the minimum power routing (MPR) scheme, as described and analyzed in [7]. The MPR scheme is based on the well-known Dijkstra's algorithm and uses the required transmitted power previously detailed to combat the path loss as the cost of the link. This cost results in reduction of the overall interference enhancing the spatial reuse in the network. The spanning tree constructed herein forms the Communication Graph $G = (V, E)$ used throughout the scheduling algorithms.

Since a shortest path spanning tree is created that rooted at the designated gateway node, the links that need to be scheduled are $N - 1$ in all numerical investigations. Additionally, we consider the often-used scenario where each link requires to be scheduled in only one timeslot. Therefore, the frame length computed by the scheduling algorithms expresses the minimum possible number of timeslots so that each link transit at least once. Finally, figure 2.1 shows an example of a network of 10 nodes, where all feasible links that can establish a communication are colored in blue. Besides, colored in red is shown the shortest path spanning tree constructed based on all feasible links and rooted at the gateway node.

2.1.4 Degree Constraint

Scheduling is taking place under the assumption that a node cannot transmit and receive at the same timeslot. This can be accomplished by the following two constraints: the indegree constraint which ensures that only one node can send traffic to the same receiving node at each timeslot and the outdegree constraint which ensures that a transmitting node can only send traffic to one receiving node per timeslot [39]. In other words, two links can not be scheduled at the same timeslot if they share an endpoint. Figure 2.2 illustrates the degree constraint for a node with 3 incoming and 3 outgoing links.

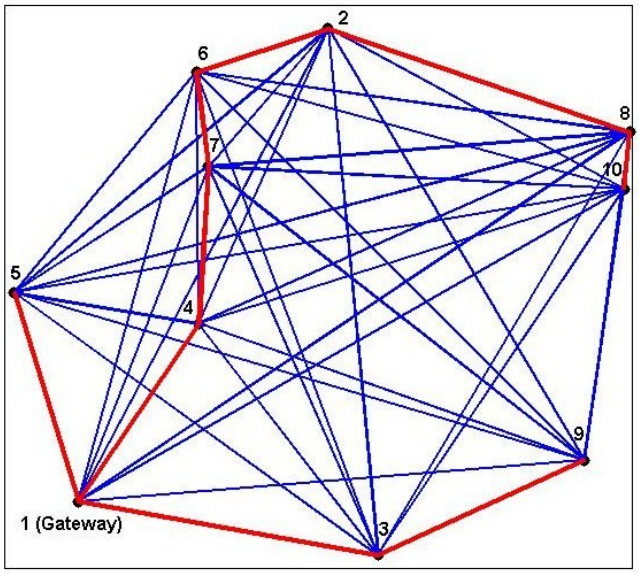

Figure 2.1: Example of a network of 10 nodes. In blue, all possible links; in red, the spanning tree, in other words, the links to be scheduled

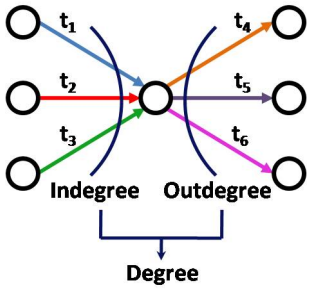

Figure 2.2: Degree Constraint of a node with 6 adjacent links.

Note that using Dijkstra's algorithm as a routing protocol, each node in the network only receives data from one node (except the gateway that does not receive data from any node) and it might transmit to several nodes. Hereby, the only incoming and all the outgoing links of each node or, as defined in the graph theory book [40], all the adjacent links of each node, will be scheduled in different timeslots.

The degree constraint bounds the minimum number of timeslots (TS) of the network. Thus, the minimum number of timeslots is the maximum degree of the network, in other words, the maximum number of adjacent links among all nodes. In conclusion, since the number of links to be scheduled are $N-1$, the number of timeslots necessaries in the network is bounded by the following inequality,

$$max(D_i) \leq TS \leq N-1 \tag{2.3}$$

For instance, it can be seen in figure 2.1 that the links to be scheduled in this network are 9 (network of 10 nodes) and the minimum number of timeslots is 3 (degree of the gateway). Hence, the number of timeslots obtained by any scheduling algorithm in this network will be bounded between this range of values.

2.2 Antenna Model

In chapters 3 and 4 the networks will employ omnidirectional antennas. This kind of antennas are rather simple and no assumption is required; they spreads its power in all directions and transmits with unit gain. In chapter 5 though, a more sophisticated system is utilized, which despite its simplicity requires a detailed description of the antenna model.

The baseline assumption regarding the antenna model employed in chapter 5 is that each node in the WMN is equipped with a switched beam by using phase array antennas forming a radiation pattern with K identical and selectable beams. The radiation pattern of a beam is approximated by a main lobe of constant gain g_m and beamwidth θ_m and a side lobe of constant gain g_s and beamwidth (2π - θ_m), as it is illustrated in figure 2.3.

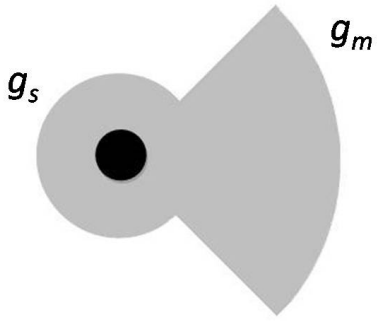

Figure 2.3: The "Keyhole" radiation pattern used to model a beam of a directional antenna.

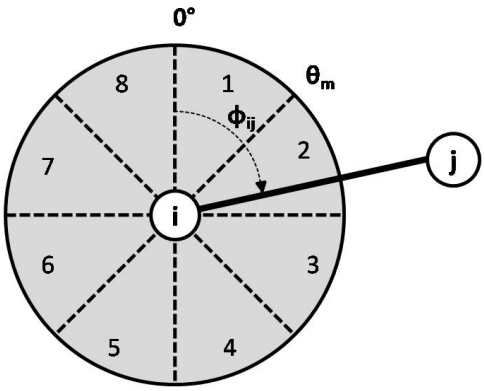

Figure 2.4: An example of a node with a switched beam antenna with 8 beams, where the angle of each beam (θ_m) is 45°. The selection of the beam that is used for link (i, j) is based on angle ϕ_{ij} which depict which beam will be used for this communication link.

Without loss of generality, we assume that the direction of each beam is fixed and the boresights of the first sector are always directed towards the 0° and θ_m on a

polar plane. When a link needs to be established for communication between nodes i and j, then node i calculates the relative angle, ϕ_{ij}, between the 0° in the polar plane and link (i, j) to determine the employed antenna beam. Based on the above assumptions, the selected beam ξ_{ij} that node i would use to communicate with node j via link (i, j) is given by equation 2.4 below,

$$\xi_{ij} = \left\lceil \frac{\phi_{ij}}{\theta_m} \right\rceil \tag{2.4}$$

Note that node j will apply the exact same procedure to determine the beam ξ_{ji}. Figure 2.4 shows the distribution of the antenna beams and the beam selected by node i to establish link (i, j).

Directional antennas focus a significant amount of the transmitted energy in a specific localized area. In order to model directional antennas it is necessary to find an expression that relates the gains and beamwidth with the total amount of energy transmitted by an omnidirectional antenna with unit gain. Several models have been considered in the literature, and in this work we adopt a model similar to the one proposed in [22]. The model assumes a 2 dimensional radiation pattern to calculate the parameters of a directional antenna. Equation 2.5 relates the parameters of the directional antenna.

$$g_m \frac{1}{2\pi}\theta_m + g_s \frac{1}{2\pi}(2\pi - \theta_m) = 1 \tag{2.5}$$

Equation 2.5 expresses how the total amount of transmitted power is spread between the main and side lobes. This same equation is also valid when $N_b > 1$ beams are activated simultaneously. In that case, to model the antenna gains it is considered an antenna that has a main lobe with beamwidth $N_b \cdot \theta_m$ and a side lobe with beamwidth (2π - $N_b \cdot \theta_m$).

2.3 Interference Model

The main factor that limits capacity in mesh networks (and in wireless networks in general) is interference, which is a consequence of using a shared communication medium. Therefore, an accurate modeling of interference is necessary in order to obtain significant theoretical and/or simulation-based results. In the literature, two main interference models have been proposed [41]: the protocol and the physical interference models.

Most algorithms assume that transmission ranges are limited (usually circular), and beyond this no interference is caused. Thereby, a communication between nodes u and v is successful if no other node within a certain interference range from the receiver v is simultaneously transmitting. Due to its simplicity, the protocol interference model has been mostly used in the literature [6] [10]. Using this model, scheduling can be transformed into a graph-theoretical problem, i.e. the network is represented as an interference graph as described in section 3.3.1. This model thereby makes it simple to create STDMA algorithms, though a disadvantage is that this model does not take into consideration the combination of interference from several nodes which can cause transmissions to fail.

A more realistic, although even more complex, is the physical interference model. In this case, a communication between nodes u and v is successful if the SINR (Signal to Interference and Noise Ratio) at the receiver v is above a certain threshold, whose value depends on the desired channel characteristics (e.g., data rate). This model is less restrictive than the protocol interference model: it may occur that a message from node u to node v is correctly received even if there is a simultaneous transmitting node w close to v (for instance, because node u is using a much larger transmit power than node w). As a result, higher network capacity may be achieved by applying the physical interference model.

Observe that the interference-based scheduling is representative of a scenario that does not use CSMA techniques; instead, transmissions should be carefully scheduled so that only links that do not conflict with each other transmit simultaneously. In other words, the physical interference model is suited for use with spatial TDMA-like channel access schemes. This model have been used by several STDMA scheduling algorithms [7] [9].

In this work, we employ both models to find schedules in a polynomial time. However, to fairly compare the performance of both models, all schedules obtained need to be feasible, in other words, all receiver nodes have to satisfy a Signal to Interference and Noise Ratio ($SINR$). The $SINR$ whether omnidirectional or directional antennas are utilized are defined in the following sections.

2.3.1 SINR for Omnidirectional Antennas

For a single transmission bit-rate and omnidirectional antennas, each link $(i, j)\ \epsilon\ E$ at timeslot t needs to satisfy a signal to interference noise ratio ($SINR$) threshold (γ) for successful packet decoding in the receiver node j. More specifically, the SINR inequality that needs to be satisfied can be written as follows,

$$\frac{g_{ij}p_{ijt}}{\sum_{(m,n)\epsilon E\setminus\{(i,j)\}} g_{mj}p_{mnt} + W} \geq \gamma \tag{2.6}$$

where p_{ijt} denotes the transmission power for link (i,j) at timeslot t, g_{ij} is the link gain for link (i,j) (defined as the inverse of the path-loss $PL(d(i,j))$) and W expresses the lump sum power of background and thermal noise.

2.3.2 SINR for Directional Antennas

For a single transmission bit-rate and directional antennas, the previous SINR inequality that each link $(i,j)\ \epsilon\ E$ must satisfy to ensure its feasibility, may be rewritten as follows,

$$\frac{g_{ij}A_{ij}^{ab}p_{ij}}{\sum_{(m,n)\epsilon E\setminus\{(i,j)\}} g_{mj}A_{mj}^{cd}p_{mn} + W} \geq \gamma \tag{2.7}$$

where p_{ij} denotes the transmission power for link (i,j), g_{ij} is the link gain for link (i,j) (defined as the inverse of the path-loss $PL(d(i,j))$), A_{ij}^{ab} is the antenna gain of the transmission node i when using beam pattern a multiplied with the antenna gain of the receiver node j with pattern b and W expresses the power of background and thermal noise. Note that each antenna can have a number of different patterns depending on which beams are active at the same time. The number N_p of different possible patterns for a switched beam antenna is given by equation 2.8 below,

$$N_p = \sum_{k=1}^{B} \binom{B}{k} \ \mid \ B = \frac{2\pi}{\theta_m} \tag{2.8}$$

In fact, the above expression is an absolute upper bound; the different beam patterns will be limited by the number of active links that belong in different beams. Therefore, the actual number of possible beam patterns for node i can be calculated using expression 2.8 if we substitute B with B_i, where $B_i \leq B$ is the number of beams that contain at least one active link. An example on the different beam forming patterns is shown in figure 2.5 for a node with a degree 3. Note that in the case where all beams operate concurrently (pattern shown in figure 2.5 for the case where $k = 7$) there will be no beam switch, whereas in all other cases the number of beam switchings will depend on how the links are ordered by the scheduling algorithm.

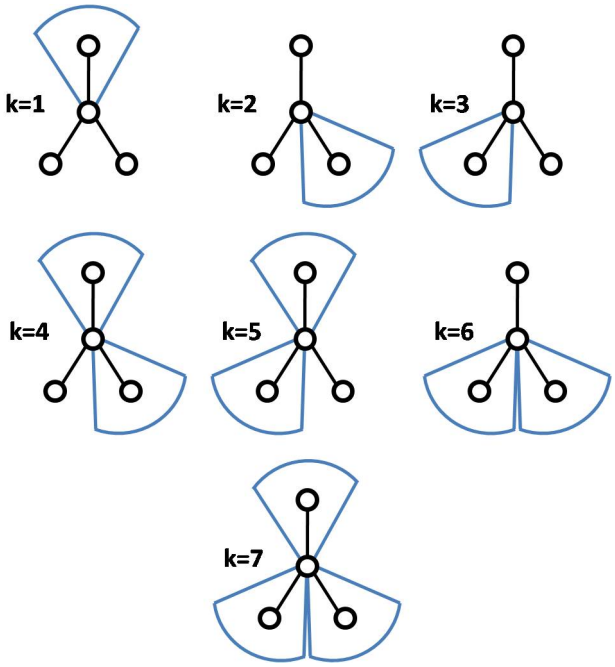

Figure 2.5: Different possible beam forming patterns for a node with degree 3. For this example $N_p = 7$.

2.3.3 Greedy Time Complexity Reduction

The link scheduling procedure for the physical interference model requires to check the feasibility of the same node several times. Although, it is not necessary to re-calculate all parameters of the SINR of a node each time that is needed to check its feasibility. It is proved in the following expressions, that each time a new node (u, v) is trying to be scheduled in timeslot t, the only parameter that need to be calculated is the interference produced by the new node to each node that is already scheduled in timeslot t. Let us assume that the new node (u, v) is scheduled at timeslot t, then the inequality described above (2.6) that each link (i, j) ϵ E scheduled at timeslot t must satisfy to ensure its feasibility, can be rewritten as follows,

$$\begin{aligned} \sum_{(m,n)\epsilon L\setminus\{(i,j)\}} g_{mj}p_{mnt} + W &\leq \frac{g_{ij}p_{ijt}}{\gamma} \\ \Leftrightarrow \sum_{(m,n)\epsilon L\setminus\{(i,j)\}} g_{mj}p_{mnt} &\leq \frac{g_{ij}p_{ijt}}{\gamma} - W \end{aligned} \tag{2.9}$$

Up to this point, depending on which node is checked, the inequality that must

be satisfied is different. The constraint that the new node *(u,v)* needs to ensure is detailed below. Observe that in this case link (i, j) corresponds to link (u, v).

$$\begin{aligned} \sum_{(m,n)\epsilon L\setminus\{(u,v)\}} g_{mv}p_{mnt} &\leq \frac{g_{uv}p_{uvt}}{\gamma} - W \qquad (2.10a) \\ \Leftrightarrow 0 &\leq \frac{g_{uv}p_{uvt}}{\gamma} - W - \sum_{(m,n)\epsilon L\setminus\{(u,v)\}} g_{mv}p_{mnt} = \beta'_{ij} \end{aligned}$$

For each node that was previously scheduled in timeslot t,

$$\begin{aligned} g_{uj}p_{uvt} + \sum_{(m,n)\epsilon L\setminus\{(i,j),(u,v)\}} g_{mj}p_{mnt} &\leq \frac{g_{ij}p_{ijt}}{\gamma} - W \qquad (2.10b) \\ \Leftrightarrow g_{uj}p_{uvt} &\leq \frac{g_{ij}p_{ijt}}{\gamma} - W - \sum_{(m,n)\epsilon L\setminus\{(i,j),(u,v)\}} g_{mj}p_{mnt} = \beta_{ij} \\ \Leftrightarrow 0 &\leq \beta_{ij} - g_{uj}p_{uvt} = \beta'_{ij} \end{aligned}$$

Therefore, the close form of the constraint that needs to be satisfied to ensure the feasibility of the new set of nodes at timeslot t is

$$\beta'_{ij} = \begin{cases} \dfrac{g_{uv}p_{uvt}}{\gamma} - W - \displaystyle\sum_{(m,n)\epsilon L\setminus\{(u,v)\}} g_{mv}p_{mnt}, & \text{(i,j) = (u,v)} \\ \beta_{ij} - g_{uj}p_{uvt}, & \text{(i,j) } \neq \text{ (u,v)} \end{cases} \qquad (2.11)$$

Looking at expression 2.10b, we can see that the parameters to obtain β_{ij} where already calculated in previous iterations; note that β_{ij} denotes the maximum amount of extra interference that a node can accept. Therefore, node (u, v) can be scheduled at timeslot t, if the set of nodes at timeslot t satisfies that $\beta'_{ij} \geq 0$, which implies that the new set of nodes is feasible. If node (u, v) is scheduled successfully, we update the values of β'_{ij}, using expression 2.11, so they can be utilized afterwards. Finally, observe that the value β'_{ij} for node (u, v) requires more calculations than the rest of values of β'_{ij}.

An analogous expression of the constraint that needs to be satisfied to ensure the feasibility whether directional antennas are employed, can be derived from inequality 2.7. The procedure is analogous to the previous case, thus in this case β'_{ij} can be rewritten as follows,

$$\beta'_{ij} = \begin{cases} \dfrac{g_{uv}A^{ab}_{uv}p_{uvt}}{\gamma} - W - \displaystyle\sum_{(m,n)\epsilon L\setminus\{(u,v)\}} g_{mv}A^{cd}_{mv}p_{mnt}, & \text{(i,j)} = \text{(u,v)} \\ \beta_{ij} - g_{uj}A^{cd}_{uj}p_{uvt}, & \text{(i,j)} \neq \text{(u,v)} \end{cases} \tag{2.12}$$

Another interesting result that we can extract analyzing how the feasibility of the nodes is checked, is described next. Note that the distance considered hereafter is the Euclidean distance between the transmitter node of a link a and the receiver node of a link b.

Lemma 1 *Being $L \subseteq E$ a set of feasible links scheduled in timeslot t and a link $(i,j) \epsilon E$. If link (i,j) is feasible with respect of the set of links L, then if the feasibility of the set of links L with respect of link (i,j) is checked starting with the link with minimum distance to link (i,j), whenever a link (u,v) with minimum distance to link (i,j) has also minimum SINR and is feasible, the rest of links further located than link (u,v) are feasible with respect of link (i,j).*

Proof 1 Link (i,j) is always transmitting with the same power to its receiver, thereby the different levels of interference that the set of links scheduled in timeslot t are receiving from link (i,j) depends only on the link gain. Note that the link gain is defined as the inverse of the path loss. Then, being the path loss at the close-distance and the path loss exponent constant, the only parameter not constant to calculate the path loss is the distance between link (i,j) and the the rest of links, as can be observed in expression 2.1. The path loss decreases as the distance increases, so the link with the minimum distance to link (i,j) suffers the maximum interference. On the other hand, as can be deducted looking expression 2.11 (or 2.12), the link with lower SINR is the link with lower value of β and therefore the link that can accept less interference. Hence, if the link that can accept less interference is able to accept the maximum interference that comes from link (i,j), then the rest of links further located than link (u,v) are feasible with respect of link (i,j).

Lemma 2 *Being $L \subseteq E$ a set of feasible links scheduled in timeslot t and a link $(i,j) \epsilon E$. If link (i,j) is feasible with respect of the set of links L, then if the feasibility of the set of links L with respect of link (i,j) is checked starting with the link with maximum distance to link (i,j), whenever a link (u,v) with maximum distance to link (i,j) has also maximum SINR and is infeasible, the rest of links closer located than link (u,v) are infeasible with respect of link (i,j).*

Proof 2 Analogous to proof 1, the link with maximum distance to link (i,j) suffers the minimum interference and the link with higher SINR is the link that can accept

more interference. Therefore, if the link that can accept more interference is not able to accept the minimum interference that comes from link (i, j), then the rest of links closer located than link (u, v) are infeasible with respect of link (i, j).

Observe that lemmas 1 and 2 can be used to decrease the time consumption of the scheduling algorithms employed hereafter.

Chapter 3

STDMA Link Scheduling

3.1 Introduction

Spatial reuse TDMA has been proposed as a medium access scheme for multi-hop radio networks, such as WMNs, where real-time service guarantees are important. The idea is to allow several radio terminals to share the same timeslot when the radio units are geographically separated such that small interference is obtained. The transmission rights of the different users are described with a schedule which will be considered valid or feasible if satisfies the SINR constraint defined in 2.6. In order to maximize the spectral efficiency of the network, i.e., minimizing the frame length of the schedule, is necessary an accurate modeling of the interference. Two main interference model have been widely studied: the protocol interference model transformed into a graph-based model and the interference model.

In this thesis we will study various STDMA link scheduling algorithms based on both network models. Algorithms that utilize the interference model always produces feasible schedules, however the graph-based model may lead to infeasible timeslot allocations when the SINR thresholds are taking into account. Therefore, to fairly compare the performance of both models, all those schedules which do not accomplish the SINR thresholds will be discarded. Finally, note that we consider the often-used scenario where each link requires to be scheduled in only one timeslot. Hence, the frame length computed by the scheduling algorithm expresses the minimum possible number of timeslots so that each link transit at least once.

The remainder of the chapter is organized as follows. In section 3.2, two algorithms that uses the interference model are described and compared. An algorithm that utilize graph coloring techniques is outlined in section 3.3. The chapter

concludes in section 3.4 with simulation-based results on the performance of the scheduling algorithms presented.

3.2 STDMA Link Scheduling using Interference-Based Network Model

The strategy followed by several scheduling algorithms that utilize the interference-based network model, commonly called the physical interference model, consists in firstly sorting the links based on a pre-defined criterion and then greedily packing the links into timeslots to generate feasible schedules. We detail in the sequel two well known heuristic scheduling algorithms, namely the Greedy Physical [18] and the Packing Heuristic [7] algorithms.

3.2.1 Greedy Physical (GP)

Greedy physical (GP) is a simple heuristic to prove the approximation bound for the scheduling problem with physical interference. Firstly, algorithm starts by ordering the links to be scheduled according to the interference number, which is explained next. The interference number of a link $(i, j)\ \epsilon\ E$ is the number of links $(m, n)\ \epsilon\ E \setminus (i, j)$ that cannot establish a communication at the same time such the links (m, n) and (i, j) do not share an endpoint and is infeasible. A set of two links is considered infeasible when the receiver nodes do not satisfy the constraint $SINR$ described in equation 2.6. The idea behind the notion of the interference number is to measure, to some extent, the amount of interference generated in the network by a communication going on along a certain link. Thereby, a sorted list is created with the higher interference number first, since the aim of these algorithm is to schedule firstly the links that generate relatively higher interference (which are more difficult to schedule). Thereafter, links are packed according to the scheduling strategy stated in algorithm 1.

3.2.2 Packing Heuristic (PH)

The Packing Heuristic presented herein is the algorithm that has been detailed in [7], which is also a variation of the heuristic algorithm used in [42] and [43], where different weights are utilized to sort the links. This algorithm tries to pack as many links as possible in each timeslot, having as a starting point a list where the links are

Algorithm 1: Pseudo-code of the GP algorithm

Input : L, a list containing all links sorted in decreasing order by its
: interference number.
Output : S, a feasible schedule.
: TS, frame length found for S.

1: $TS \leftarrow 0$
2: **for** *each link* L_i *in* L **do**
3: Schedule link L_i in the first available slot such that the resulting set of scheduled transmission is feasible with the physical interference model.
4: **if** *currently available slots are not sufficient to schedule* L_i **then**
5: Add a new slot at the end of the schedule S and schedule link L_i in this slot.
6: Let $TS \leftarrow TS + 1$
7: **endif**
8: **endfor**

sorted with the links that require higher transmitted power first. The pseudo-code of the proposed scheme is shown in algorithm 2.

Algorithm 2: Pseudo-code of the PH algorithm

Input : A, a list containing all links sorted by its power levels (highest
: power first)
Output : B, a feasible schedule
: TS, frame length found for S

1: $t \leftarrow 1$
2: $B \leftarrow$ Empty list
3: At timeslot t schedule the first link in list A for transmission and shift it from list A to list B.
4: **repeat**
5: Proceed down the current list A scheduling links for transmission in timeslot t, if feasible, and shifting them to list B if they transmit.
6: Let $t \leftarrow t + 1$
7: **until** A *is empty*
8: Let $TS \leftarrow t - 1$

3.2.3 Comparison between Packing techniques

It has to be highlighted that there is only one difference between the Packing Heuristic and the Greedy Physical that results in different schedules. The difference is the way the links are sorted in the initial list. In the Packing Heuristic the first links to be scheduled are the ones that have the highest transmitted power, whereas in the Greedy Physical the priority is given to the links that cause more interference. There is another difference between the GP and the PH though, in this case, does not lead to any different schedule. This difference lies in the way the algorithm proceeds to pack the links. In the Packing Heuristic we fix a timeslot and we try to pack in it all the links that have not yet transmitted, whereas in the Greedy Physical we fix a link and we try to pack it in the first timeslot available.

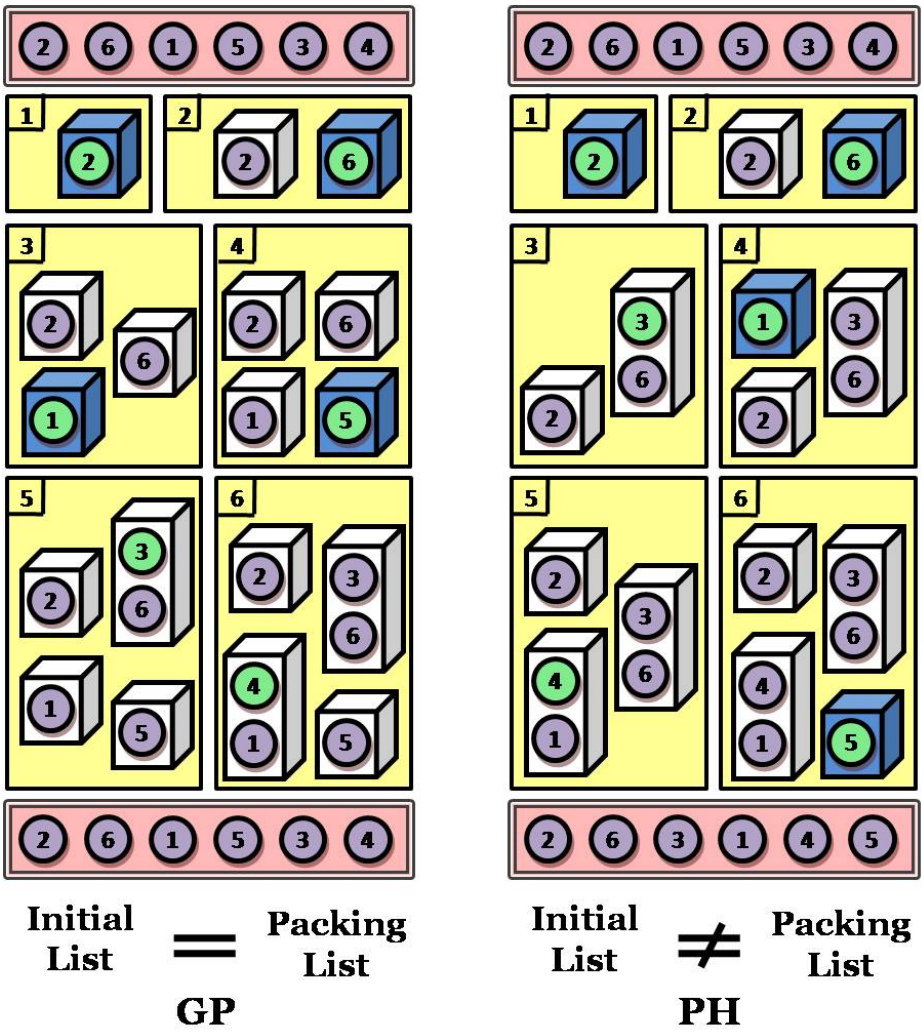

Figure 3.1: Left: example of a packing procedure using the GP strategy. Right: same example but using the PH strategy.

Figure 3.1 shows the packing procedure that the GP and the PH follows. The list on the top of the figures is the initial list and the list on the bottom is the packing list. The initial list contains the links to be scheduled sorted following an specific order and the packing list is created adding the links according to the order that they have been scheduled. Note that given the same list though the algorithms pack the links in a different order both algorithms results in the same scheduling. In particular, in the greedy physical, the initial list is the same as the packing list and in the packing heuristic, the initial list is not the same as the packing list. As previously mentioned, this does not lead to different schedules, but it is important to highlight what is the packing list since it will be an important aspect in section 4.2.3.

Up to this point, it has been shown that given an identic list both algorithms obtain equal schedules, despite following different scheduling strategies. Therefore, the question arose here is whether the strategy to which links are packed does affect to the execution time of the algorithms or not. The answer is that this fact does not affect to the final execution time, but it does affect in the number of calculations that each link needs to perform to be scheduled in a timeslot, in other words, the number of links that each link has to checked to be scheduled in a timeslot.

Firstly, a few assumptions need to be clarified. Remember that in section 2.3.3 we have seen that depending on which link is checked its feasibility, the number of calculations varies. However, for simplicity and without loss of generality, we consider herein that each calculation or each SINR checking has the same complexity, in other words, we use for each node the SINR expression 2.6 to check its feasibility. In addition, to check the feasibility of a set of links scheduled in timeslot t when a new link is added, we first check the feasibility of the new link. The order to which links are checked (first the new link and then the links already scheduled in the timeslot) has to be the same in both algorithms so the number of final calculations are equal. Empirically, it can be seen that approximately half of the times the new link is the one that does not accomplish the feasibility constraint. However, checking the new link first or last does not necessary lead to a reduction in the number of calculations since the complexity of this calculation is similar to the complexity of checking all the links already scheduled in timeslot t (using the β constraint described in section 2.3.3). Therefore, checking first the new link or the links already scheduled in a timeslot is similar in terms of calculations.

Figure 3.2 shows the cumulative function of the number of links that has to be checked to schedule each link for the Packing Heuristic and for the Greedy Physical. This figure has been obtained simulating 1000 different WMN's topologies with 60 nodes, but the same behavior holds for distributions with more or less nodes. This results are empirically since it is impossible to have an exact theoretical model due

to the number of calculations depends randomly on the distribution of the WMN.

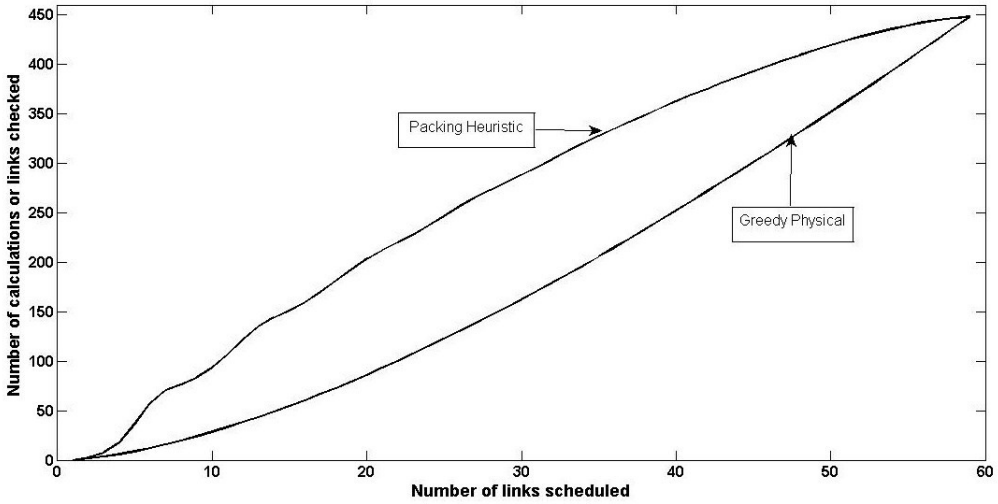

Figure 3.2: Cumulative function of the number of calculations or links checked to perform an schedule for the Packing Heuristic and the Greedy Physical.

Observe that the number of calculations for the PH follows a concave like function and for the GP follows a convex like function. The important fact to remark herein is that the strategy followed by the Packing Heuristic requires more calculations to schedule the initial links than the GP. An example on how to take advantage of this fact will be shown in chapter 4.

3.3 STDMA Link Scheduling using Graph-Based Network Model

In this section, we are going to formulate the scheduling problem in terms of graph-theoretic colouring problem. Previous research work focused on graph theoretic solutions by conceiving link scheduling as a graph-coloring problem can be found in the literature [14] [15] [16]. In this work, we will employ the Recursive Largest First algorithm proposed by Leighton [44] to find a timeslot allocation based on the protocol interference model. Note that in the protocol interference model, a communication between nodes u and v is successful if no other node within a certain interference range from v (the receiver) is simultaneously transmitting.

As graph colouring is the basis upon which the algorithm is performed, we recall shortly the statement of the coloring problem that is of interest in the scheduling context. For a further comprehension on basic graph theory we refer the reader to

[40]. Graph coloring serves as a model for conflict resolution in problems of the following type. Suppose that in a set V certain pairs of elements are incompatible. The problem is to find a partition of V into a minimal number of subsets of mutually compatible elements. The situation is described by a graph $G = (V, E)$ with vertex set V and edge set E formed by all pairs of incompatible elements. Eventually, partitioning of V into k subsets is equivalent to coloring the vertices of G with k colors.

The limitation of the graph theoretic tools stem from the fact that the aggregate effect of interference of links transmitting in concurrent timeslots (reflected in the signal-to-interference noise ratio (SINR)) is not taken explicitly into account [17]. Hence, a schedule provided by a graph-coloring technique (for instance, the RLF) may lead to an infeasible allocation when the SINR thresholds are taken into account. Since the aim of this chapter is to compare the performance of different schedule techniques, the schedule obtained by the RLF will be re-calculated until a feasible allocation is achieved. Thereby, we can fairly compare the different approaches presented in this chapter, as will be shown in section 3.4.

3.3.1 The Interference Graph (IG)

In order to apply the Recursive Largest First (RLF) to the scheduling problem it is needed to construct an interference graph, which is similar to the one that appears in [45]. The interference graph presented in this work is defined as follows. $G^{int} = (V^{int}, E^{int})$ is a graph consisting in a set of vertices V^{int} representing the links or edges of the communication graph $(G = (V, E) \mid V^{int} \subseteq E)$ and a set of edges E^{int}, connecting the links that are interfering to each other. Despite links have become into vertices in the interference graph, as a matter of notation, we will refer to elements (vertices) in V as links. Link v_1^{int} is interfering link v_2^{int} if the interference range, with center in the transmitter of link v_1^{int}, contains the receiver of link v_2^{int}. Note that, though v_1^{int} could not be interfering v_2^{int}, v_2^{int} could interfere v_1^{int} (if the transmitter of link v_2^{int} contains the receiver of link v_1^{int}) and the result is that v_1^{int} and v_2^{int} interfere to each other.

As it has been stated in section 2.1.4 we consider that links that share an endpoint are not able to communicate simultaneously. Therefore, links that share an endpoint in the communication graph are considered as if they interfere to each other, because they can not be scheduled at the same timeslot. As a matter of notation, two different links or edges that share an endpoint are called adjacent, as it is defined by R. Diestel in its graph theory book [40]. Furthermore, the set of all edges or links in E at a vertex or node v (or all the adjacent edges of node v) is

denoted by $E(v)$.

Algorithm 3: Pseudo-code of the Interference Graph

Input : $G = (V, E)$, Communication Graph.
: $Distance$, Distance matrix between each node in the network.
: R_0, Interference Range increment ($R_0 \leq 1$)
Output : $G^{int} = (V^{int}, E^{int})$, Interference Graph.

```
1:  V^int ← E
2:  for each node i in |V| do
3:    L_i ← E(v(i)), Adjacent links of node v(i).
4:    Create an edge e^int for each possible pair of links in L_i, where each link
      corresponds to a vertex in V^int.
5:    for each link j in |L_i| do
6:      if node v(i) is the transmitter of link L_i(j) then
7:        R_i ← R_0 * Distance(L_i(j) transmitter, L_i(j) receiver)
8:        for each node k in |V| do
9:          if v(i) ≠ v(k) then
10:           if Distance(v(i), v(k)) ≤ R_i then
11:             L_k ← E(k), Adjacent links of node v(k).
12:             for each link l in |L_k| do
13:               if L_i(j) ≠ L_k(l) and node v(k) is the receiver of link L_k(l)
                  then
14:                 Create an edge e^int between links L_i(j) and L_k(l), where
                    each link corresponds to a vertex in V^int.
15:               endif
16:             endfor
17:           endif
18:         endif
19:       endfor
20:     endif
21:   endfor
22: endfor
```

The interference graph can be constructed as stated in algorithm 3. Note that, the interference range R_i of the transmitter node of a link is the range within which other receiver (different than the receiver of the link) is interfered by the ongoing communication. The interference range is not a static value, but a function of the distance between the transmitter and receiver of a the active link. Thereby, we

define the interference range R_i of the transmitter of a link as R_0 times of the distance between the transmitter and receiver of the link. The parameter R_0 is always bigger or equal than 1 (a value of 1 will mean that the interference range of a node is equivalent to its transmission range). In our work, our goal is to find a value of R_0 within which the schedule achieved by the RLF will be feasible, as it is detailed in 3.3.2. Therefore, we will calculate different interference graphs for different values of R_0 until we manage to find a feasible schedule. Observe that, a transmitter node may have different interference ranges depending on which node is transmitting to. More specifically, the interference area consists of a circle of radius R_i in the transmitter of a link, as can be seen in figure 3.3. This figure shows a communication graph with the interference range of all the transmitter nodes and the resulting interference graph is presented in figure 3.4. Note that, the interference range considered in this example is an arbitrary value bigger than 1. Finally, note that an exhaustive analysis of this value will be described in section 3.3.3.

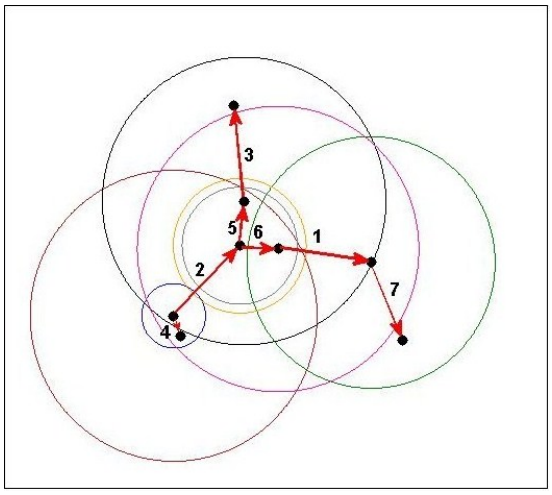

Figure 3.3: Communication Graph of a network composed by 8 nodes and 7 links.

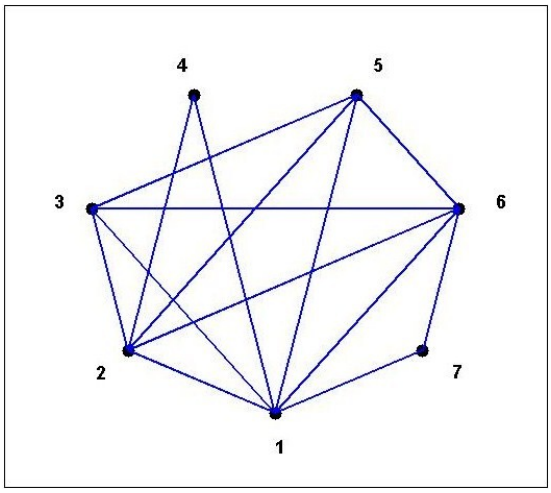

Figure 3.4: Interference Graph.

3.3.2 The Recursive Largest First (RLF)

The Recursive Largest First (RLF) is a graph colouring technique proposed by Leighton [44]. In this work, we apply the RLF to the interference graph as it is detailed next. RLF incrementally constructs an independent set $V_1^{int} \subset V^{int}$ of vertices that are coloured with the first colour, then, considering only vertices in $V^{int} \setminus V_1^{int}$, constructs the next independent set V_2^{int} as a second colour class and so on. The first vertex placed in V_i^{int}, $i \in \{1, 2, ...\}$, is a vertex of maximum degree in the subgraph induced by $V^{int} \setminus \cup_{j=1}^{i-1} V_j^{int}$. At each other step, the algorithm selects

a vertex with the maximum number of adjacent vertices that are uncoloured but already inadmissible for the i_{th} colour. Formally, the algorithm can be stated as follows.

Algorithm 4: Pseudo-code of the RLF algorithm

Input : $G^{int} = (V^{int}, E^{int})$, Interference Graph.
Output : C, A feasible schedule or colour assignment of the edges.
: TS, Number of colours of C or frame length of the schedule.

1: Set $C \leftarrow \emptyset$, $V' \leftarrow V_{int}$, $U \leftarrow \emptyset$, $q \leftarrow 0$.
2: Choose a vertex k of maximum degree in the subgraph induced by V'. Increment q by 1 and proceed to 3.
3: Assign colour q to k. Move k from V' to C and $\forall\ i\ \epsilon\ V'$ that are adjacent to k from V' to U. If V' remains nonempty, then proceed to 4. Otherwise check whether $C = V_{int}$. If so, then stop with G coloured with q colours. If not, then set $V' \leftarrow U$, $U \leftarrow \emptyset$ and return to 2.
4: Choose a vertex $k\ \epsilon\ V'$ that has the maximum number of edges to vertices in U. Go to 3.
5: Let $TS \leftarrow q$

As it was previously explained, the RLF may lead to infeasible timeslot allocations when the SINR thresholds are taking into account. The procedure to obtain a feasible allocation is detailed next. First, we create the interference graph (IG) corresponding to the communication graph. Thereafter, we apply the RLF algorithm to the IG and we obtain a schedule of length TS. Then, we check if the schedule obtained is feasible; a schedule is feasible only if all links of each timeslot have a SINR higher than a certain threshold. In case the schedule is not feasible, we increment the interference range of all nodes in order to obtain a new interference graph more restrictive aiming to get a feasible schedule. The election of the interference range increment is detailed in section 3.3.3. Eventually, this procedure is repeat until a feasible allocation is achieved. Note that incrementing the interference range of all nodes is a simple method to obtain a new interference graph, although other strategies might lead to better solutions. Algorithm 5 shows the procedure describe above.

3.3.3 The Interference Range Increment

In the previous section, we have seen that in order to obtain a feasible schedule using the RLF algorithm, we need to calculate different interference graphs for dif-

Algorithm 5: Pseudo-code of the feasible RLF

Input : $G = (V, E)$, Communication Graph.
: $\Delta_{R_O} \leftarrow 0.025$, Interference Range Increment.
Output : S, A feasible schedule.
: TS, Frame length found for S.

1: $R_0 \leftarrow 1$, Interference range constant. Initially the interference range is equivalent to the transmission range.
2: **repeat**
3: $G^{int} = (V^{int}, E^{int}) \leftarrow$ Obtain the Interference Graph from G and R_O.
4: S, $TS \leftarrow$ Apply RLF to G^{int}.
5: **if** *S is not feasible* **then**
6: $R_0 \leftarrow R_0 + \Delta_{R_O}$
7: **endif**
8: **until** *S is feasible*

ferent interference ranges. Hereafter, we are going to analyze how many iterations are required to obtain a feasible schedule depending on the interference range increment chosen. The interference range increment affects to the execution time of the algorithm and to the length of the timeslot allocation. Since the time consumption is an important issue when studying WMN's, we will chose a value of the interference range increment that will give us a good timeslot allocation without penalizing considerably the exectution time of the algorithm.

Figure 3.5 shows the Cumulative Distribution Function (CDF) of the number of iterations necessary to achieve a feasible schedule depending on the interference range increment chosen. In this example, we have initialized the interference range to 1, what means that initially the interference range is equivalent to the transmission range. Despite this example shows the CDF for a network composed by 20 nodes, similar results holds for networks with higher or lower number of nodes. Observe from this figure, that the number of iterations increases as the interference range increment decreases, as it was obviously expected. Table 3.1 shows that increasing the interference range increment entails a worse timeslot allocation.

In the numerical results, section 3.4, we will use increments of 0.025. As we can see in figure 3.5, the 90% of times the number of iterations required is 54. This is an acceptable number of iterations if we compare it with the case of increments of 0.010. Furthermore, the improvement achieved in number of timeslots by incrementing the interference range in 0.010% instead of 0.025% is negligible. On the other hand, if we choose a higher increment ,i.e., 0.05, the reduction of the number of iterations is not worth it compared to the increase in terms of timeslots (Table 3.1). Therefore,

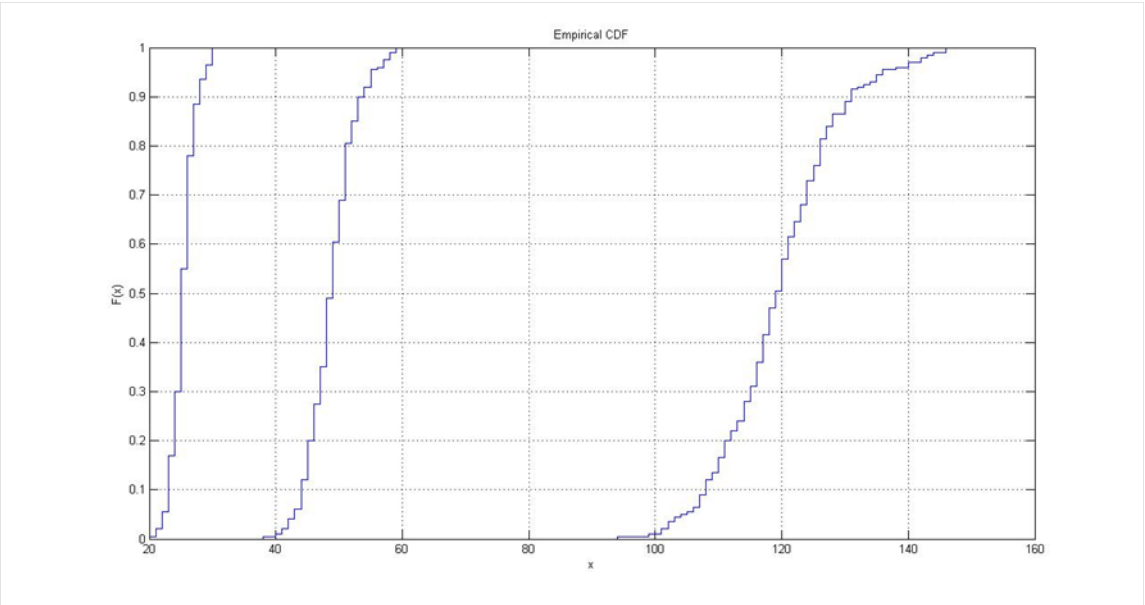

Figure 3.5: Empirical CDF for 20 nodes and for distance increments of 0.05, 0.025 and 0.01 (from right to left)

incrementing the distance by 0.025 seems to be a good value for satisfying both the number of timeslots and the number of iterations.

Table 3.1: Number of timeslots depending on the interference range increments.

Interference Range increments	**Number of Timeslots**
0.01	9.985
0.025	9.99
0.05	10.075

Finally, figure 3.5 shows another interesting fact; it is never found a feasible schedule in the first iterations. This means that the initial interference range could be initialized to a higher value without losing any solution. Therefore, the number of iterations needed and the time consumption could be reduced. However, the initial value of the interference range varies depending on the number of nodes in the network, the interference range increment and other parameters like the SINR, the minimum distance between nodes, etc. Hence, the number of useless iterations changes for each network and it is necessary an individual study for each case to reduce as maximum the time consumption of the algorithm.

3.4 Numerical Investigations

Numerical investigations have been conducted over 200 randomly generated wireless mesh network topologies with a single gateway in the topology which provides Internet connectivity. The simulation is set up according to what was explained in detail in chapter 2. The length of the square area is chosen aiming to study the behavior of the network for low and high node densities and at the same time not incrementing considerably the time consumption of the algorithms. The complete set of the simulation parameters used in the numerical investigations are summarized in Table 3.2 below.

Table 3.2: Simulation Parameters.

Notation	**Explanation**	**Values**
A	Length of the Square Area	700 meters
N	Number of Nodes	5 - 140
L	Number of Links	4 - 139
d_0	Close-in reference distance	50 meters
SNR	SNR required	15 dB
γ	SINR threshold	8 dB
η	Path loss exponent	3.5
P_{max}	Maximum transmitted power	20 Watt
f_c	Carrier frequency	3.8 GHz
W	Thermal & background Noise	-132 dBW

The performance on the number of timeslots achieved so far by the greedy STDMA link scheduling algorithms presented in this work is shown in figure 3.6. Observe that the frame length of the schedules augments as the number of nodes does. More in detail, adding new nodes to a small topology entails a considerable increase on the number of timeslots, though it is becoming less significant as the density of the network augments.

As we can see in figure 3.6, the Greedy Physical algorithm achieves the minimum frame length for most of the simulation. Figure 3.7 shows the difference between the GP and the rest of the algorithms. Observe that for small topologies (less than 10 nodes) the RLF obtains better timeslot allocations. For topologies with more than 10 nodes, the GP achieves less number of timeslots following a concave like

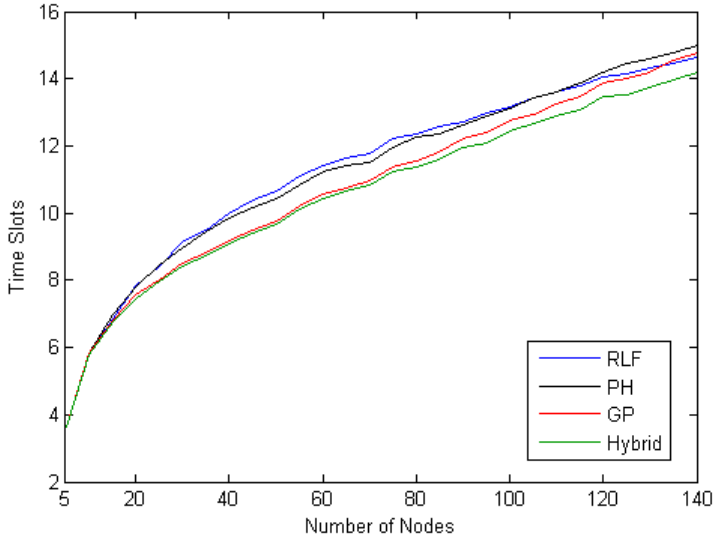

Figure 3.6: Packing heuristic (PH), Greedy physical (GP) and Recursive largest first (RLF) algorithms. Hybrid chooses for each topology the best solution between all the algorithms.

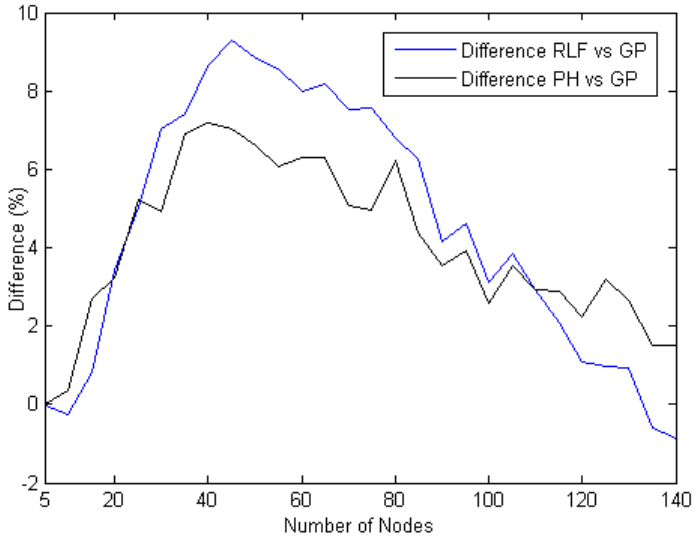

Figure 3.7: Difference between the RLF and the PH compared to the GP.

function. The maximum difference between the GP and the RLF is around 9% (taking the GP as the reference) when the number of nodes is 45 and between the GP and the PH around 7% for 40 nodes. From this point, we observe that this difference is becoming less significant as the number of nodes increases, becoming

the RLF even better than the GP, in terms of timeslots, for distributions with more than 135 nodes. Note that, the PH is never better than the GP for distributions with less than 140 nodes in this scenario.

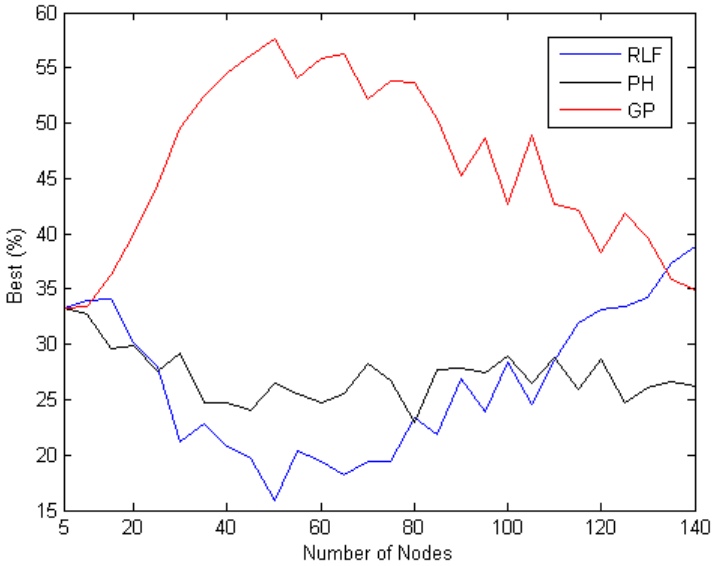

Figure 3.8: Best Algorithm

Eventually, another interesting fact to remark is that the best algorithm depends on the distribution of the nodes. Figure 3.8 shows the percentage of times that each algorithm finds the best solution, in other words, the minimum number of timeslots, over the 200 topologies simulated. Note that more than one algorithm could find the same best solution. This fact is also shown in figure 3.6 with the hybrid algorithm, which chooses for each distribution the best solution between the three scheduling algorithms. Despite the Greedy Physical obtains the best performance for most of the topologies, the limited information of a graph model leads to significant loss of spatial reuse as compared to the interference-based model in most of the cases, these results shows that there is not any algorithm that is always better than the others. Therefore, it is interesting to greedily modify all of them in order to achieve better timeslot allocations.

Chapter 4

Enhanced STDMA Link Scheduling

In chapter 3 we have seen the performance of different scheduling algorithms and hereafter we are going to ameliorate the proposed algorithms. In section 4.1.1 an enhanced version of the RLF is proposed, the Selective RLF Random algorithm (SRLFR). The first main contribution of this thesis is presented in section 4.2, the Randomized Link Swap Packing (RSP) algorithm.

4.1 Selective RLF Random

In this section, we aim to improve the performance of the feasible RLF algorithm (alg. 5) described in section 3.3.2. Two different approaches are designed in this section to enhance the original feasible RLF, a randomization and a selective interference range.

4.1.1 Selective RLF Random algorithm (SRLFR)

The first enhancement applied to the RLF is a simple randomization. We are going to generate different solutions just varying the initial link scheduled; instead of starting the algorithm scheduling the link with maximum degree, we will start with a random link and then we will proceed the link allocation as normal. Note that, another randomization may be applied into the RLF though this randomization achieves satisfactory results for a reduced number of solutions. In case of increasing

the number of solutions employed, other types of randomization might achieve better solutions. The pseudo-code of the RLF random is reported in algorithm 6.

Algorithm 6: Pseudo-code of the RLF random algorithm

Input : $G^{int} = (V^{int}, E^{int})$, Interference Graph
Output : C, a feasible schedule or colour assignment of the edges
: TS, number of colours of C or frame length of the schedule

1: Set $C \leftarrow \emptyset$, $V' \leftarrow V_{int}$, $U \leftarrow \emptyset$, $q \leftarrow 0$.
2: **if** *q=0* **then**
3: Choose a random vertex k. Increment q by 1 and proceed to 7.
4: **else**
5: Choose a vertex k of maximum degree in the subgraph induced by V'. Increment q by 1 and proceed to 7.
6: **endif**
7: Assign colour q to k. Move k from V' to C and $\forall\ i\ \epsilon\ V'$ that are adjacent to k from V' to U. If V' remains nonempty, then proceed to 8. Otherwise check whether $C = V_{int}$. If so, then stop with G coloured with q colours. If not, then set $V' \leftarrow U$, $U \leftarrow \emptyset$ and return to 2.
8: Choose a vertex $k\ \epsilon\ V'$ that has the maximum number of edges to vertices in U. Go to 7.
9: Let $TS \leftarrow q$

Recalling what was stated in section 3.3.2, to obtain a feasible schedule, the interference range (IR) was equally incremented in all nodes of the network each time an infeasible schedule was achieved. Thus, incrementing the IR of all nodes was not an efficient proceeding, since we omit several solutions which might obtain smaller timeslot allocations. The criterion to decide which nodes should be incremented their IR is of difficult election; incrementing only the IR of the node which maximum/minimum SINR or just incrementing the IR of a random node does not entail significant improvements and consequently the time consumption of the algorithm increases.

In the proposed scheme, the criterion employed to enhance the RLF algorithm is not to re-schedule all links each time an infeasible schedule is achieved, but schedule all links belonging to infeasible timeslots and some of the links allocated in feasible timeslots. The feasible timeslots selected to be re-scheduled are those ones with less links assigned. In case of having some timeslots with equal number of links, we choose between them randomly or if there are not enough timeslots, we select as many as possible. Therefore, it is only incremented the IR of those nodes that need to be re-scheduled and since the re-scheduling does not consider all nodes of

the network, the time consumed by the algorithm decreases. Figure 4.1 shows an example of the selective RLF, where only the selected links are re-scheduled and the mergence of this re-scheduled links with the links that does not need to be re-scheduled forms the new schedule. This procedure is repeated as many iterations as necessary until a feasible scheduled is achieved. The pseudo-code of the proposed scheme is reported in algorithm 7.

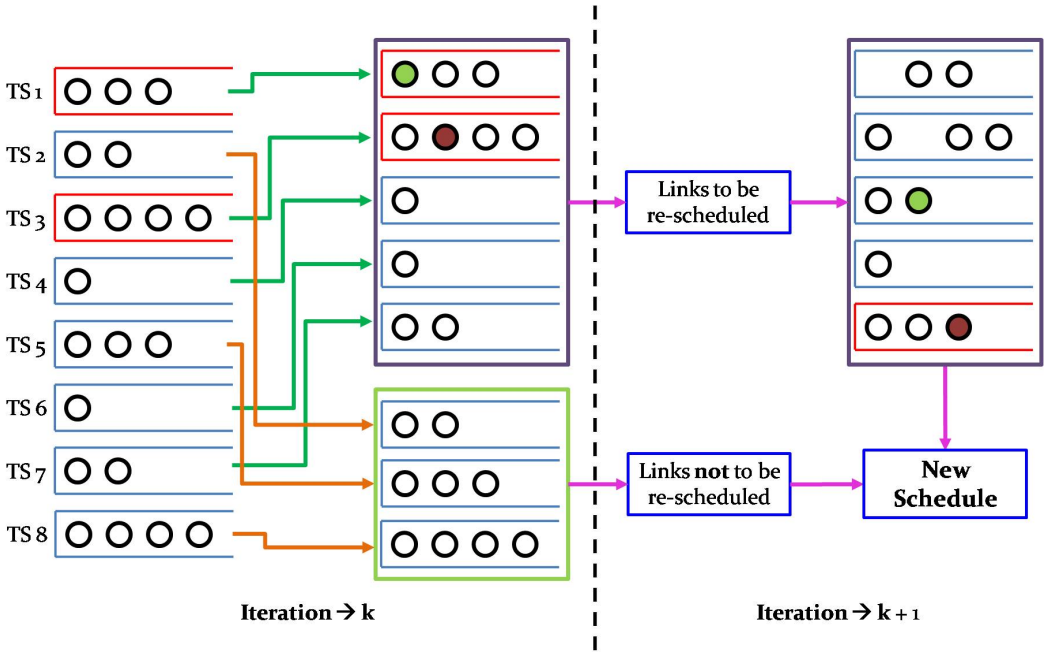

Figure 4.1: Example of the Selective RLF for a network of 20 links. In red, timeslots that are infeasible; in blue, timeslots that are feasible. Observe that the number of timeslots to be re-scheduled is 3 and therefore the number of links to be re-scheduled at iteration $k+1$ are now 11 instead of 20.

The number of feasible timeslots (N_{FTSR}) selected to be re-scheduled determines the performance of this algorithm. The N_{FTSR} varies depending on the topology of the network and the number of nodes, but for most of the cases simulated throughout the numerical investigations, the best schedules are obtained for values of N_{FTSR} between 1 to 4.

Different criteria can be applied in order to determine the best schedule when schedules with the same frame length as the best one found so far are generated. For instance, to improve the interference robustness of the network, possible criteria are (i) to choose the schedule with the best averaged SINR or (ii) the schedule with the maximum average min-SINR across all timeslots.

Algorithm 7: Pseudo-code of the Selective RLF Random (SRLFR)

Input : $G = (V, E)$, Communication Graph.
: $\Delta_{R_O} \leftarrow 0.025$, Interference Range Increment.
: $Sols$, Number of feasible solutions to be found.
Output : S, A feasible schedule.
: TS, Frame length found for S.
: $S_{BEST} \leftarrow \emptyset$, A feasible schedule with the minimum frame length
: found so far.
: $TS_{BEST} \leftarrow 0$, The minimum frame length found so far.
: $N_{FTSR_{min}}$,$N_{FTSR_{max}}$, Min/Max number of feasible timeslots to be
: re-scheduled.

1: $N_{FTSNR} \leftarrow \emptyset$, contains all feasible timeslots *not* to be re-scheduled.
2: $R_0 \leftarrow 1$, Interference range constant. Initially the interference range is equivalent to the transmission range.
3: **repeat**
4: **for** $N_{FTSR} \leftarrow N_{FTSR_{min}}$ **to** $N_{FTSR_{max}}$ **do**
5: **repeat**
6: $G^P = (V^P, E^P)$, Remove from G all links belonging to the set of timeslots N_{FTSNR}.
7: $G^{int} = (V^{int}, E^{int}) \leftarrow$ Obtain the Interference Graph from G^P and R_O.
8: $S^P, TS^P \leftarrow$, Apply the *RLF random* algorithm to G^{int}
9: S, $TS \leftarrow$ Join the links scheduled in timeslots N_{FTSNR} (timeslots kept from the last iteration) and the links in S^P. Compute the new frame length TS.
10: **if** *S is not feasible* **then**
11: $N_{FTSNR} \leftarrow$ Timeslots that are feasible but N_{FTSR}.
12: $R_0 \leftarrow R_0 + \Delta_{R_O}$
13: **endif**
14: **until** *S is feasible*
15: **endfor**
16: **if** $TS \leq TS_{BEST}$ **then**
17: $TS_{BEST} \leftarrow TS$
18: $S_{BEST} \leftarrow$ BestSchedule(S,S_{BEST})
19: **endif**
20: **until** *$Sols$ solutions found*

4.1.2 Numerical Investigations

We analyze the performance of the Selective RLF Random (SRLFR) by comparing it with the standard RLF (alg. 5) and the GP (alg. 1) which was the best scheduler

up to this point. Since we are employing a randomization technique in the proposed SRLFR algorithm, we will also compared our algorithm with a Random Packing Algorithm (RP), which is a randomization version of the Greedy Physical (GP) or the Packing Heuristic (PH). The RP consists on sorting the links randomly before packing the links as the GP or the PH does (remember that both algorithms generates the same schedule given an equal initial list). As the list is created randomly, it is necessary several iterations of the RP algorithm to achieve satisfactory timeslot allocations. More specifically, we will execute the RP algorithm the same number of iterations as the SRLFR, so both algorithms will consume a similar amount of time. The number of iterations is the number of times that the RLF random is executed inside the SRLFR algorithm. Thus, the RP scheduler is stated in algorithm 8. Observe that the best schedule is obtained applying the same criteria as used for the SRLFR algorithm.

Algorithm 8: Pseudo-code of the RP algorithm

Input : L, a list containing all links sorted radnomly.
: $Iter$, Number of Iterations.
Output : $S_{BEST} \leftarrow \emptyset$, A feasible schedule with the minimum frame length
: found so far.
: $TS_{BEST} \leftarrow 0$, The minimum frame length found so far.

```
1: for i ← 0 to Iter do
2:     S, TS ← Pack the links according to the GP or PH strategy.
3:     if TS ≤ TS_BEST then
4:         TS_BEST ← TS
5:         S_BEST ← BestSchedule(S,S_BEST)
6:     endif
7: endfor
```

We note that all numerical investigations have been conducted over 200 randomly generated wireless mesh network topologies with a single gateway in the topology which provides Internet connectivity. The simulation is set up according to what was explained in detail in section 2. The complete set of the simulation parameters used in the numerical investigations are summarized in Table 4.1.

Table 4.2 shows a comparison between the standard RLF, the GP, the SRLFR and the RP. Therein, we can see that the randomization versions of the RLF (the SRLFR) and the GP (the RP) obtains better results than the original algorithms. However, they require a huge number of iterations to achieve these results. In addition, we can see that the GP generates shorter frame lengths than the RLF, as seen in section 3.4; this behavior holds for their randomization version but the differ-

Table 4.1: Simulation Parameters.

Notation	Explanation	Values
A	Length of the Square Area	850 meters
N	Number of Nodes	10 - 90
L	Number of Links	9 - 89
d_0	Close-in reference distance	50 meters
SNR	SNR required	15 dB
γ	SINR threshold	8 dB
η	Path loss exponent	3.5
P_{max}	Maximum transmitted power	20 Watt
f_c	Carrier frequency	3.8 GHz
W	Thermal & background Noise	-132 dBW
Sols	Solutions	5
$N_{FTSR_{min}}$	Min. num. of feasible timeslots to be re-scheduled	1
$N_{FTSR_{max}}$	Max. num. of feasible timeslots to be re-scheduled	4

Table 4.2: The table shows the improvement in terms of timeslots (in percentage) of an algorithm respect another algorithm and the number of iterations employed.

Nodes	GP vs RLF	RP vs GP	RLF vs SRLFR	RP vs SRLFR	Iterations
10	0,25%	0,34%	0,59%	0,00%	475
20	2,67%	3,72%	5,72%	0,61%	660
30	5,16%	5,09%	8,65%	1,46%	751
40	7,53%	5,60%	10,76%	2,18%	815
50	8,33%	6,09%	11,69%	2,52%	865
60	8,81%	5,77%	11,80%	2,57%	900
70	9,12%	5,49%	11,16%	3,32%	935
80	8,17%	5,92%	11,84%	2,01%	960
90	7,97%	5,70%	11,92%	1,48%	978

ence in this case has been reduced. Hence, independently of the number of nodes, the RP algorithm obtains the best schedules. Finally, these results suggest that a greedy randomization might reduce even more the frame length of the schedules in a shorter time, since the RP does not utilize any useful information to reduce the solution space. Thus, a greedy randomization is detailed in the next section.

4.2 Fast Randomized STDMA Link Scheduling

4.2.1 Introduction

In the previous section, we have seen how a randomization can help to decrease the frame length of the network. However, no logic in the randomization would require to many iterations to achieve significant results. For instance, Simulated Annealing (SA) is a random-search technique, which can be regarded as a variant of local search, that was first introduced by Metropolis [46] and then used to optimization problems by Kirkpatrick [47]. The basic idea in SA is to track a path in the feasible solution space of the given optimization problem trying to avoid local optima by accepting probabilistically moves to worse solutions. However, one of the problems of this method is that the parameters of the algorithm are difficult to adjust such as the design of a suitable cost function (which determines which solutions should be kept or discard) for the scheduling problem. Furthermore, the complexity of an scheduling algorithm that uses the SA technique compared for instance with the GP is too elevated.

In this section, we are going to develop a random-search algorithm similar to a Local Search algorithm. This class of heuristics iteratively improves an initial feasible solution using local search technology. In our case, we will also search the solution space of a proven good solution though herein every new solution will be obtained using the original solution. Thus, in the following section is detailed the Randomized Link Swap Packing (RSP) algorithm designed herein to reduce the frame length of the network. Finally, in [48] a randomized distributed STDMA scheduling algorithm (DRAND) is developed to solve the scheduling problem utilizing a randomized method. The difference with our proposed scheme is that DRAND does not take the $SINR$ constraints into account. Also the randomization has a different rational compared to the proposed RSP algorithm. In DRAND the randomization is on how neighbor nodes are selecting timeslots, whereas in RSP the randomization is on how to deviate from an already feasible allocation and search alternative feasible (hopefully better) solutions.

4.2.2 Randomized Link Swap Packing (RSP) Algorithm

A very fast randomized link scheduling algorithm for STDMA wireless mesh networks that is build upon previously proposed greedy scheduling schemes is proposed in this section. The RSP algorithm is based on altering the interference number list by swapping N_s (number of swaps) times the order of two elements selected randomly from the list. The number of swaps applied to the list characterizes the degree to which the original list is distorted. After the swapped list is generated, the links are scheduled according to the GP or PH algorithms as described in the previous sections. Hence, a new feasible schedule is obtained. The same criteria applied in algorithm 7 in order to determine the best schedule when schedules with the same frame length as the best one found so far are generated can be applied herein. This process is repeated for a pre-defined number of iterations (M_{iter}). The pseudo-code of the proposed RSP scheme is shown in algorithm 9.

As can be observed from algorithm 9, the proposed RSP algorithm can be easily parallelized and run in P processors. In fact, the RSP algorithm can run without requiring any communication between the different processors, therefore there is no communication cost or delay for exchanging information between the different processors. Hence, the RSP algorithm enables embarrassingly parallel computations [49] since different schedules can be calculated independently, offering a convenient way to use multiple processors concurrently to solve the problem.

We note that a brute force enumeration of all possible ways to pack the $N-1$ links in a predefined number of timeslots would be $(N-1)!$. But since both the GP and the PH heuristics provide good initial feasible solutions, few iterations of the above proposed approach can provide significant benefits. As shown in section 4.2.4, the gains with the number of iterations follow a concave like function, which means that the net benefit of performing higher number of iterations diminishes with the number of iterations.

4.2.3 Calculation Savings

As mentioned above, the RSP algorithm modifies a list by swapping N_s elements and then proceeds to obtain a new feasible schedule. Observe that it is not necessary to re-schedule all links each time a new list is generated after applying the N_s swaps independently of using the GP or the PH algorithm. The set of links, from the first link until the previous link of the first link that has been swapped in the Packing List (which contains all links sorted by the order in which they have been packed, as explained in section 3.2.3), will remain in the same timeslots as the original schedule

Algorithm 9: Pseudo-code of the RSP

Input : L, A list containing all links sorted by its interference number, or
: power levels.
: N_s, Number of swaps.
: M_{iter}, Maximum number of iterations.
: P, Number of Processors.

Output : $S_{BEST,p}$, A feasible schedule with the minimum frame length
: found so far at processor p.
: $T_{BEST,p}$, The minimum frame length found so far at processor p.

1: $S_{BEST,p}, T_{BEST,p} \leftarrow$ Schedule(L)
2: **for** *each processor* **do** in parallel
3: **for** $i \leftarrow 1$ **to** $\left\lceil \frac{M_{iter}}{P} \right\rceil$ **do**
4: $L_{SWAP} \leftarrow L$
5: **for** $j \leftarrow 1$ **to** N_s **do**
6: $L_{SWAP} \leftarrow$ Swap two elements from L_{SWAP}
7: **endfor**
8: $S, T \leftarrow$ Schedule(L_{SWAP})
9: **if** $T \leq T_{BEST,p}$ **then**
10: $T_{BEST,p} \leftarrow T$
11: $S_{BEST,p} \leftarrow$ BestSchedule(S,$S_{BEST,p}$)
12: **endif**
13: **endfor**
14: **endfor**
15: $T_{BEST} \leftarrow \min\{T_{BEST,\forall p}\}$
16: $S_{BEST} \leftarrow$ BestSchedule$\{S_{BEST,\forall p}\}$

calculated. Therefore, it has sense to use the information of the original schedule to reduce the number of calculations of the new schedule.

The average number of links that does not have to be re-scheduled is determined by (4.4). Obviously, it is necessary to previously calculate the probability density function (pdf) shown in (4.1). In order to obtain the pdf, it is calculated in (4.2) the probability of saving at least k links after applying N_s swaps and in (4.3) the probability of saving at least more than k links after applying N_s swaps. Each swap is an independent event, hence the probability of saving at least k links after applying N_s swaps is the probability of saving at least k links for one swap and raise it to the power of the number of swaps applied, N_s. The probability of saving at least k links for one swap is the probability of selecting a link from the set L-k (as shown on the left side of figure 4.2) and then selecting another different link from

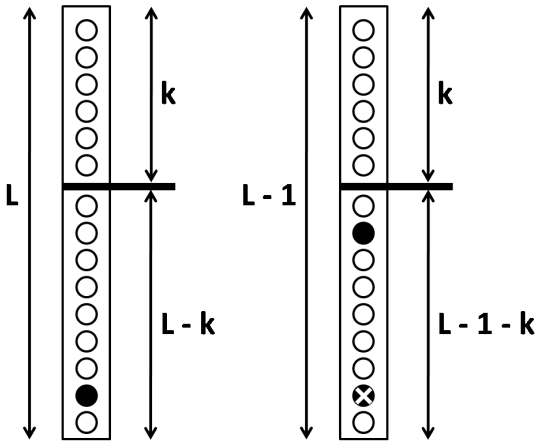

Figure 4.2: Left, Packing list after selecting one link (in black) to apply a swap. Right, Packing list after selecting another different link (in black) to apply the swap.

the same set (as shown on the right side of figure 4.2).

$$P(S = k) = P(S \geq k) - P(S > k) \;\; | \;\; 0 \leq k < L - 2 \tag{4.1}$$

$$P(S \geq k) = \left(\frac{L-k}{L} \cdot \frac{L-1-k}{L-1} \right)^{N_s} \tag{4.2}$$

$$P(S > k) = \left(\frac{L-k-1}{L} \cdot \frac{L-1-k-1}{L-1} \right)^{N_s} \tag{4.3}$$

$$\overline{S} = \sum_{k=0}^{L-2} k \cdot P(S = k) \tag{4.4}$$

Table 4.3 shows the average number of links that may be saved after one execution of the RSP algorithm for a different number of swaps and different nodes. Note that the maximum number of saving is $L - 2$ since it's necessary at least two links to perform a swap. The number of links saved increase as the number of nodes in the network increase and decrease as the number of swaps increase. Since the iteration reduction achieved decreases as the number of swaps augments, depending on the size of the network it may be more convenient to apply a lower number of swaps and execute the RSP algorithm more times to obtain a better schedule.

Table 4.3: Average number of links saved

Number of Swaps	N = 20	N = 40	N = 60	N = 80	N = 100	N = 120
1	6.00	12.67	19.33	26.00	32.67	39.34
3	2.31	5.15	8.01	10.86	13.72	16.58
5	1.31	3.11	4.92	6.74	8.55	10.37

Finally, let us recall some statements stated in section 3.2.3. First, given an equal initial list, both algorithms obtain identic schedules despite of using different packing procedures. Secondly, both algorithms utilize the same number of calculations to schedule all links, but the initial links scheduled in the PH strategy require more calculations than the GP strategy. Thus, since the initial links are the links that does not need to be re-scheduled, the PH strategy will re-schedule the links faster than the GP strategy. In conclusion, the GP should pack the links as the PH to re-schedule the links faster.

4.2.4 Numerical Investigations

We evaluate the performance of the RSP by comparing it with the two well known and tested greedy STDMA scheduling schemes that utilize the physical interference model, namely the Greedy Physical [18] and the Packing Heuristic [7] algorithms as have been explained in detail in section 3.2.1 and 3.2.2 respectively. We note that all numerical investigations have been conducted over 200 randomly generated wireless mesh network topologies with a single gateway in the topology which provides Internet connectivity. The simulation is set up according to what was explained in detail in chapter 2. The complete set of the simulation parameters used in the numerical investigations are summarized in Table 4.4.

The quality of the solution provided by the RSP algorithm scheme (T_{RSP}) is compared to the corresponding solutions from the GP (T_{GP}) and the improvement, denoted as (**I**) is measured as follows,

$$\mathbf{I}(\%) = \frac{T_{GP} - T_{RSP}}{T_{GP}} \tag{4.5}$$

The same measure is used to compare the solution quality of the proposed RSP algorithm with the PH (T_{PH}) algorithm.

Table 4.4: Simulation Parameters.

Notation	Explanation	Values
A	Length of the Square Area	850 meters
N	Number of Nodes	20 - 120
L	Number of Links	19 - 119
d_0	Close-in reference distance	50 meters
SNR	SNR required	15 dB
γ	SINR threshold	8 dB
η	Path loss exponent	3.5
P_{max}	Maximum transmitted power	20 Watt
f_c	Carrier frequency	3.8 GHz
W	Thermal & background Noise	-132 dBW

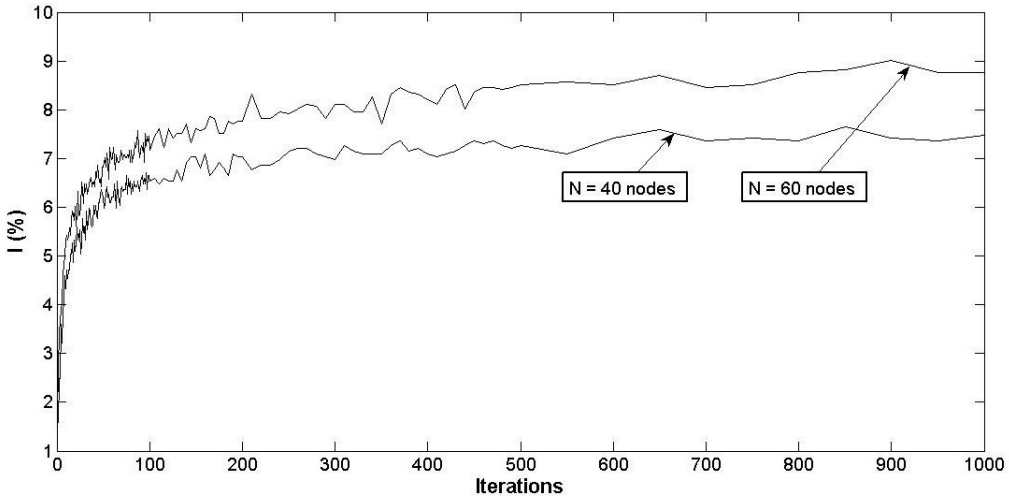

Figure 4.3: Performance gains on the minimum frame length using the RSP algorithm compared to the GP algorithm with respect to the number of iterations for topologies with 40 and 60 nodes. These results have been calculated using 3 swaps

Fig. 4.3 shows the performance gains on the minimum frame length using the proposed randomized scheduling scheme compared to the Greedy Physical algorithm with respect to the number of iterations. Observe that substantial improvements can be achieved with a reduced number of iterations, for instance, with just 15 iterations the schedule allocation is ameliorated above 5 % for topologies with 40

and 60 nodes. This improvement is even better as the number of iterations increases. However, it is becoming less significant as the number of iterations augments.

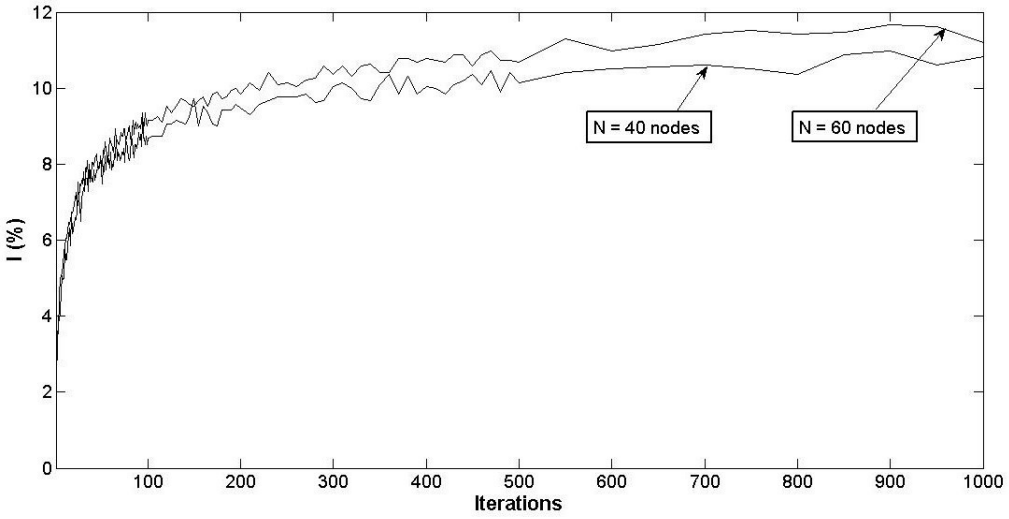

Figure 4.4: Performance gains on the minimum frame length using the RSP algorithm compared to the PH algorithm for topologies with 40 and 60 nodes (3 swaps are assumed).

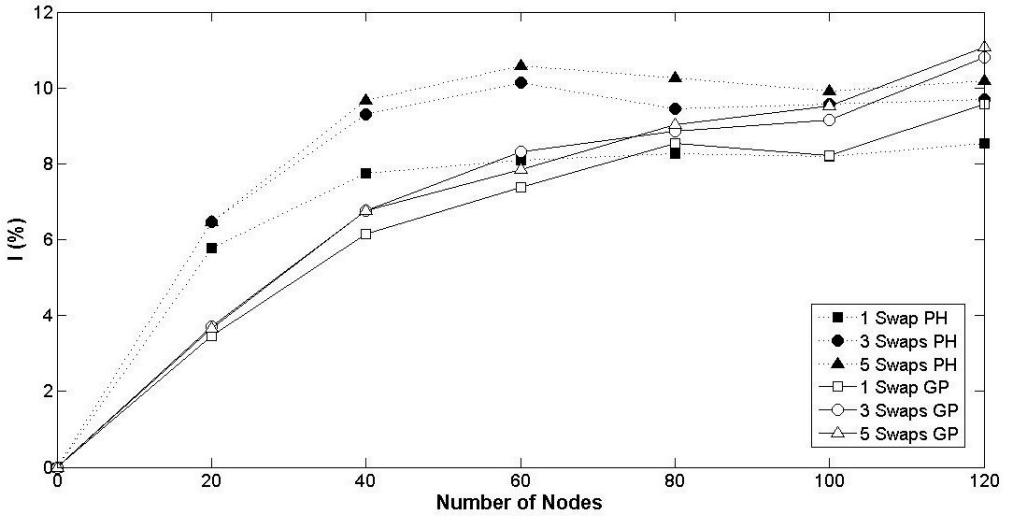

Figure 4.5: Performance gains on the minimum frame length using the RSP algorithm (with different number of link swaps) compared to the GP and PH.

Note that the same behavior holds when the RSP is applied to the Packing Heuristic, as Fig. 4.4 shows. In this case, the gain obtained is slightly higher

and, in consequence, with less than 10 iterations we achieve an improvement above 5 %. Figure 4.5 describes the performance improvement on the minimum frame length using RSP (with different number of link swaps) compared to the GP and PH for different number of nodes in the network. As has been mentioned above, the number of swaps applied to the list influences the degree to which the original list is distorted. Figure 4.5 shows that it is only necessary a small number of swaps to achieved a significant reduction on the frame length; for a bigger number of swaps the performance stops increasing. Note that the number of swaps can vary depending on the number of links on the list (as the number of links increases, the number of swaps to achieve better results needs to be increased) or on the scheduling algorithm employed.

Furthermore, the improvement achieved using the RSP algorithm is even better than the improvement achieved by the Random Packing algorithm (RP) shown in the previous section, which was the best scheduler up to that point. Finally, we should note that the iterations of the algorithm can be parallelized and therefore to optimally utilize multi core processor units. This is a crucially important feature of the proposed scheme since it is now widely accepted that the number of cores to even double every two years creating in that respect a need to design scheduling algorithms that can be easily parallelized.

Chapter 5

Reduction of Antenna Beam Switchings

5.1 Introduction

A set of algorithms that minimize the overall number of required beam switches in a wireless mesh network with switched directional antennas without penalizing the spatial reuse of timeslots are proposed in this chapter. Recalling section 1.3.1, the change of radiation pattern in an antenna is commonly called in the literature as a *switch*. Therein lies the aim of this chapter since each time a switch is performed, the beam switched antenna consumes energy (≈ 40 μW) and requires an amount of fixed time (between 5 μs to 0.25 ms) to stabilize the new radiation pattern. The characteristics of the antennas employed throughout this chapter are outline in section 2.2.

Observe that in this chapter, links are scheduled using the Greedy Physical algorithm, which is explained in detail in section 3.2.1, but the proposed techniques for reducing the number of beam switchings are actually independent of the scheduling algorithm. This chapter is organized as follows. Section 5.2 describes how link scheduling and antenna beam switching are intertwined and examples are illustrated to aid the understanding. In section 5.3 the proposed set of algorithms for minimizing the overall number of beam switching in the network are described. Finally, numerical investigations are reported in section 5.4.

5.2 The Effect of Link Scheduling on Antenna Beam Switching

The pattern and order of the beam switchings for a specific node in the network depends on the link scheduling algorithm. For every beam switch, the antenna consumes energy and requires an amount of time to stabilize to the new radiation pattern. Therefore, reducing the number of beam switching increases the performance and robustness of the network.

To illustrate how the order of the link scheduling algorithm itself affects the number of beam switchings in a node we proceed with an example. Figure 5.1 shows a node equipped with an antenna that can form 4 different beams (each of 90°) and has 4 incoming or outgoing links. The number of beam switches varies depending on the timeslot allocation, i.e., the link scheduling. Since there are four links these need to be scheduled in different timeslots. In this scenario, it is assumed that links e_1 and e_2 use a common beam to establish a communication and the other two links, i.e., e_3 and e_4 utilize another (common) beam to communicate. In the upper part of figure 5.1, links are active following the sequence e_1-e_2-e_3-e_4. As links e_1 and e_2 use the same antenna beam and are activated consecutively, there is no antenna beam switching between these transmissions. The same argument applies for links e_3 and e_4. On the other hand, the antenna must always apply a beam switch between links e_2 and e_3 as they utilize different radiation patterns and are activated consecutively; the same applies for links e_4 and e_1 taking into account that after link e4 is activated, link e_1 will be the next link to be activated. Therefore, in this case the antenna will need to apply two beam switchings. The same links as described above, but activated in different order e_1-e_3-e_2-e_4 are presented at the bottom part of figure 5.1. In this case, in each timeslot the antenna needs to change the previous radiation pattern. Hence, the antenna applies a switch between each pair of links, even between links e_4 and e_1, as it was explained in the previous example. As a consequence, in this case four beam switches are required by the node.

This example shows how an antenna with just two different active radiation patterns can have a different number of beam switchings depending on the order the links have been scheduled by the scheduling algorithm in the mesh network. On the other hand observe that, for a given timeslot allocation, the number of switches might decrease as the number of different pattern in an antenna does. In addition, despite the fact that in this example the antenna utilize a single beam for every link activation, the same procedure of counting the number of switches applies for antennas where simultaneous beams are activated at the same time. In the next sections the proposed algorithms consider more generalized cases compared to the

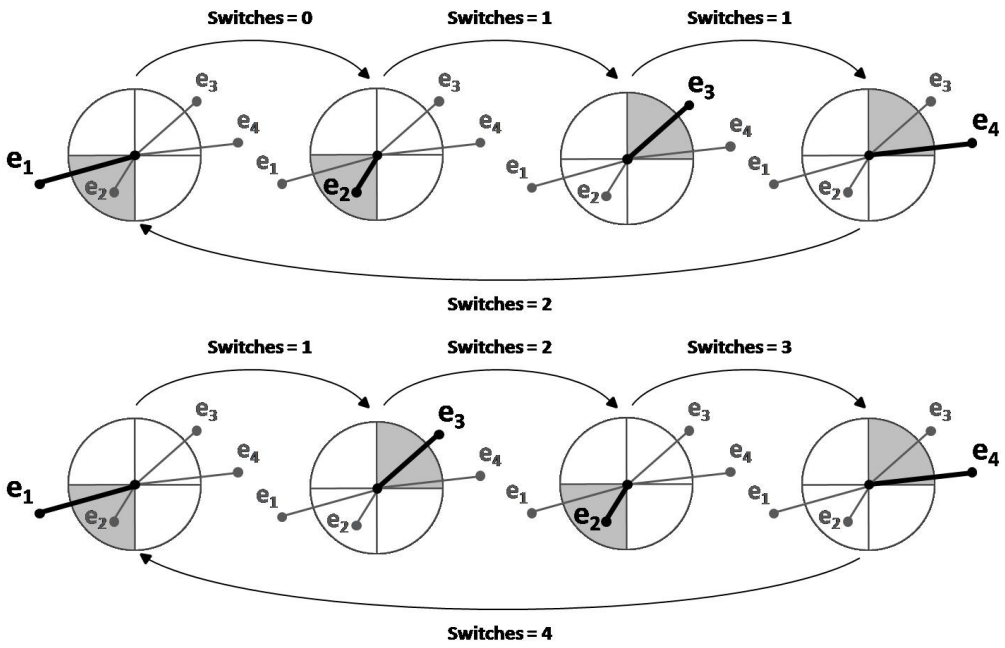

Figure 5.1: Sequences of radiation patterns with 2 and 4 switches. Note that the links in bold represents the links that are active each time.

above example for reducing the number of beam switching by also joining different beams without though affecting the feasibility of each link transmission.

The result below provide an upper bound on the worst-case scenario regarding the interaction between link scheduling and beam switching.

Lemma 3 *For a node i with active beams $B_i > 1$ and degree $D_i > 2$, the number of beam switch reductions is bounded by $D_i - 2$.*

Proof 3 Using figure 5.2 it can be seen that in the worst case scenario where the scheduling order of links is as follows $1 \rightarrow 2 \rightarrow 3 \rightarrow \ldots \rightarrow D_i - 1 \rightarrow D_i$ the number of beam switchings is equal to D_i (or $D_i - 1$ if D_i is an odd number). The minimum number of beam switchings is 2; this is the case when the scheduling order is as follows, $2 \rightarrow 4 \rightarrow 6 \rightarrow \ldots \rightarrow D_i \rightarrow 1 \rightarrow 3 \rightarrow \ldots \rightarrow D_i - 1$.

Using the above result, the upper bound on the number of beam switch reductions in a WMN with N nodes is

$$\sum_{i=1|D_i>2}^{N} (D_i - 2) \tag{5.1}$$

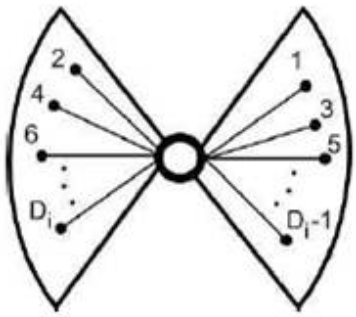

Figure 5.2: Upper bound on the number of beam switchings when scheduling is agnostic on the pattern of the beam switchings at the antenna.

5.2.1 Complexity of finding the minimum number of beam switchings in the network

Finding in a brute force manner the minimum number of beam switchings that can still produce a feasible STDMA scheduling can only take place for small wireless mesh network topologies since the complexity increases exponentially with the number of nodes in the network.

Table 5.1 shows the different possible antenna configurations for a node with 3 and 4 beams. In the table the horizontal axis is the timeslots and the beam configuration is encoded as a binary string where binary 1 and 0 denotes if the beam at the specific timeslot is active or not respectively. The bold binary strings denote the case where a node has 3 beams. Note that the number of configurations might differ from the number of possible antenna patterns as can be seen, for instance, for the case of 3 beams.

Figure 5.3 gives an example of the combinatorial explosion of searching possible different antenna patterns for perfect binary trees[1]. Clearly, a brute force enumeration of all possible beam configurations to find a configuration with the minimum number of beam switchings can only take place for topologies with small number of nodes. In the following sections we detail algorithms which their complexity increases polynomially with the number of nodes in the network.

[1]The average degree of a perfect binary tree is $2 - \frac{2}{N}$, where N is the number of nodes in the network. This closely resembles the average degree of the randomly generated shortest path tree topologies in section 2.1.3.

Table 5.1: Binary Encoded Antenna Beam Configurations for 3 (in bold) and 4 Beams.

Timeslots →	**1**	**2**	**3**	**4**
1	**100**|0	**010**|0	**001**|0	0001
2	**100**|0	**011**|1	**011**|1	0111
3	**010**|0	**101**|0	**101**|0	0001
4	**001**|0	**110**|1	**110**|1	1101
5	**111**|0	**111**|0	**111**|0	0001
6	0010	1100	1100	0001
7	0100	1001	1001	0010
8	0100	1011	1011	1011
9	0101	0101	1010	1010
10	0110	0110	1001	1001
11	1000	0011	0011	0100
12	1000	0101	0101	0010
13	1000	0110	0110	0001
14	1100	1100	0011	0011
15	1111	1111	1111	1111

5.3 Antenna Beam Joining Strategies

It is well known in the literature ([22],[50],[51]) and as it will also be demonstrated in section 5.4, that the performance of a multi hop wireless network increases when the antenna beam width is decreasing. This improvement is due to the fact that directional antennas improve the spatial reuse of the network. As the beam width of the antennas decreases it is expected that the number of beam switchings will be increased. Therefore, reducing the number of beam switchings becomes more important in the cases where the spatial timeslot reuse in the network is high.

Initially each link in the network is assigned one radiation pattern for the receiving antenna and one for the transmitting antenna and each pattern is initially formed by only one beam. The intuitive rational is that when the number of different beam patterns in an antenna decreases, it is expected that the number of beam switches is likely to decrease too. Hence, minimizing the number of different

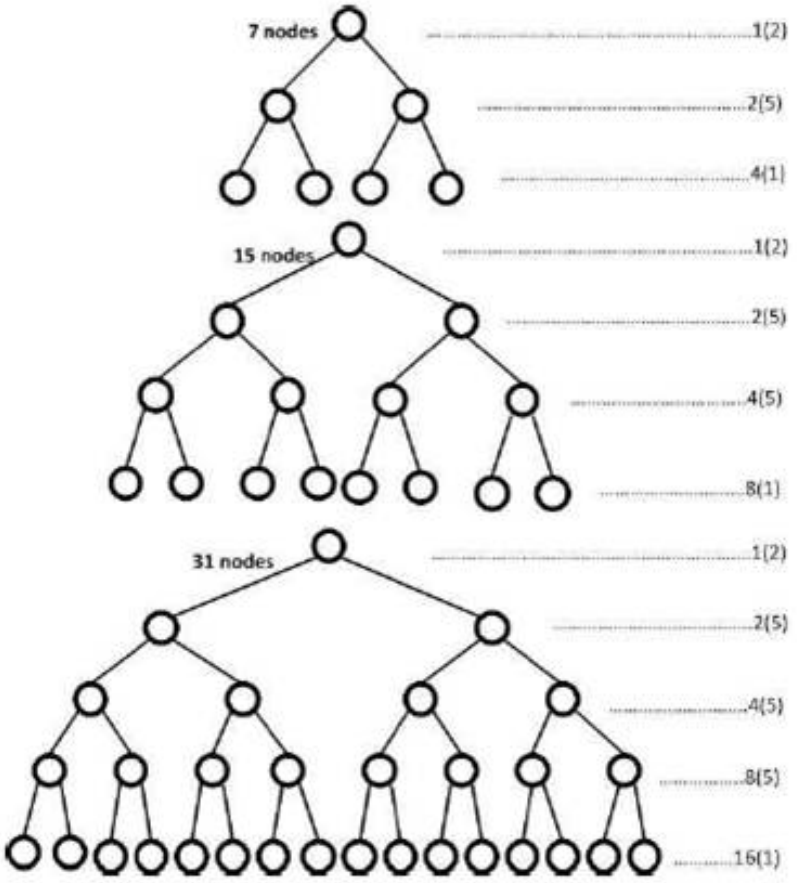

Figure 5.3: Number of antenna configurations for perfect binary tree topologies with different number of nodes. In the notation $A(B)$, A expresses the number of nodes in that binary tree level and B is the number of antenna configuration for these nodes. For the topology with 7 nodes the number of possible antenna configurations are $C_7 = 2 \cdot 5^2$, for the network with 15 nodes are $C_{15} = 2 \cdot 5^6$, while for the case of 31 nodes are $C_{31} = 2 \cdot 5^{14}$.

patterns of an antenna, or in other words, allowing multiple beams per node to be activated at the same time, is the main strategy for reducing the number of beam switchings. We further assume that the timeslot allocation is not feasible when all nodes operate in an omnidirectional mode (that case would negate the need to decrease the number of beam-switchings). An absolute upper bound on the number of beam-switchings, bs, can be derived from the degree, D_i, of the nodes in the network. It can be easily shown that the following inequality holds,

$$1 \leq b_s \leq \sum_{i=1}^{N} D_i \cdot \mathbf{I}(D_i) \tag{5.2}$$

where $\mathbf{I}$(x) is the indicator function defined as follows,

$$\mathbf{I}(x) = \begin{cases} 1 & \text{if x} > 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Case Study: For $N = 50$ nodes in a wireless mesh network and assuming an average degree in the network equal to 2 then $b_s = 100$. To put this into perspective, if we further assume the links are packed by the STDMA scheduling algorithm in 10 timeslots and that each timeslot has a duration of 2.5 msec[2] then in this wireless mesh network we would have 4000 beam switchings per second.

To reduce the number of beam switching in the network suitable beam joining should be performed at each node. The consequence of joining different beams implies that when a node i wants to transmit to a neighbor node j, its power will be spread along several beams of the antenna instead of just along one beam, i.e., the beam that link (i, j) belongs to. As a result, there will be an increase of the interference in the regions where there was not any active beam before the joining. Increasing levels of interference across the network caused by joining different beams of an antenna may create an infeasibility on the timeslot allocation.

In this work two different approaches are presented aiming of reducing the aggregate number of beam switchings in the network, while at the same time ensuring that the scheduling remains feasible. In the first strategy, beams belonging to the antennas with lower number of different radiation patterns are selected to be joint first, as will be detailed in 5.3.1. In the second strategy, beams are sorted in increasing order by their interference number(IN), which is explained in 5.3.2. Another important criterion is how to validate if the requirements of the network are still satisfied in case that two different radiation patterns are activated at the same time. Two different approaches can be applied whether a fixed scheduling of the links is considered or whether a rescheduling of them is allowed.

Sections 5.3.1 and 5.3.2 detail the proposed beam joining algorithms based on the pattern order and interference number strategies, when both of them employing a fixed schedule. Finally, section 5.3.3 describes how to modify the previous algorithms to allow rescheduling of the links.

[2]Frame length based on IEEE 802.16e transmission characteristics.

5.3.1 Beam Joining algorithm with Pattern Order

The first algorithm presented in the sequel tries to join firstly the patterns of those antennas that initially have less different number of patterns. The rational is to minimize the number of beams at the node (after joining as many pointing beams as possible) since the power emitted will be spread in less directions. Furthermore, when two antennas have the same number of different patterns, the antenna with lower degree (number of incoming and outgoing links of an antenna) is sorted first. The purpose of this distinction is to initiate merging firstly the patterns of those antennas that affect less timeslots and as a consequence they are more likely to interfere with less links.

The pseudocode of the proposed scheme is shown in Algorithm 10. The algorithm, starts by selecting a node from the list N, in which nodes are sorted in increasing order of the number of different patterns that they have; in case of being equal by the degree of the node. Thereafter, a list Ls is created with the incoming and outgoing links of the selected antenna. At this point, two links are select from list Ls and it is checked if the associated patterns of these links are suitable to be joined, i.e., if they are different.

Once two patterns are suitable candidates the Pattern list is updated. Note that originally each link in the network uses only one beam (or a pattern with a single beam) to transmit and one beam to receive. The Pattern list initially contains the beam associated to each link, where from the first element until $|E|$ identifies the beam of the transmitter of each link and from $|E|+1$ until $2|E|$ contains the beam that links utilize to receive. Furthermore, initially we add to each link the pattern of those links that uses the same radiation pattern in a common antenna, since links that uses equal patterns in the same antenna are initially considered merged. Then as the patterns of two links are merged, the Pattern list is updated adding to each beam that is associated to these two links, the rest of beams that are associated to these links. For instance, we want to join the transmitter pattern of link m, formed only by beam m, with the receiver pattern of link n, formed only by beam $n+|E|$. That means that link m will use to transmit the beams m and $n+|E|$ and so n will use $n+|E|$ and m to receive. Therefore, the set of beams merged together, forms a new pattern. Assume for instance that we want to join the transmitter pattern of a link o (link o uses the beam o) with the transmitter pattern of link m. As a design criterion of the algorithm, if we want to join the pattern of link o with the pattern of link m which is already joined with the pattern of link n, link o has to be also joined with n. Therefore, links m,n and o will use to communicate the initial patterns m, $n+|E|$ and o activated at the same time.

After a new Pattern list is found the gains are updated. All patterns that have

Algorithm 10: Beam Joining with Pattern order

Input : A graph $G = (V, E)$, $E(v)$ denotes the set of all edges in E at a
: vertex v, as defined in [40].
: N, List containing all nodes (antennas) sorted by the number of
: different patterns in increasing order and if they are equal by
: degree.
: $Ls \leftarrow \emptyset$, List containing the incoming and outgoing links of a
: node
: S, A feasible schedule with length TS.
: $Gains$, gains of all patterns.

Output : $Patterns$, List of all patterns and their respective associated
: patterns that are activated at the same time.

```
1:  for i ← 1 to |N| do
2:      Ls ← E(N[i])
3:      a ← 1
4:      for j ← 1 to |Ls| do
5:          a ← a + 1
6:          for k ← a to |Ls| do
7:              if Different_Patterns(Ls[j], Ls[k]) then
8:                  Temp_Patterns ← Join(Ls[j], Ls[k])
9:                  Temp_Gains ← Update_Gains(Ls[j], Ls[k])
10:                 if Feasible(S) then
11:                     Patterns ← Temp_Patterns
12:                     Gains ← Temp_Gains
13:                 endif
14:             endif
15:         endfor
16:     endfor
17: endfor
```

been modified, whose number of active beams has been changed, need to recalculate their gains, as it has been explained in section 2.2. The last step consists in checking whether the link schedule is still feasible with the modified radiation patterns. It is only necessary to check the feasibility of the set of links scheduled in those timeslots that contain a link whose patterns has been changed. If the schedule is feasible, the new radiation patterns and gains of the network are preserved. Thereupon, the algorithm tries to join the associated patterns of another pair of links from list Ls. When all the possible combinations have been explored, the same procedure is

repeated for a different antenna.

5.3.2 Beam Joining algorithm with Interference Number

The idea behind this algorithm is to find the beam that is causing less interference to its neighbors in the network and try to join this beam (taking into account that this beam might be already joined with other beams) with the next beam from the same antenna (which might be also joined with others) that causes less interference. In order to do that a metric that measures the interference that a transmitting link causes needs to be used. The interference number (IN), defined in [18] and previously explained in section 3.2.1, serves as a good approximation to find the pair of patterns that are causing less interference to the network. Thus, The interference number of a link $(i, j)\epsilon E$ is the number of links $(m, n)\epsilon E \setminus (i, j)$ that cannot establish a communication at the same time such the links (m, n) and (i, j) do not share an endpoint and is infeasible. A set of two links is considered infeasible when the receiver nodes do not satisfy the SINR constraint described in equation 2.3.2 for directional antennas. In this algorithm, we calculate the IN with the initial radiation patterns (where all patterns are formed by single beams). Consequently, using this IN to select the pair of patterns to be joined does not imply that this pair is the one that causes the less interference since finding the best pair of patterns would require the calculation of the new IN each time the radiation patterns change. We defer from doing so since that would substantially increase the computational complexity.

The pseudo-code of the proposed scheme is shown in Algorithm 2. The Beam Joining algorithm based on the interference number, initiate by sorting the links in increasing order of their IN. Thereafter, the algorithm selects link $e_1 = (u, v)$ from list L and decide equiprobably from which antenna u or v to start joining beam patterns. Then, it is created a sorted list by IN (Ls) with the incoming and the outgoing links of the antenna selected without including link e_1. Based on that, the first candidate to perform the beam joining from list Ls is selected. Note that the antenna selected has associated a radiation pattern for each incoming and outgoing link, where each pattern can be compound for a different number of beams. Once both links are selected, it is checked if these patterns are suitable to be joined.

Patterns are suitable to be joined whether they are different or even being equal the links that use these patterns are different. The first situation is very reasonable since the goal of the algorithm is to join different patterns in order to reduce the number of switches. The second case could be more confusing as joining equal

Algorithm 11: Beam Joining algorithm with Interference Number

Input : A graph $G = (V, E)$, $E(v)$ denotes the set of all edges in E at a
: vertex v.
: r, uniformly distributed [0, 1] random variable.
: $p \leftarrow 0.5$
: L, List containing all links sorted in increasing order by its IN.
: S, A feasible schedule with length TS.
: $Gains$, gains of all patterns.

Output : $Patterns$, List of all patterns and their respective associated
: patterns that are activated at the same time.
: S, A feasible schedule with length TS.

1: **for** $i \leftarrow 1$ **to** $|L|$ **do**
2: $[u, v] \leftarrow L(i)$, where (u,v) are the nodes of link i
3: **if** $r > p$ **then**
4: $A[1] \leftarrow u$; $A[2] \leftarrow v$
5: **else**
6: $A[1] \leftarrow v$; $A[2] \leftarrow u$
7: **endif**
8: **for** $k \leftarrow 1$ **to** 2 **do**
9: $Ls \leftarrow E(A[k]) \setminus L(i)$
10: $Ls \leftarrow OrderbyIN(Ls)$
11: $jointSuccess \leftarrow 0$
12: $j \leftarrow 0$
13: **while** $j < |Ls|$ *and* $jointSuccess = 0$ **do**
14: $j \leftarrow j + 1$
15: **if** $Candidates(L[i], Ls[j])$ **then**
16: $Temp\_Patterns \leftarrow Join(L[i], Ls[j])$
17: $Temp\_Gains \leftarrow Update\_gains(L[i], Ls[j])$
18: **if** $Feasible(S)$ **then**
19: $jointSuccess \leftarrow 1$
20: $Patterns \leftarrow Temp\_Patterns$
21: $Gains \leftarrow Temp\_Gains$
22: **endif**
23: **endif**
24: **endw**
25: **endfor**
26: **endfor**

patterns does not lead to a reduction of the switches. However, patterns are not considered merged initially, in this algorithm, in order to give more flexibility to the joining procedure. For instance, suppose that patterns of links e_1 and e_2 are initially considered merged. Following the order established in the IN list we want to join the patterns associated to links e_3 and e_2. However, due to requirements of the system, the pattern corresponding to link e_3 can not be joined with the original pattern of link e_1 (without merging it with pattern of link e_2), but it can be joined with the original pattern of link e_2 (without considering it merged with e_1). Hence, patterns of link e_3 and e_2 can not be joined because of having merged initially patterns of links e_1 and e_2. Figure 5.4 shows how one switch would have been saved if links e_1 and e_2 had not been merged initially and patterns would have been joined following the interference number.

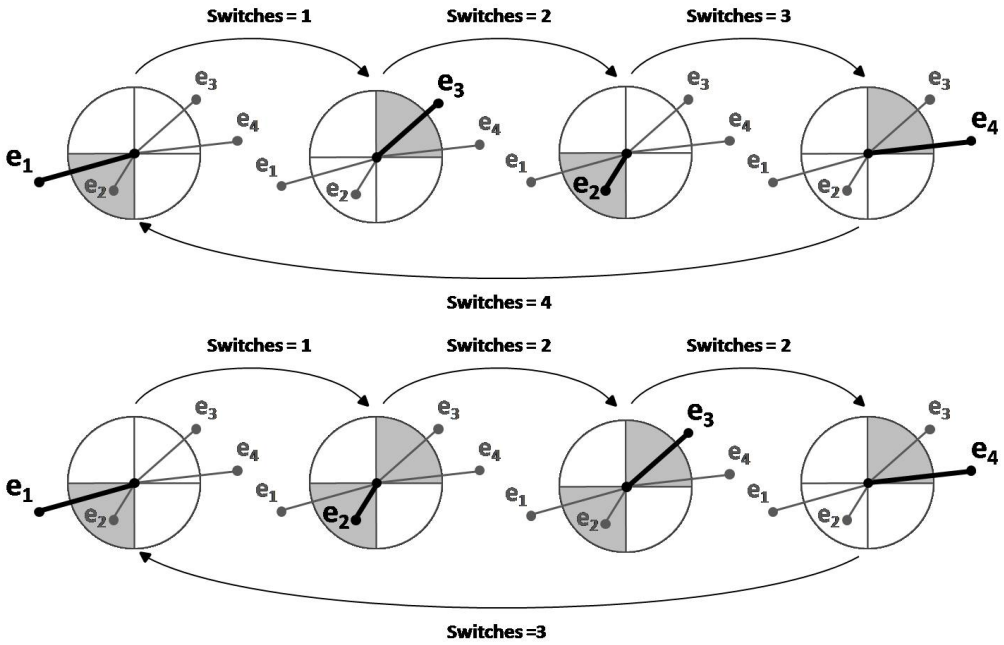

Figure 5.4: Above: Sequence of radiation patterns with single beams. Below: Sequence of radiation patterns with single and joined beams. Note that the links in bold represents the links that are active each time.

As previously explained in 5.3.1, when two patterns are suitable candidates the Pattern List and the antenna gains are updated. After that, it is checked if these patterns are suitable to be joined in terms of $SINR$ threshold feasibility; if the schedule is feasible we keep the new radiation patterns and gains of the antenna, whereas if it is not feasible, we select another candidate from list L_s. Thereupon, whether we are in the first antenna and we managed to join the pattern of link

e_1 with another pattern or whether the pattern of link e_1 can not be joined with any pattern from list L_s, we change of antenna and apply the same procedure as above. In case we have checked both antennas, we select another link from list L to perform the joining. The same process is followed until all links from list L have been selected.

Once the algorithm is terminated, it is checked for each antenna if the number of beam switchings are lower with the initial patterns than with the current beam forming patterns. Figure 5.5 proves that joining patterns not always leads to a reduction of the number of switches. For instance, let us consider that links are activated following the order e_1-e_2-e_3-e_4 for both link sequences. In the upper case, when the antenna uses single beams, two switches are only required since the radiation pattern between links e_1 and e_2 and between links e_3 and e_4 are equal. However, in the lower sequence, the number of switches not only have not been reduced, but they have been increased from two to four, despite two beams have been joined (when link e_1 or link e_3 is activated, the antenna uses the beams originally assigned to the link e_1 and e_3). Note that this problem can only happen when equal patterns are not considered initially merged; otherwise, after joining patterns we keep at least the same number of switches.

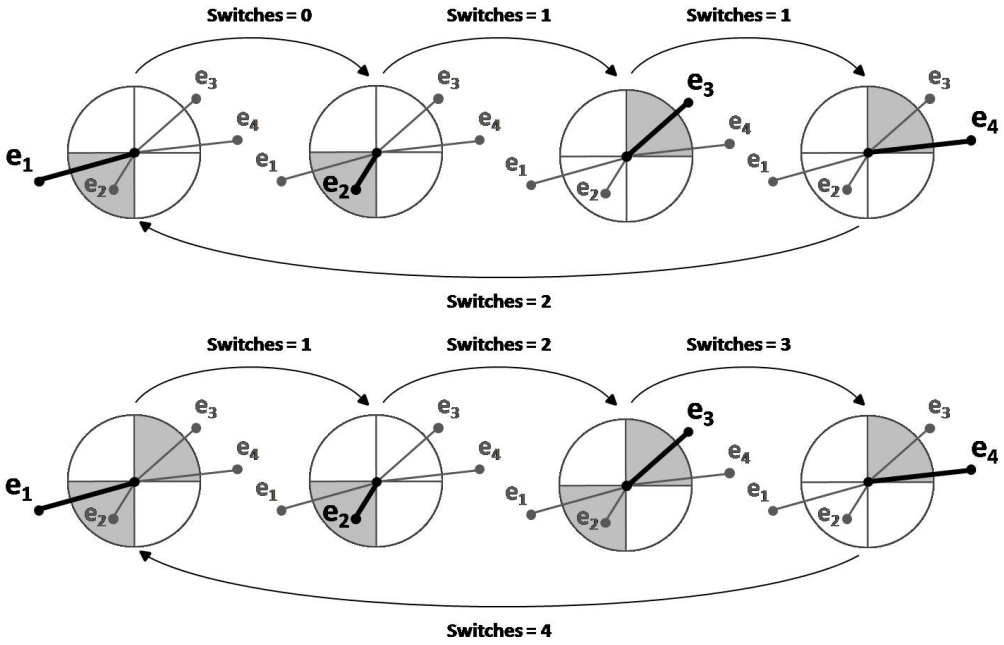

Figure 5.5: Above: Sequence of radiation patterns with single beams. Below: Sequence of radiation patterns with single and joined beams. Note that the links in bold represents the links that are active each time.

In each antenna that this situation occurs, it is tried to be corrected restoring the initial patterns of that antenna. However, not always is possible to restore the initial patterns of an antenna, because that modification might not accomplish the feasibility of the timeslot allocation. The pseudocode to perform the switch correction is reported in algorithm 12.

Algorithm 12: Beam Switch Restoration

Input : V, list of all nodes.
: S, Schedule of the network.
: $Patterns_{initial}$, List of all initial patterns.
: $Patterns$, List of all patterns and their respective associated
: patterns that are activated at the same time after the joining.
: $Gains$, gains of all patterns.
Output : $Patterns$, Updated patterns.
: $Gains$, Updated gains of all patterns.

1: $Sw_{initial} \leftarrow Count\_switches(Patterns_{initial}, S)$
2: $Sw \leftarrow Count\_switches(Patterns, S)$
3: **for** $i \leftarrow 1$ **to** $|V|$ **do**
4: **if** $Sw[i] > Sw_{initial}[i]$ **then**
5: $Temp\_Patterns \leftarrow Restore\_beams(i)$, disjoint beams of antenna i
6: $Temp\_Gains \leftarrow Restore\_gains(i)$, set up initial gains of antenna i
7: **if** $Feasible(S)$ **then**
8: $Patterns \leftarrow Temp\_Patterns$
9: $Gains \leftarrow Temp\_Gains$
10: $Sw[i] \leftarrow Sw_{initial}[i]$
11: **endif**
12: **endif**
13: **endfor**

5.3.3 Beam Joining by allowing STDMA re-scheduling

Up to this point, a fixed link schedule was assumed and joining antenna beams was accepted if the link schedule remained feasible. Joining beams causes a change of the antenna patterns, and so in its gains, which implies a redistribution of the interference that nodes cause to each other. Consequently, re-allocating the links each time a beam joint is performed, might take profit of this redistribution allowing more simultaneous beams to be activated at the same time. Hence, each time a

pattern is modified, instead of checking the feasibility of the set of links scheduled in those timeslots that contain a link whose patterns have been changed, we may re-schedule all links in order to redistribute more efficiently the interference. Thereby, a beam joint is successful if the new schedule has the same or less number of timeslots as the initial schedule since it is not the purpose of this work the reduction of switches at the expense of a deterioration in the timeslot allocation, in other words, increasing the number of timeslots. Algorithm 13 shows how to modify algorithms 10, 11 and 12 to apply this strategy.

Algorithm 13: Re-scheduling in Beam Joining Algorithms

1: % Modification in Beam Joining algorithms; replace line 10 in algorithm 10 and line 18 in algorithm 11 for
2: $TS_{new}, S_{new} \leftarrow Calculate\_Schedule()$, with the updated Patterns
3: **if** $TS_{new} \leq TS$ **then**
4: $\quad S \leftarrow S_{new}$

Undoubtedly, in terms of number of switches, re-scheduling might permit a higher reduction of switches, however this approach is less computationally efficient since requires a complete re-allocation of all links each time beams are jointed. The re-allocation might be complete or partial depending on the scheduling algorithm employed. For instance, if we use a packing algorithm, as the Greedy Physical employed here, we do not need to re-schedule the set of links, from the first link until the previous link of the first link that its radiation patterns has been modified in the Packing List of the algorithm (which contains all links sorted by the order in which they have been packed), since they will remain in the same timeslots as the original schedule calculated. In fact, this is the same idea that we apply to the RSP algorithm in section 4.2.3. Observe that a similar expression of the average number of links that do not need to be re-schedule can be easily obtained for this case.

5.4 Numerical Investigations

In this section we present the computational results obtained based on the set of algorithms proposed in section 5.3. The results presented are the averaged values over 200 randomly generated wireless mesh network topologies within the predefined area. The complete set of the simulation parameters used in the numerical investigations are summarized in Table 5.2 below.

Figure 5.6 illustrates the improvement achieved in terms of spatial reuse of timeslots employing switched beam forming antennas compared to omnidirectional an-

Table 5.2: Simulation Parameters.

Notation	Explanation	Values
A	Length of the Square Area	850 meters
N	Number of Nodes	20 - 60
L	Number of Links	19 - 59
d_0	Close-in reference distance	50 meters
SNR	SNR required	15 dB
γ	SINR threshold	8 dB
η	Path loss exponent	3.5
P_{max}	Maximum transmitted power	20 Watt
f_c	Carrier frequency	3.8 GHz
W	Thermal & background Noise	-132 dBW

tennas. Despite the fact that demonstrating the performance improvement of directional antennas in WMNs is not the main goal of this work, it is important to state this fact in order to justify the rational of deploying directional antennas. As expected, the spatial reuse decreases as the beam width of the antenna is increasing. This expected behavior occurs since directional antennas focus most of their transmitted power in an area controlled by the beamwidth of the main lobe; as the beamwidth increases larger areas are interfered and, as a consequence, this affects the spatial reuse of timeslots. Finally, observe that the improvement is becoming more significant as the number of nodes of the network increases.

A similar behavior for the spatial reuse with directional antennas can be observed for the reduction of the number of beam switchings. The reduction of switches increases as the directionality of the antenna beam increases, as can be noted from figure 5.7. Observe from the figure that as the number of nodes increases the percentage of switches reduced after applying algorithm 11 decreases. However, this does not mean that the absolute value of beam switchings saved has diminished, as it is shown in table 3. From the table it can be noted that as the number of nodes increases the initial number of switches augments. The same behavior holds for algorithm 1, except for a slightly reduction in the percentage of improvement in terms of how much the number of beam switches have been reduced. Hence, in these scenarios algorithm 11 achieves better performance than algorithm 10.

In figure 5.7 we also evaluate the reduction in the number of beam switches in the case where a fixed scheduling is considered or when link rescheduling is allowed.

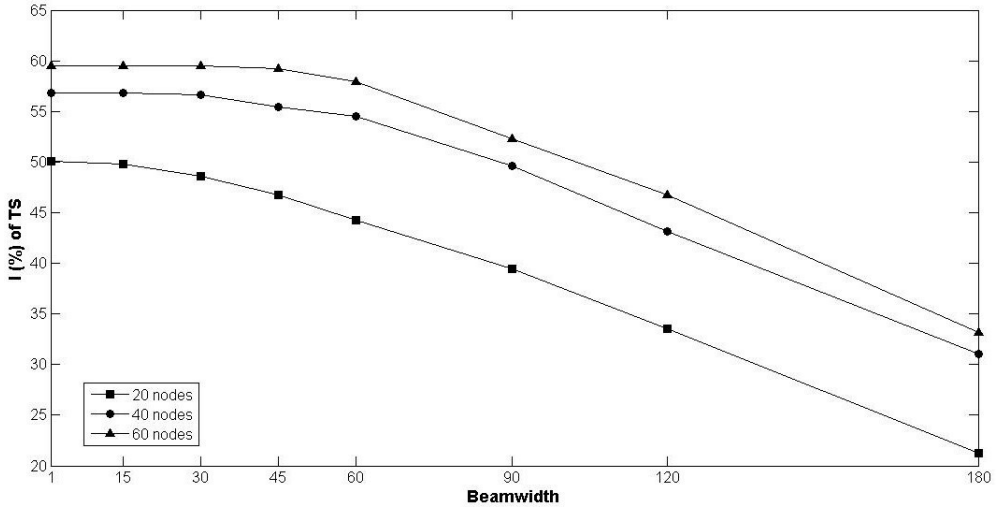

Figure 5.6: Reduction of timeslots employing directional antennas compared to employing omnidirectional antennas.

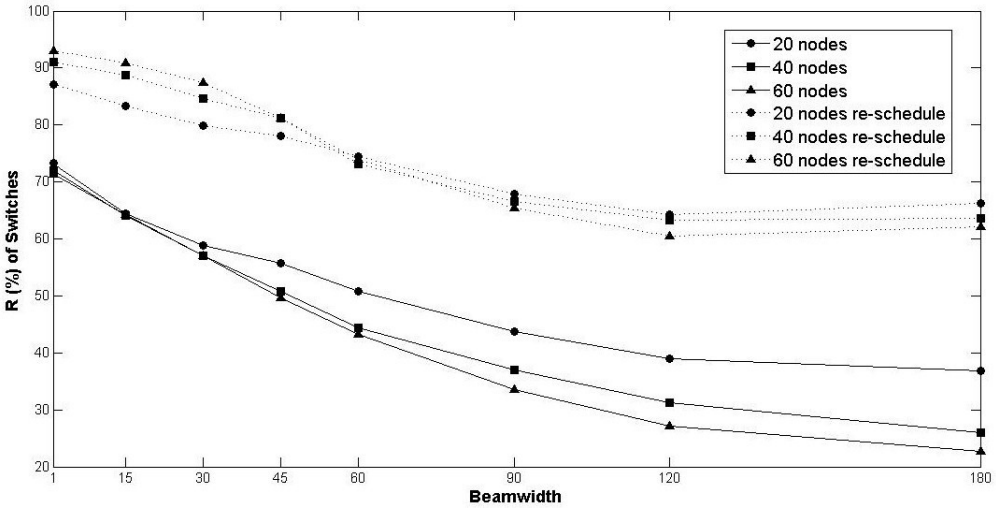

Figure 5.7: Number of switches reduced (in percentage) using algorithm 11 compared to the initial switches with directional antennas depending on their beamwidth.

When rescheduling of the links is permitted the scheduling algorithm is able to gain from the redistribution of the interference, entailing in an increased reduction of the number of beam switchings. Therefore, by rescheduling the links in the WMN can considerably decrease the number of switches, although the computational complexity increases compared to the case where there is a fixed schedule.

Figures 5.8 and 5.9 illustrate the performance of directional beam switched an-

Table 5.3: Comparison between beam joining algorithms with Interference Number (IN) and with Pattern Order (PO).

Nodes	θ_m	**Initial Switches**	**Final Switches IN**	**Final Switches PO**	**Reduction IN**	**Reduction PO**
20	30	31.7	13.1	14.1	58.8	55.6
20	60	31.6	15.6	16.6	50.7	47.5
40	30	66.1	28.4	30.1	57.0	53.6
40	60	65.7	36.6	39.1	44.4	40.6
60	30	99.9	42.9	46.3	57.1	53.6
60	60	99.6	56.5	60.2	43.2	39.5

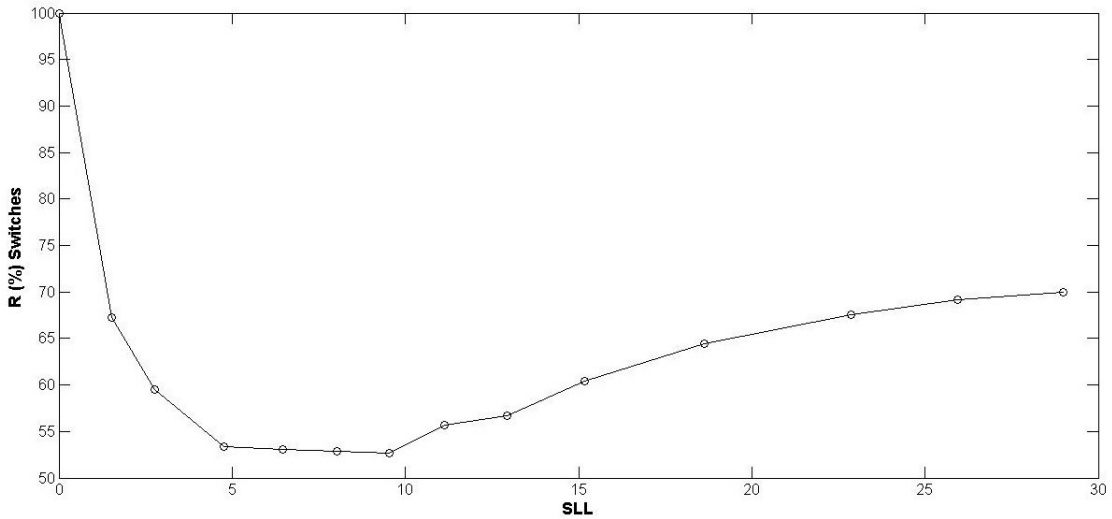

Figure 5.8: Number of switches reduced (in percentage) using algorithm 11 compared to the initial switches with directional antennas depending on the SLL. Note that despite 20 nodes are considered in this example, the same behavior holds for bigger networks.

tennas when varying the Side Lobe Level (SLL). Increasing the SLL results in a more directional antenna and as a consequence link scheduling can be performed with less required timeslots. Observe from figure 5.8 the convex structure of the curve regarding the percentage of reduction in beam switchings as a function of SLL. This result reveals that there is a worst SLL value for which the total aggregate number of beam switchings in the network is maximized. Figure 5.9 shows

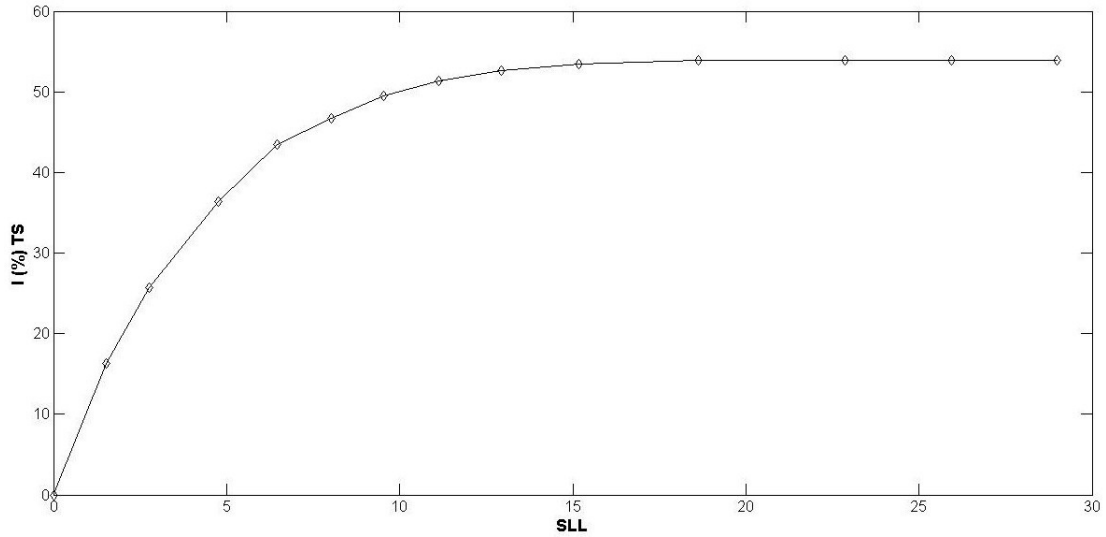

Figure 5.9: Reduction of timeslots employing directional antennas compared to employing omnidirectional antennas depending on the SLL. The same networks used in the previous example are considered herein.

in percentage the reduction of the frame length (in terms of timeslots) for different values of SLL. As can be seen from the figure, initially the timeslots decrease fast, since overall interference is decreasing and therefore links can be packed more efficiently in each timeslot. Note that each additional increase in the value of SLL yields smaller improvements in the number of timeslots and also that beyond some point the reduction of timeslots reaches a plateau where no more improvement can be achieved. However, the initial reduction of timeslots produces an increase on the number of beam switches, since the number of links per timeslot has also increased and the beam joining becomes more difficult. Observe that when no considerable improvement is achieved so far in terms of timeslots, the beam joining algorithm takes advantage of the overall reduction of interference diminishing the number of switches.

Despite the fact that figure 5.9 has been obtained when considering a fixed schedule, the same behavior holds also when rescheduling is allowed. The reason is that even if rescheduling is allowed the scheduling algorithm will still reach a limit on how many links can allocate in each timeslot due to the SINR constraint.

Chapter 6

Conclusions

In this thesis we have studied different heuristic STDMA link scheduling algorithms using both interference-based network model and graph-based network model that had been previously proposed in the literature. We have shown how the Greedy Physical algorithm achieves better timeslots allocations than the Packing Heuristic and the Recursive Largest First algorithms.

Furthermore, we have developed different enhance versions of the scheduling algorithms presented herein ending up with the design of a fast randomized link scheduling algorithm for Spatial-TDMA enabled wireless mesh networks, named Randomized Link Swap Packing algorithm. The randomization is based on swapping links on a list that is created by well known greedy scheduling algorithms such as the Greedy Physical and the Packing Heuristic. Extensive numerical investigations reveal that the proposed fast scheduling scheme can improve by up to 11% the timeslot reuse compared to the previous mentioned link scheduling algorithms. Another important characteristic of the proposed scheme is that its structure is amenable for parallel processing and therefore, emerging multi-core and multi-CPU enabled network elements can be fully utilized. The simplicity of the algorithm, the achieved gains and the potential of parallel computation clearly demonstrate the potential benefits of the proposed scheme. Future avenues of research include a theoretical characterization of the proposed randomized scheduling scheme. In addition, would be worthwhile investigating the potential of integrating the proposed scheme with routing so that a joint randomized scheduling and routing scheme can be implemented.

In addition to the STDMA link scheduling problem, we also considered herein the application of directional antennas in a WMN. Compared to the omnidirectional case, directional antennas significantly improve the spatial reuse of resources

in WMNs, due to the increased directivity of the antenna beam which allows the nodes to avoid inferring signals arriving at the receiver from other concurrent transmissions. A switch beam antenna system generates a predefined number of beams and can select and switch between the beams depending on which link is activated, i.e., which link is scheduled for transmission. In this thesis we have shown that link scheduling and beam switching are closely intertwined and in fact the number of beam switchings can unnecessarily increase depending on the order in which the links are scheduled for transmission. We have illustrated how by jointly considering beam switching and link scheduling the number of beam switchings can be dramatically decreased. To this end, a set of algorithms are proposed to jointly optimize scheduling and antenna beam switching. A wide set of numerical investigations reveal that the number of beam switchings can be reduced by almost 90% without affecting the frame length (the spatial reuse of timeslots in the network). Eventually, note that to increase the speed of the algorithms presented here, instead of merging beams one by one, several joints can be performed at the same time. Therefore, the number of times that a schedule needs to be check if it is feasible or the number of times that links have to be re-schedule can be reduced.

Bibliography

[1] I. F. Akyildiz, X. Wang, and W. Wang., "'wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.

[2] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications*, pp. 123–131, 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[4] "Wimax forum [online]." Available: http://www.wimaxforum.org/.

[5] *IEEE Standard for Local and metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE Std. 802.16*, 2004.

[6] R. Nelson and L. Kleinrock, "Spatial-tdma: A collision-free multihop channel access protocol," *IEEE Transactions on Communications*, vol. 33, pp. 934–944, September 1985.

[7] V. Friderikos and K. Papadaki, "Interference aware routing for minimum frame length schedules in wirelessmesh networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2008, 2008.

[8] J. Grönkvist, "Traffic controlled spatial tdma in multihop radio networks," *IEEE PIMRC 98*, October 1998.

[9] O. Somarriba, "Multihop packet radio systems in rough terrain," radio communication systems,department of s3, Royal Institute of Technology, Stockholm, Sweden, October 1995.

[10] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 456–460, 1990.

[11] S. Krumke, M. Marathe, and S. Ravi, "Models and approximation algorithms for channel assignment in radio networks," *ACM Wireless Networks*, vol. 7, pp. 575–584, 2001.

[12] J. Grönkvist and A. Hansson, "Comparison between graph-based and interference-based stdma scheduling," *IEEE MobiHoc*, pp. 255–258, 2001.

[13] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *ACM Mobicom*, pp. 66–80, 2003.

[14] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, pp. 910–917, 1988.

[15] C. G. Prohazka, "Decoupling link scheduling constraints in multihop packet radio networks," *IEEE Transactions on Computers*, March 1989.

[16] A. M. Chou and V. O. Li, "Slot allocation strategies for tdma protocols in multihop packet radio networks," in *IEEE INFOCOM 092*, 1992.

[17] A. Behzad and I. Rubin, "On the performance of graph-based scheduling algorithms for packet radio networks," in *IEEE GLOBECOM*, (San Francisco, CA), December 2003.

[18] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 2–13, ACM, 2006.

[19] A. Das, R. Marks, P. Arabshahi, and A. Gray, "Power controlled minimum frame lenght sheduling in tdma wireless networks with sectored antennas," in *IEEE INFOCOM'05*, (Miami), March 2005.

[20] I. Chlamtac and S. Kutten, "Sa spatial reuse tdma/fdma for mobile multi-hop radio networks," in *IEEE INFOCOM 85*, vol. 1, pp. 389–394, 1985.

[21] B. Hajek and G. Sasaki, "Link scheduling in polynomial time,," *IEEE Trans. Inform. Theory*, vol. 34, pp. 910–917, September 1988.

[22] R. Ramanathan, "On the performance of ad hoc networks with beamforming antennas," *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pp. 95–105, October 2001.

[23] R. Ramanathan, *Antenna Beamforming and Power Control for Ad hoc Networks*, ch. Book Chapter on Mobile Ad Hoc Networking. IEEE Press/Wiley, 2004.

[24] K. Sundaresan, R. Sivakumar, M. Ingram, and T. Chang, "A fair medium access control protocol for ad hoc networks with mimo links," *Conference on Computer Communications (INFOCOM)*, pp. 2559–2570, 2004.

[25] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for ieee 802.11 wireless networks," *International Conferences on Broadband Networks (BroadNets)*, 2004.

[26] J. So and N. Vaidya, "Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pp. 222–23, May 2004.

[27] M. McHenry, "Frequency agile spectrum access technologies." FCC Workshop on Cognitive Radios, May 2003.

[28] B. Lane, "Cognitive radio technologies in the commercial arena." FCC Workshop on Cognitive Radios, May 2003.

[29] J. Mitola III, "Software radio architecture: Object-oriented approaches to wireless system engineering," *Wiley Inter-Science*, 2000.

[30] E. Karapistoli, I. Gragopoulos, I. Tsetsinas, and F. Pavlidou, "A mac protocol for low-rate uwb wireless sensor networks using directional antennas," *Computer Networks*, no. 53, pp. 961–972, 2009.

[31] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S.Das, "Mobisteer: using steerable beam directional antenna for vehicular network access," in *Proc. of IEEE MobiSys 2007*, pp. 192–205, June 2007.

[32] A. Subramanian1, V.Navda, P. Deshpande, and S. Das, *A Measurement Study of Inter-Vehicular Communication Using Steerable Beam Directional Antenna*. Computer Science Department, Stony Brook University, Stony Brook, NY 11794, U.S.A.

[33] M. Blanco, R. Kokku, K. Ramachandran, S. Rangarajan, and K. Sundaresan, "On the eectiveness of switched beam antennas in indoor environments," *Passive and Active Network Measurement*, vol. 2008, 122–131.

[34] G. Li, L. L. Yang, W. Conner, and B. Sadeghi, "Opportunities and challenges for mesh networks using directional antennas," in *IEEE WiMESH'05*, September 2005.

[35] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaida, "Using directional antennas for medium access control for ad hoc networks," *Mobicom*, 2002.

[36] M. Sanchez and J. Zander, "Adaptive antennas in spatial tdma multihop packet radio networks," in *RVK 99*, June 1999.

[37] S. Guo, O. Yang, and V. Leung, "Joint optimization of energy consumption and antenna orientation for multicasting in static ad hoc wireless networks," *IEEE Transactions on Wireless Communications*, vol. 5, September 2006.

[38] T. S. Rappaport, "Wireless communications principles and practice," in *Upper Saddle River*, (NJ, USA:Prentice Hall), 2002.

[39] V. Friderikos, K. Papadaki, D. Wisely, and H. Aghvami, "Multi-rate power-controlled link scheduling for mesh broadband wireless access networks," *IET communications*, vol. 1, no. 5, pp. 909–914, 2007.

[40] R. Diestel, *Graph Theory*. Springer-Verlag, second ed., 2000.

[41] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Info. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[42] K. Papadaki and V. Friderikos, "Approximate dynamic programming for link scheduling in wireless mesh networks," *Computers & Operations Research*, vol. 35, no. 12, pp. 3848–3859, 2008.

[43] J. Grönkvist, "Traffic controlled spatial reuse tdma in multi-hop radio networks," in *IEEE PIMRC 98*, vol. 3, pp. 1203–1207, September 1999.

[44] F. T. Leighton, "A graph coloring algorithm for large scheduling problems," in *Journal of Research of the National Bureau Standard*, vol. 84, pp. 79–100, September 1979.

[45] W. Lin, C. Zhang, M. Li, and M.-Y. Wu, "An interference graph based mac protocol for ad hoc wireless networks," in *CIT '06: Proceedings of the Sixth IEEE International Conference on Computer and Information Technology*, (Washington, DC, USA), p. 94, IEEE Computer Society, 2006.

[46] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculation by fast computidg machines," *Journ. of Chem. Phys.*, 1953.

[47] S. Kirkpatrick, C. Gellat, and M. P. Vecchi, "Optimizatis by simulated annealing," *Science*, 1983.

[48] I. Rhee, A. Warrier, J. Min, and L. Xu, "Drand: Distributed randomized tdma scheduling for wireless adhoc networks," *ACM MobiHOC*, 2006.

[49] I. Foster, *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering.* Addison Wesley, 1995.

[50] V. Ramamurthi, A. Reaz, S. Dixit, and B. Mukherjee, "Link scheduling and power control in wireless mesh networks with directional antennas," *IEEE International Conference on Communications (ICC)*, 2008.

[51] I. Martinez and J. Altuna, "Influence of directional antennas in stdma ad hoc network schedule creation," in *International Workshop on Wireless Ad-hoc Networks*, (London, UK), 2005.