ROUTING AND SCHEDULING USING COLUMN GENERATION IN IEEE 802.16J WIRELESS RELAY NETWORKS

Tomás M. Murillo

A thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Computer Science

Concordia University

Montréal, Québec, Canada

August 2011

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Tomás M. Murillo

Entitled: Routing and Scheduling Using Column Generation in IEEE 802.16j Wireless Relay Networks

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee :

Dr. Dhrubajyoti Goswami __________ Chair

Dr. Thomas G. Fevens __________ Examiner

Dr. Jaroslav Opatrny __________ Examiner

Dr. Brigitte Jaumard __________ Supervisor

Approved by ____________________

Chair of Department or Graduate Program Director

Dr. Robin A. L. Drew __________

Dean of the Faculty of Engineering and Computer Science

Date AUG 09 2011 __________

Abstract

Routing and Scheduling Using Column Generation in IEEE 802.16j Wireless Relay Networks

Tomás M. Murillo

Worldwide Interoperability for Microwave Access (WiMAX) has become an important standard in wireless telecommunication networks in recent years due to the increasing bandwidth requirements, as well as to customer demand for having ubiquitous access to the network. One of the most recent versions of WiMAX is IEEE 802.16-2009, but in this thesis we work with its 802.16j amendment. This amendment includes the use of relay stations (RS) to improve the network's throughput, with the RSs becoming intermediaries between the base station (BS) and the subscriber stations (SS).

In the literature, there have been several authors claiming to perform joint routing and scheduling in wireless networks using the column generation technique. Nevertheless, these papers are not performing scheduling since they do not specify how time slots are allocated to each transmitting node over time (they only count the time slots it takes to transmit data).

That is why we developed an optimization model (that is solved using column generation) having in mind the fact of performing real scheduling, not only counting time slots but taking into account the allocation of resources over a period of time. The model we developed chooses among a set of possible configurations (a set of transmitting links over a predetermined period of time slots) to calculate the time it takes to transmit data from end to end.

After obtaining some simulation results with our model, we compared them with those of a model that does not perform real scheduling. The results show only minor differences in the total number of time slots that a transmission lasts since we can only assign a small number of time slots per configuration.

Acknowledgments

Writing and presenting this thesis is the culmination of my master degree, a dream that I pursued most of my life. And at this point in time and in this section of the thesis I am going to thank all the people who helped me to get here.

First of all, I want to thank my thesis supervisor, Dr. Brigitte Jaumard, for giving me the idea for my thesis topic, guiding me through the thesis creation process, and providing me economic support to pay for my master degree.

Next, I would like to mention my partners and friends from the lab, the students who are also under Dr. Jaumard's supervision, and especially thank Dr. Samir Sebbah and Mr. Hoang Hai Anh for explaining me theoretical concepts and helping me solve programming issues, as well as Mr. Oscar Delgado for the discussions about wireless networks.

I owe a big debt to my parents Mr. Guillermo Murillo and Mrs. Brigitte Bochert, for their unfailing encouragement and support of my education. In addition, I would like to thank Dr. Brigitte Schroeder-Gudehus, who opened the doors of her house to me when I arrived in Montreal and supported me economically throughout the duration of my studies at Concordia University. Finally, I want to thank my girlfriend Ms. Camila Oliveira for her heartening support throughout these last years of study.

I assure all these people of my gratitude.

Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
BS	Base Station
BWA	Broadband Wireless Access
CID	Connection Identifier
CPS	Common Part Sublayer
CS	Convergence Sublayer
DL	Downlink
DSL	Digital Subscriber Line
FDD	Frequency Division Duplexing
ILP	Integer Linear Program
LAN	Local Area Network
LOS	Line of Sight
LP	Linear Program
LTE	Long Term Evolution
MAC	Media Access Control
MCN	Multihop Cellular Networks
MIMO	Multiple Input and Multiple Output
MP	Master Problem
NIC	Network Interface Card
NLOS	Non-Line of Sight

OFDMA	Orthogonal Frequency-Division Multiple Access
OPL	Optimization Programming Language
OSI	Open Systems Interconnection
PDA	Personal Digital Assistant
PHY	Physical (Layer)
PHY SAP	Physical Layer Service Access Point
PMP	Point-to-Multipoint
PP	Pricing Problem
PTP	Point-to-Point
QoS	Quality of Service
RAM	Random Access Memory
RMP	Restricted Master Problem
RS	Relay Station
SAP	Service Access Point
SC	Single Channel
SINR	Signal to Interference-plus-Noise Ratio
SS	Subscriber Station
STDMA	Spatial Time Division Multiple Access
TDD	Time Division Duplexing
TDMA	Time Division Multiple Access
UL	Uplink
WiMAX	Worldwide Interoperability for Microwave Access

Chapter 1

Introduction

1.1 Project Background

Nowadays, the wireless telecommunication industry is slowly entering a new generation in which high bandwidth and ubiquitous access to wireless networks are increasingly gaining importance. In this scenario, we see two technologies competing for market share to prevail as the most used standard. These two technologies are Worldwide Interoperability for Microwave Access (WiMAX) and Long Term Evolution (LTE).

While both standards (WiMAX and LTE) have a similar way to operate, their main difference resides in their cost of deployment. WiMAX is cheaper to deploy in locations where there are minimal or no wired networks (such as the countryside or some developing countries) [ACH10], while LTE is a better investment for locations that have already a wired network (big cities or developed countries).

In our research, we chose to work with WiMAX as opposed to LTE. The reason we selected this technology is because, as we mentioned earlier, it is cheaper to implement than LTE in the developing world, where there are millions of potential customers that still have to be reached by high bandwidth services.

More specifically, the standard that we utilize is IEEE 802.16j, which is an amendment of the IEEE 802.16-2009 standard. The 802.16j version utilizes relay stations (RS), which are intermediary nodes in the wireless network, located between the base station (BS) and subscriber stations (SS). The main reason for the use of RSs is to provide more speed in the network, since the RSs act as helpers of the BS when transmitting and receiving data to and from the SSs.

1.2 Problem Statement

Several authors have published papers that claim to be performing routing and scheduling on wireless networks utilizing column generation to find an optimal solution for their models. However, the way these authors developed their model does not fully comply with the definition for scheduling that we cite in Section 3.1. According to the scheduling definition, we should be able to see how information flows through each node in the network across time, knowing which resources will be available to each node at each time slot.

The models that we just mentioned show us how the information flows in the network by telling us which configurations are created and used (with each configuration being the set of active or transmitting links during a time slot). Nevertheless, the results obtained by these models do not state in which order in time the configurations should be utilized. In addition, they do not take into account that a node has to have data in it before being able to transmit anything. Hence, the reason why these models do not perform real scheduling is that they do not consider when each configuration should be utilized and therefore do not show which resources are available to each node during each time slot (since there is no order in time in which those resources are assigned).

Considering the fact that the models we just mentioned did not do proper scheduling, we decided to elaborate a routing and scheduling optimization model for WiMAX IEEE 802.16j (wireless relay network). To solve our model (Model II) we also utilize the column generation tool. What makes our model different from the others is that it actually performs scheduling within each configuration that we generate.

If we analyze any configuration generated by our model, we will be able to see how data is routed in the network throughout time. We utilize buffers to help the simulation keep track of how much data is contained by each node at each time slot. The disadvantage of our model is that it does not state in which order in time the configurations should be utilized. Even so, to compensate for the lack of scheduling in the selection of configurations, we designed the model so that data is sent from end to end within the time slots that form a configuration. That way, it does not matter in what order the configurations could be used, because in each configuration we send data from end to end and we will never have the situation of a node transmitting data when it does not have anything to send yet (as it happens in the previously mentioned models).

With this introduction to the problem, we can now say that one of the objectives of this research is to show how the model that we developed (Model II) approaches as much as possible the definition of scheduling. We also demonstrate how our model is able to route and schedule traffic based on the conditions of the network (such as traffic and node distribution). In addition, the other objective of our investigation is to compare the results obtained by simulating our model, with the results obtained by simulating a model (Model I) based on one of the previously mentioned models that do not perform scheduling as the definition (of scheduling) states.

1.3 Scope of Thesis Work and Contribution

As we said in the previous section, our model (Model II) performs scheduling on a configuration by configuration basis, but does not do so when it comes to selecting when each configuration should be utilized. Notwithstanding, our model approaches more the definition of scheduling than other previously proposed models (such as Model I) for routing and scheduling in wireless networks utilizing the column generation tool.

Therefore, our contribution is to provide a planning tool that simulates routing and scheduling in a WiMAX 802.16j network utilizing the column generation technique and approaching the definition of scheduling as much as possible. In addition, we compare the results obtained by our model (Model II) with those obtained by a model (Model I) that claims to be performing scheduling but does not do so in the pure definition of the term.

1.4 Thesis Outline

We have organized the thesis in 7 different chapters, starting with the current chapter which is the introduction to the thesis. In Chapter 2, we give background information about the WiMAX technology. In Chapter 3, we cite the definition of the term ”scheduling” and make a literature review mentioning papers that perform scheduling with heuristics or using the column generation method. Chapter 4 contains the definitions of the two mathematical models we work with, while Chapter 5 explains how we use the column generation tool to solve both of these models. Finally, Chapter 6 contains the results obtained from the experiments done after simulating Models I and II, and Chapter 7 provides a conclusion based on these experiments.

Chapter 2

WiMAX Technical Background

In the first place we have to explain what WiMAX is. WiMAX stands for the term Worldwide Interoperability for Microwave Access and it is a telecommunications technology based on the IEEE 802.16 standard that provides broadband wireless access (BWA) [WIS07] [SIJT09]. Its characteristic is that it can provide these services over long distances and in fixed and mobile modes (we explain these later on).

As we mentioned, WiMAX is ruled by the IEEE 802.16 standard, which is also known as Air Interface for Fixed Broadband Wireless Access Systems [StIMTS09a]. This standard was developed and keeps being modified by the IEEE 802.16 Working Group on Broadband Wireless Access Standards. The working group is a unit of the IEEE 802 LAN/MAN Standards Committee. The main task of the IEEE 802.16 Working Group is to develop standards to enable a uniform development of Broadband Wireless Metropolitan Area Networks [Sta10].

We also have another entity that influences in the development of the IEEE 802.16 standard. This entity is the WiMAX Forum, which is a non-profit organization that is formed by over 500 companies [PH09], including important companies in/or related to the telecommunications industry such as Motorola, Nokia, Samsung, Sprint, Nextel, Cisco Systems, Fujitsu, and Comcast Corporation among others [For10a]. The main purpose of the WiMAX Forum is to certify that the broadband wireless products manufactured by all enterprises in the industry are compatible and are able to interoperate with other WiMAX products, in order to introduce these items as soon as possible in the market. That is why the Forum works with operators (telephone, Internet, or cable operators) as well as with regulators to meet the demands of the public and also of the government.

2.1 The Importance of WiMAX

In the last few years, we have seen an increase in the use of technology for telecommunications. Since the 1990's, the general public has been using the Internet to connect to the world. The number of Internet users has grown to reach 1 billion during the last 10 years [Par06]. This increase in users is due to the always growing demand for access to the Internet and to telecommunications.

However, since the year 2000 we can also see an ever growing demand of traffic. Users are increasingly utilizing the features provided by the Internet and by their smart phones, personal digital assistants (PDA), and personal computers. They utilize these devices to communicate, but also to download video and audio. Companies are realizing that providing their customers with these triple play services (high speed Internet, video, and audio) requires them to provide broadband access [WIS07]. Also, users want to have ”anytime and anywhere” access, which means that they want to use their triple play services at the time they need them and in any place they go to. This implies that companies have to provide a way for users to connect at home, at their office, and in public places (away from home). That is why the need for broadband that can be accessed anywhere is growing. The fact that it can be accessed anywhere leads us in the direction of wireless broadband services.

Broadband is already provided to 1 billion customers by telephone companies through digital subscriber line (DSL) or cellular technology (to mobile phones), and by cable companies through cable modem. However, there are still 5 billion potential customers in the world that have not been reached by any broadband services. This amount of population is a very tempting business opportunity for broadband service providers. Nevertheless, to reach this population is still a challenge because most people are located in developing countries that do not possess a good communications infrastructure yet. The investment would be too big if companies tried to expand or create a wired copper or cable network. Therefore, the most economic alternative to cover potential customers, who are out of reach of wired networks, would be to make a wireless network that would cover all these users [Par06].

If we analyze more the situation, we will see that to connect people wirelessly is much more economic than DSL and Internet through cable. A mile of copper cable would cost $22,750 and a mile of coaxial cable would cost $29,250. On the other hand, the cost for one mile for a wireless tower (that is, a mile reached by the signal of this tower) would be $11,083 [Par06]. The savings of using wireless towers instead of copper or coaxial cable are considerable.

The money value per mile of the different infrastructures used for connectivity were obtained from [Par06], which is a source from the year 2006. Therefore, these numbers could have varied somewhat with the years but still remain valid to compare the difference in cost.

As we can see, telecommunication companies need to expand their broadband network, and the most economic way to do this is by utilizing wireless technology. This is where WiMAX comes into play, because it provides both: broadband access as well as an economic implementation to provide wireless connection to customers. WiMAX will then reach customers in locations that are remote to wired networks, a concept that is normally called providing ”last mile access”.

To get connectivity service, a user has many technological options such as cellular phone, wireless local area networks (LAN), or Bluetooth among others. These technologies compete in a certain way with WiMAX because they offer connectivity. But none of them offers broadband mobile service, which is the segment where WiMAX is competing [WIS07]. The direct competitor of WiMAX in this segment is long-term-evolution (LTE), a technology created by 3rd Generation Partnership Project (3GPP) which is an organization formed by standardization entities from all over the world as well as by telecommunication companies [ACH10].

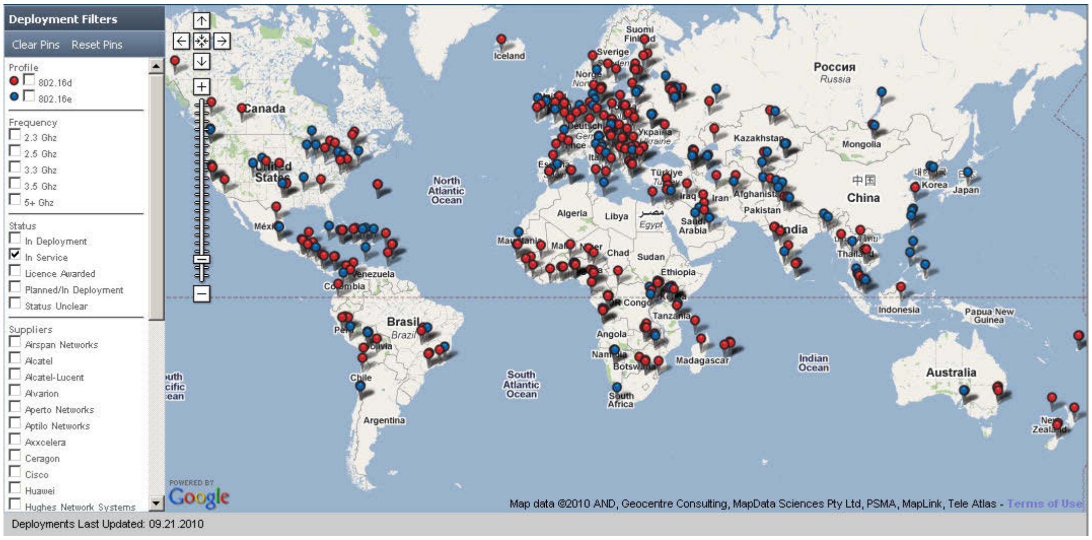
LTE is very similar to WiMAX in the way they operate in the physical aspects, as well as in latency (except for WiMAX 802.16j that has higher latency or delay than LTE when sending data through a RS [YHXM09]), efficiency, and security. Nevertheless, both technologies come from different backgrounds and present some differences. LTE comes from a telecommunications background in which dynamic traffic is given priority (to adapt to the flow of cellular telephone communications). WiMAX on the other hand comes from a networking approach (more oriented to computers and the Internet) and is less dynamic than LTE [ACH10].

The popularity of WiMAX and LTE among telecommunication companies is an important factor in the competition of these two technologies. The director of the WiMAX Forum claims that currently WiMAX antennas could cover up to 650 million users [Con10], a very promising number of potential customers for a telecommunications company. Notwithstanding, there are only 10 million WiMAX users worldwide (a very low number compared to the 650 million potential customers), with 1.7 million subscribers in the United States covered by the service of Clearwire in association with Sprint. On the other hand, companies such as Verizon and AT&T (in the United States) are investing on LTE because it is simpler and cheaper for them to upgrade their current networks to support this standard [Duf10]. These companies do not offer LTE service yet, but they are expecting to attract their current 3G service customers to subscribe to the new LTE service (a number expected to be higher than 200 million subscribers for companies providing LTE service worldwide).

Seeing the technological similarities and differences between the two standards, as well as their popularity among investors, the question that appears now is what technology a company should invest in: LTE or WiMAX. The answer lies mostly in the network's deployment. LTE is a technology that can better be deployed in locations of the world that possess a wired network already (developed countries and big cities), while WiMAX is a technology focused on locations that have minimum or no wired networks (the countryside or developing countries). The difference in the locations where the two technologies can be deployed is what will make both technologies coexist without extinguishing each other in a competition [ACH10]. Therefore, a new entrant operator could implement WiMAX because the investment is not as high as with LTE. In addition, existing operators could use both technologies at the same time, with LTE in areas where they already have a wired network built, and WiMAX in locations without wired networks.

The battle between WiMAX and LTE is already indicating LTE as a winner in popularity among telecommunication companies. Nevertheless, as we just explained, both technologies could coexist in this world. WiMAX would probably be the option to choose in markets not led by LTE (due to the lack of a wired network). However, if WiMAX competed against LTE in a location that already has a wired network that could be upgraded, telecommunication companies will probably choose to invest in LTE [Con10].

WiMAX is currently deployed in several locations of the world as we can see on the map in Figure 1 obtained from [For10b]. Every pin on the map represents the location of the headquarters of that deployment in each country. That way, a company could implement WiMAX in several cities, but in the map it is just marked with one pin for that country. The map only shows deployments of 802.16d and 802.16e versions of WiMAX. The importance of WiMAX is shown by its many deployments that will grow with the coming years.



Deployment Filters

Clear Pins Reset Pins

Profile
- 🔴 □ 802.16d
- 🔵 □ 802.16e

Frequency
- □ 2.3 Ghz
- □ 2.5 Ghz
- □ 3.3 Ghz
- □ 3.5 Ghz
- □ 5+ Ghz

Status
- □ In Deployment
- ☑ In Service
- □ Licence Awarded
- □ Planned/In Deployment
- □ Status Unclear

Suppliers
- □ Airspan Networks
- □ Alcatel
- □ Alcatel-Lucent
- □ Alvarion
- □ Aperto Networks
- □ Aptilo Networks
- □ Axxcelera
- □ Ceragon
- □ Cisco
- □ Huawei
- □ Hughes Network Systems

Deployments Last Updated: 09.21.2010

Figure 1: Locations of the world where WiMAX is currently deployed. Source: [For10b]

2.2 WiMAX History

WiMAX was not created overnight, it was a process that took years. Before WiMAX, the world had wireless cellular telephones. Also, communications started being digitized; meaning by this that triple play services such as the Internet, video and voice were available to the customer all in the same network with one connection. Mobile phones and digitization of communications were the reasons why networks began to grow. Customers increasingly demanded more speed in their connections and wanted to be connected whenever they wanted and in any location they were. This was a challenge for operators who had to provide high bandwidth as well as ubiquitous access (omnipresent access, which is everywhere at the same time) so that customers could access the network with any device, at any moment in any geographical location. There was a need for providers, such as telephone companies and cable operators, to expand their networks in locations where there was no cabling to provide services; remote locations such as the countryside or even entire countries without the infrastructure required to provide these services. This is when the idea of WiMAX was born, with the purpose of providing high bandwidth and ubiquitous access even in places without a wired telecommunication network [WIS07].

Without the needed infrastructure, the idea of WiMAX became a very good alternative to provide communication services. As we mentioned in Section 2.1, most remaining potential customers (the largest amount of world population) live in developing countries that have an underdeveloped cable and only copper networks [WIS07]. Therefore, it is a very tempting business for operators to reach these customers in the cheapest way possible, which implies not deploying a cable network and utilizing wireless technology. Also, these companies could provide a better service to their current customers by giving them higher speed at any time and location.

With these needs in the industry, the original 802.16 standard was created in 1999 giving birth to the first generation of WiMAX. However, its commercial success was very limited because there were not many products available with this technology yet. Line of sight (LOS) was required between the base station (BS) and subscriber stations (SS) because this fixed equipment operated in the 10-66 GHz range and the short radio waves could not go through buildings, a fact that made LOS the only way for two stations to communicate with each other at those high frequencies [AI08]. Equipment operating in that range provided little interference and high data rates; but they were also expensive to manufacture, difficult to install because of the LOS requirement, and could not work with mobile SSs [WIS07].

At that time, there was also lack of an organization responsible to make sure all the equipments manufactured for WiMAX were compatible with each other. This is why the WiMAX Forum was created in 2001 [For10a] to create system profiles based on the 802.16 standard and certify the products to be manufactured to work with it. The system profiles that the Forum creates are wireless network architectures and functionalities for a variety of scenarios [OZV08].

Other versions and amendments of the 802.16 standard were published since the first one in 1999, but it was in 2004 when the second generation of WiMAX began. The second generation improved the drawbacks of the first generation. In 2004, the 802.16-2004 standard was published. This standard provided fixed wireless solutions and included all previous standards to be compatible with them. Later on, the amendment of the 2004 standard was published under the name 802.16-2004/Cor1-2005 [WIS07]. The 802.16-2004 standard and its amendment became the basis for fixed WiMAX. They are compatible with all the previous standards and allow the use of WiMAX with LOS operating in the 10-66 GHz range. They also allow non-line-of-sight (NLOS) because it specifies the use of a frequency range from 2 to 11 GHz. This range does not provide such a high speed as the 10-66 GHz range and has more transmission errors, but it does not require LOS between the BS and the SSs. Also, this version of the standard provided the mesh network topology, in which the SSs connected to other SSs in the network and could use them as relays to exchange data with the BS.

In 2005, the 802.16e-2005 amendment was published including specifications for mobility, becoming the basis for mobile WiMAX. This amendment specifies the operation below 6 GHz, which makes it slower than if it used a higher frequency range but also makes it usable by fixed and mobile SSs [AI08]. WiMAX with these characteristics can operate with LOS as well as with NLOS, making it ideal to provide service indoor. The amendment also supports the use of the multiple input and multiple output (MIMO) technology, which means that the sender can use multiple antennas to send information to the receiver that can also utilize more than one antenna. This way, MIMO improves the reception and increases the rate of transmission between the two communicating nodes [WIS07].

In 2009, the 802.16-2009 standard was released to continue developing WiMAX. In this release, the mesh network topology that was provided in the standard of 2004 was not used anymore. Later during the same year, the 802.16j amendment was released. This amendment provided a way to cover the drawbacks of the previous standards, which were coverage holes within the cell and weakness of signal on the border of cells. The problem was solved by adding relay stations (RS) within each cell. The RSs increase the capacity of the network and also give a better quality of signal. These types of networks are called multihop cellular networks (MCN), because data could pass through more than one link or hop to get from the BS to the SSs (or vice-versa) [PH09].

In order to facilitate the publication of the 802.16j amendment, its designers took into account that this amendment had to be compatible with previous versions of WiMAX because there were some devices compatible with the 802.16e-2005 amendment already developed. This limited the features that could be added to the new 802.16j. With that in mind, most of the changes of this new version were made on the MAC layer, while the physical layer was similar to the 802.16e-2005 amendment. Therefore, the users who bought SSs do not need to modify their equipment since the standard is already compatible with it. Only the BSs have to be modified to implement 802.16j and allow the functioning of RSs in the network [PH09].

Since the first standard to the most recent ones, the size of service cells has shrunk to increase capacity and provide better services to users [AI08]. Also, we can see the growing priority that mobility is given with the newer developments of the WiMAX technology.

2.3 WiMAX With Relays

As we saw, WiMAX has evolved in its history to adapt to the needs of its users, service providers, and vendors. Nevertheless, WiMAX has to take into account some coverage issues since it competes against Internet through cable and against mobile phone service providers. To compete against cable service providers, WiMAX has to be able to provide a reliable and clear signal even at the edge of the coverage cell. To compete against cellular phone providers, WiMAX should be able to cover holes within a service cell in order to provide high mobility [PH09].

To address these coverage issues, WiMAX service providers would need to decrease the size of each cell and add more BSs to create a higher number of smaller cells. The smaller cells would increase the capacity of the overall network but would also create more interference and be very costly for the WiMAX service provider. If new cells were created, the service provider would have to consider the expenses of renting space to place the antennas, it would need wires to connect the extra BSs to the core network (if LOS connection between the BSs were not used), and it would need to buy the BSs which are expensive because of the digital and radio frequency equipment they have [PH09].

Since it would be too costly to add more cells and decrease cell size, a better solution would be to add relay stations (RS) to help to forward data in the communication between the BS and the SSs. RSs help to improve the capacity of the network and throughput, since they act as a BS with which the users interact, relieving the BS from having too many different connections (one for each user) and allowing it to connect to the RS which in turn is connected to several of those users. The RSs also increase coverage since they provide higher signal quality in locations where the BS cannot reach with its own signal [YHXM09]. Also, a RS is cheaper than the BS and it is faster to install [OZV08], which makes it a convenient tool for the wireless broadband network.

The RSs in the new 802.16j amendment can be used in several situations and scenarios, which explains why using RSs might become the best choice to solve the previous 802.16 standard's coverage and poor signal problems. RSs can be used as a fixed infrastructure, to provide coverage within buildings, to provide temporary coverage, or to provide coverage in mobile vehicles [PH09]. We now explain each of these four situations.

The RS can be used as a fixed infrastructure to emulate the way the BS acts. In that case, the RS will be placed in locations where it will serve general traffic. This type of RS will help to increase throughput and coverage, since it will carry data to the BS and cover locations in the cell with better quality of signal than the BS. These RSs will usually be placed on rooftops to have LOS with the BS and obtain a good data rate.

To provide coverage within a building, a RS can be placed near or inside the building. This way, if there is a coverage hole within the building, the RS will cover it. As an example, this can be applied to locations where people do not get cellular phone service such as inside a building, in the subway or in tunnels. These RS would work with NLOS channels, could use battery power, and be very simple since they would just forward data to the nearest RS or directly to the BS.

Temporary coverage can be provided as well with RSs that could operate on batteries. Some situations where this could be applied are emergency or disaster areas where the BSs have been damaged, or next to stadiums when there are events that many people attend.

Finally, the RS would be very useful to provide coverage on a mobile vehicle that transports a large number of people, such as a train or a bus. While in movement, the vehicle would be going through different coverage cells. That is why a RS could be placed on the vehicle. This way, all the users inside the vehicle would be connected to the RS, which would in turn be connected to the BS of the current cell and take care of reconnecting to a new BS if the vehicle exited the current cell.

After explaining the basics of a WiMAX network operating with RSs, we have a broad perspective of what relays are capable of doing to benefit the wireless network. Therefore, the following step is to explain in the next section how the WiMAX network functions.

2.4 How WiMAX Functions

To explain how WiMAX operates, we have to start explaining the basic concepts. The subscriber station (SS) is wireless equipment that allows the device (or devices) of the subscriber of the service (or end user) to access the network and connect to the base station. We also have the base station (BS) which is a radio transmitter and receiver that manages the wireless connection of the SSs to the wireless network, and acts as a gateway to the wired network. The SSs can have fixed access if they have a directive antenna and are fixed in a location (homes or offices), or mobile access if they have an omnidirectional antenna and have the capability of moving geographically without losing coverage within a cell (smart phones) [WIS07]. The link that goes from the BS to the SS is called downlink, and the link that goes from the SS to the BS is called uplink [StIMTS09a].

A BS provides service within an area called cell, which is similar to the way cellular networks operate. The BS antenna points in all directions (omnidirectional) forming a circular cell (even though in literature cells are represented as hexagons). A BS can also have its antenna point in a specific direction to sectorize the cell and share it with other BSs that are pointing in other directions to cover the rest of the cell [Nua07]. Sectorization of the cell provides better service and increases network capacity [AI08].

A network running under the 802.16j amendment would look very similar to a network running under the 802.16 standard. The only difference would be that there are relay stations (RS), which are antennas with equipment to "relay" the information that is exchanged between the BS and the network users. The RS has practically the same functions as the BS, with the difference that the RS does not backhaul data to the core network or backbone, because it does not have wired connection to it, and because the BS takes care of that function [YHXM09]. According to the standard [StIMTS09b], there is no limit in the amount of RSs that there can be between the BS and the SS in a path. Nevertheless, in practice no more than two relays are usually in a path, probably because if the number of RS and SS in a cell is too high the interference would increase and too many transmission errors would be caused, lowering the network throughput [YHXM09]. Also, the throughput of the network is improved when adding RSs because it provides high speed in the links that unite them with other RSs or with the BS, being able to transport data that belongs to several SSs in one time slot [OZV08].

The SSs can only send information uplink to the BS or to the RS, and they cannot send information directly to another SS. The BS can send information downlink to the RS and to the SS. And the RSs can send information uplink to the BS and downlink to the SS, or they can also send data to another RS that is in the communication path between the BS and its users. The station that communicates directly with the SS (be it the BS or the RS) is called access station, and the link between those two stations is called access link, while the link between the BS and an RS or between two RSs is called relay link [OZV08]. The station that is next on the uplink path of another station is called super ordinate. On the other hand, the station that is next in the downlink path of another station is called subordinate [PH09].

In WiMAX we have two network topologies. The first one is point-to-point (PTP) that is a topology in which there is one direct link that connects the BS to each SS (or to another BS). Since the SS has an exclusive link to the BS, it has access to a very high bandwidth and can reach very high data rates. The downside of this is that the cost is very high since the owner of the SS is paying for that link by itself. This topology is utilized by entities that need high amounts of bandwidth such as company buildings or laboratories. Also, PTP can be utilized to link the BS in the fixed cell to the core network or to other BSs wirelessly if there is no wired connection [WIS07]. The second topology is point-to-multipoint (PMP), where several SSs are connected to one BS (possibly through RSs if the 802.16j amendment is utilized). In this case, the SSs share the bandwidth provided by the BS, which brings as a consequence obtaining less speed and reducing the individual costs of utilizing that network. Users of PMP networks are small companies or end users [AI08].

The 802.16 standard allows the usage of two duplexing technologies called frequency division duplexing (FDD) and time division duplexing (TDD). FDD uses one channel to transmit uplink and the other channel to transmit downlink. It is aimed at symmetrical traffic (where the amount of traffic going uplink is the same as the quantity going downlink) and has short delay. On the other hand, TDD utilizes one channel for both uplink and downlink. Nevertheless, only uplink or downlink can be used in a time slot, but not both simultaneously, which is translated into the node not being able to send and receive data at the same time. TDD is used for symmetrical as well as asymmetrical traffic and can make better use of the frequency than FDD because it keeps its only channel occupied most of the time [AI08].

So far we have not explained what parts of the communication system are affected by WiMAX. According to the 802.16 standard and comparing with the Open Systems Interconnection (OSI) model, the layers covered by WiMAX are the Media Access Control (MAC) sublayer and the physical (PHY) layer. The MAC sublayer is actually included within the Data Link layer, which is the layer just above the PHY layer [Tan03].

The MAC sublayer takes care of telling which SS can join the network as well as of scheduling [SIJT09]. This sublayer is divided into 3 sub-sublayers. The first one is the Convergence Sublayer (CS) that transforms the data it receives from the CS service access point (SAP) into MAC packets that contain useful information for the MAC such as the connection identifier (CID) and the service flow. The other functions of the CS are to allow bandwidth allocation and to enable quality of service

(QoS) to reserve resources and accommodate the needs of data flow and users. The second sublayer is the Common Part Sublayer (CPS) that allocates bandwidth, establishes and maintains a connection, and controls access to the network. The third sublayer is the security sublayer, which deals with encryption and decryption of data, authenticating users, and the exchange of keys [Nua07].

The PHY layer takes care of transmitting and receiving data. It also defines many physical characteristics such as the transmission power to be used, the type of transmission signal, and the connection between both sides of a link. Between the PHY and the MAC CPS layers, we can find the physical layer service access point (PHY SAP), which enables these two layers to exchange information [Nua07].

When focusing on the 802.16j amendment, we can see there are different classifications of RS in the MAC and PHY layers. In the MAC layer, the RS is classified according to scheduling and it can use centralized scheduling or distributed scheduling [OZV08]. With centralized scheduling, the RS does not have control over scheduling or security for data transmission because the BS is in charge of those tasks for the whole network. On the contrary, when using distributed scheduling, the RS has control over when its subordinate SSs should start transmitting as well as over bandwidth assignment, and it may also take care of administering traffic security if the BS does not already do so [StIMTS09b].

In the PHY layer, the RS is classified as transparent and non-transparent [OZV08]. The transparent RS only relays data and does not broadcast control messages or a preamble. That is why the SS that is connected to a transparent RS "thinks" it is directly connected to the BS because it does not realize the RS even exists. On the other hand, a non-transparent RS relays data, broadcasts control messages and sends a preamble (just like the BS does). In this case, the SS is logically connected to the RS and "believes" that the RS is in fact the BS [StIMTS09b].

Knowing this, we can summarize that a transparent RS can only use centralized scheduling, since the RSs cannot send any control messages. A non-transparent RS can use centralized or distributed scheduling, since the non-transparent RS has the capability of sending control messages, which is why they are more expensive than transparent RSs (they are more complex in their design) [PH09].

Also, we have to take into account that all RSs can improve throughput, but only non-transparent RSs can extend coverage of the network because they are capable of sending control messages and a preamble [OZV08].

With this basic knowledge of how WiMAX operates, we complete the chapter that gave a brief explanation of the WiMAX technology. Now we continue in the next chapter with scheduling and provide the information we need to understand the rest of the thesis.

Chapter 3

Previous Work on Scheduling and Resource Allocation

In this thesis, we work with scheduling in WiMAX. We have already explained the WiMAX technology in Chapter 2. Therefore, it is now time to explain what scheduling is and how it operates in WiMAX. In Section 3.1 we explain the term scheduling as well as resource allocation, and how these two terms are often confused by some authors in the literature.

We also show some samples of previous works of different authors on scheduling in wireless networks. We divide these works into two categories, according to how they solve the scheduling problems: those that use heuristics without mathematical programming tools in Section 3.2, and those that use heuristics or exact methods with mathematical programming tools in Section 3.3.

3.1 Scheduling and Resource Allocation

Scheduling and resource allocation are very often interlaced in practice and sometimes confused by paper authors. Below, we attempt to clearly define both of these terms independently.

3.1.1 Scheduling

Scheduling is the action of assigning resources over a period of time to the users of a network [CGN$^+$09]. The resources that are allocated by the scheduler to all nodes in the network are the time slots during which each node is allowed to transmit data [SIJT09]. In other words, scheduling deals with looking at the whole picture of how a network operates and transmits traffic through time. With scheduling it is possible to know what nodes are transmitting at each time slot (during a given period of time slots) and the time order in which nodes are assigned resources.

The scheduler is located in the MAC layer (which is within the data link layer) and receives physical information from the resource allocator. Based on the information received, the scheduler will take its decisions on how to allocate resources over time [CTV09].

3.1.2 Resource Allocation

Resource allocation is a mechanism to assign resources to an individual user only taking into account the conditions and needs of that user at the current time slot [CGN$^+$09]. The resource allocator does its task depending on the decision of the scheduler [BBT$^+$07].

The resource allocator (in charge of performing resource allocation) is involved with the physical layer. The main functions of the resource allocator are to avoid interference with other transmitting nodes, to use power efficiently, and to deliver physical information to the scheduler.

3.1.3 Misinterpretation of Scheduling by Paper Authors

In the literature, we can see that authors often confuse the meaning of scheduling with a mixture of resource allocation and time slot estimation. As we can observe in Section 3.3, several papers claim to be performing scheduling, when in fact they only count the amount of time slots it takes to send data from end to end in a network. In these papers, we can see that resources are assigned to nodes to transmit and that the total transmission lasts a certain number of time slots. However, the authors never show that they considered how the resources were distributed over time to the different users of the network. The lack of consideration of how resources are distributed over time is what makes us claim that they do not perform scheduling according to the definition.

3.2 Heuristics

In the literature, we can find several examples of heuristics that perform scheduling. All of these algorithms fit perfectly with the definition of scheduling and clearly show what occurs during each time slot with each node in the network (that is, whether the node transmits or not and on which link). However, none of these investigations show really if the solution they provide is optimal or near optimal. Here are some examples of papers we found focusing on scheduling in wireless networks, and not taking into account details such as QoS.

In [HP09], Hong and Pang designed a linear programming (LP) model for routing and an algorithm for scheduling in IEEE 802.16j WiMAX wireless networks using Time Division Multiple Access (TDMA), which shows what links are transmitting on each time slot. Their simulation results show that their algorithm performs better than previous algorithms with respect to increasing throughput in the network. The authors claim that the results obtained from the algorithm they designed are very close to optimality.

In [GGM09], Ghosh *et al.* elaborated a heuristic for centralized adaptive uplink scheduling in IEEE 802.16j WiMAX networks operating with Orthogonal Frequency-Division Multiple Access (OFDMA). The heuristic calculates which RS or SS will send information on a link in a specific time slot throughout the duration of a frame, based on information such as the amount of RSs and SSs, the conditions of links, and the demands of bandwidth for each node. The resulting schedule takes into account traffic with different priorities by assigning more frequent time slots to nodes with higher priority. The adaptive characteristic of this scheduler is that, after each period (consisting of a given number of frames) was completed, it will re-calculate how to schedule all nodes based on the current demands, link conditions, variation in subchannel rates, and traffic priorities. All frames in the same time period will have the same scheduling structure. An interesting characteristic of this algorithm is the use of priorities, which we do not consider in our model, but could be useful to implement delays in the future.

In [LKD10], Liao *et al.* presented a mechanism that they name "Clique" to perform centralized scheduling in an IEEE 802.16 WiMAX network operating as a mesh using multiple channels to transmit data. The heuristic optimizes throughput by minimizing the number of time slots used to transmit data. Also the buffer size of each intermediary node is optimized. The Clique algorithm has two priority levels where the first level takes into account the priority of sending links by looking at the priority of the packets contained by the sending node. The second priority level takes into account the hop count based on how many hops a packet has gone through already. This algorithm fits perfectly the scheduling definition in the sense that it specifies what happens at each time slot with each node and the traffic that is transmitted. Its results are compared to other three algorithms to demonstrate the efficiency of this algorithm. What is of importance to us in this source is that they utilize buffers to store data in each node, a concept that we utilize in our model as we can see in Chapter 4, Section 4.4.

In [CLY07], Cao *et al.* introduced a heuristic that performs joint routing and scheduling in a WiMAX mesh network. This multi-path routing and centralized scheduling algorithm is shown to increase efficiency and throughput when compared to other algorithms. The mechanism operates by considering a random traffic flow and network layout, choosing the paths that will increase throughput, and at the same time trying to balance the load of the network and to guarantee some QoS features (such as delay). This algorithm might be useful for us in the future to compare ideas on how to add the delay feature to our model.

In [LO09], Lo and Ou developed a centralized routing and scheduling algorithm for an 802.16 wireless network operating in mesh mode. The routing portion of the heuristic selects from a list of possible paths the ones that would optimize traffic flow in the network, considering other links that are transmitting simultaneously. The mechanism they created is compared to other algorithms (including the basic scheduling algorithm presented in the WiMAX standard) and through simulation results, it is shown that Lo and Ou's heuristic provides higher channel utilization. This algorithm applies the definition of scheduling as it should be, since it considers what occurs in the network during each time slot. Nevertheless, no tool is provided in order to assess the quality of the solutions.

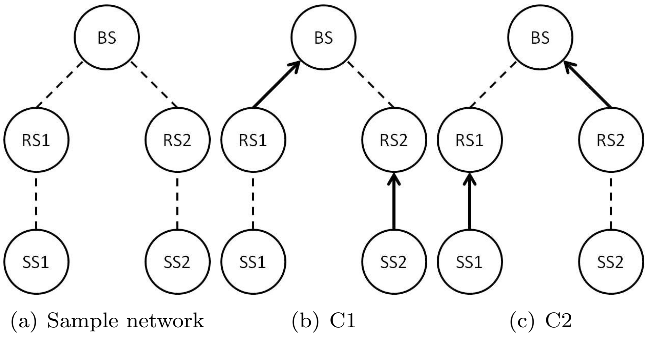
From the list of papers that we just named, the most related to our work is [LKD10]. This research implements buffers (as we do in our model) and also takes into account delay in a certain way with its Clique algorithm that considers priorities for transmitting data.

3.3 Column Generation

In the literature, we can find several papers of authors (such as [ENAJ09, CC06, CFM08, CCF+10]) who created optimization models relying on a mathematical programming model to find the optimal scheduling for a wireless network. We do not analyze classical integer linear programming (ILP) models because they are not scalable for joint routing and scheduling problems. Instead, we focused on finding investigations that utilize the column generation technique to find an optimal solution, since that is the method we utilize for our formulation in this thesis.

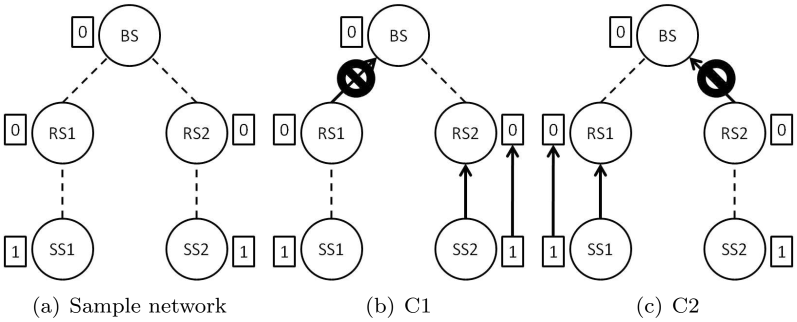
What we found in common in all the papers that we analyze in this section is that none of them does scheduling in the strict meaning of it, considering the definition we provided in Section 3.1. What the following papers call scheduling is to count the number of time slots it takes to send data from end to end (based on the links that can transmit data simultaneously), without taking into account the order in time in which data should be transmitted through the links of the network. The number of time slots they obtain might be a reference to know how long the transmission takes, but we cannot say that it is an exact number that shows the total transmission time, since scheduling is not performed as its definition states.

To make it clearer, we provide an example of a small network that can be seen in Figure 2(a). Most of the models in papers such as [ENAJ09, CC06, CFM08, CCF+10] could give as a valid result that we could use a configuration (a group of links that can be used simultaneously) to send data let us say from RS1 to BS, and from SS2 to RS2. We call that configuration C1 (Figure 2(b)). Another configuration that could be used might be SS1 to RS1 and RS2 to BS, and we name it C2 (Figure 2(c)). These configurations may or may not be chosen by the optimization models in the papers presented, but the fact is that they are potential transmission schemes in most of them. The problem here arises when we want to state the order in time in which these configurations will be used in a network with traffic such as that depicted in Figure 3(a). If we try to use first C1, we should not be able to send data from RS1 to BS because RS1 is only a relay node and does not contain any initial data to send until it has received some from the BS or from the SSs, as can be seen in Figure 3(b). The same way, in Figure 3(c) we show how using C2 first should not be allowed because RS2 does not contain initial data to be transmitted.



(a) Sample network (b) C1 (c) C2

Figure 2: Sample network and two possible configurations in it



(a) Sample network (b) C1 (c) C2

Figure 3: Traffic movement in the network according to configurations

This mistake in the interpretation of scheduling has been done in all the papers we could find related to scheduling optimization utilizing column generation. None of them considers the order in time in which resources should be allocated to nodes in the network, which is the most basic definition of scheduling. We now give the reviews of some of the papers we found in the literature dealing with this topic and show how they are missing the main part of scheduling.

In [ENAJ09], El-Najjar *et al.* designed an optimization model to perform joint routing and scheduling in a WiMAX mesh network and to try to minimize the amount of time that it took to transmit data from end to end. To accomplish this, they utilized the column generation technique, which in each of its pricing problem iterations returned a new configuration (a set of links that can be transmitting simultaneously considering signal to interference-plus-noise ratio (SINR) constraints) that would get the objective of minimizing transmission time closer to its optimal solution. To route data on the network, they formulated a version of the problem that assumed that paths were provided at the outset (limited to the k-shortest paths), and another version that built the paths while the simulation of the model was running (link formulation). They were able to see that the path routing version took less CPU time to be calculated than the link version because it contained less decision variables and provided the same simulation results. They also tested using the maximum power for transmitting on each link, and later they simulated the same situation but only utilizing the just amount of power that was needed to transmit (power aware scheme). They arrived at the conclusion that the power aware scheme was more effective in reusing the wireless spectrum than the maximum transmission power scheme, but with the disadvantage that it took longer CPU time to calculate the solution. This model is able to state how many configurations were utilized to transmit data from end to end. However, it does not comply with the scheduling definition, where we know exactly in which order the nodes will be transmitting in the network throughout time. It is not taken into account whether a node can or cannot send data, considering if it has some data to be sent at a given time slot.

In [CC06], Capone and Carello created an optimization model to minimize the number of time slots needed to transmit data from end to end, considering a wireless mesh network with a time division multiple access (TDMA) scheme and utilizing the column generation method to solve the problem. To arrive at the optimal solution, the formulation performs scheduling and utilizes different transmission power and rates, taking into account the SINR constraints. The problem finds out what combination of links can be used simultaneously in a time slot, and says how many of these configurations are used throughout the data transmission. In 2010, Capone *et al.* performed a similar research in [CCF$^+$10], but also including channel assignment in the optimization model.

In [CFM08], Capone *et al.* proposed an optimization model of joint routing and scheduling for a wireless mesh network using spatial time division multiple access (STDMA) and directional antennas to reduce interference. The authors took into account the fact that the SINR constraint has to be respected and, for that purpose, they included the capability of varying the transmission rates and power during every time slot, and considered the cumulative interference that occurs at receiving nodes caused by transmitting nodes. The objective of the model is to enhance throughput by reducing transmission time and to observe if directional antennas help to improve the results. To achieve this, the authors use the column generation method, which creates configurations that consist of sets of simultaneously transmitting links. This way, the model will choose what the best combination of configurations is and in how many time slots they can be used to obtain an optimal solution. After simulating the network with different parameters, the paper concludes that utilizing directional antennas helps to increase throughput. Nevertheless, the model does not completely fit the definition of scheduling, since it says how many times each configuration was utilized, but it does not state in which time slot order they were active.

In [KSK10], Krishnan *et al.* showed their optimization model that works with column generation and does joint routing and scheduling in a wireless network with multicast flows (a source node can send to several destinations simultaneously) and with the objective of increasing the network's capacity. Each iteration of the column generation problem generates a tree in the pricing section that will lead to a better optimal solution in the master part until the optimal solution was found. In the same study, they solved the same routing and scheduling problems but separately, and compared it to the joint method showing that the joint method is better and returns an optimal solution because it considers facts such as link interference that affect both routing and scheduling.

In [CRH$^+$08], Cao *et al.* designed a column generation optimization model based on a wireless network in order to perform joint routing and scheduling to try to maximize the utility of the network using constraints that consider average power and SINR to avoid interference among transmitting links. The pricing part of the problem returns the power level that would be used in each link simultaneously with other links, which used in the master section would lead to a better optimal solution. Once the optimal solution is obtained and the optimal power to be utilized is found, column generation will be used one more time to find a suboptimal solution to maximize network utility.

In [JX05], Johansson and Xiao presented a column generation model for wireless ad-hoc networks to find the optimal throughput and performance by considering values and aspects such as fairness, routing strategies, scheduling, power allocation, and rates in each link. The mentioned values and aspects are changed and it is shown how they affect the output of the optimal solution for the different networks that they simulate. This work also shows how the different MAC layer modes affect the interaction between the transport layer, network layer and the data link layer and finds the optimal way for these layers to work with each other.

In [YW08], Yang and Wang proposed a joint routing, scheduling, and resource allocation optimization model that utilizes column generation to maximize the utility of a wireless network that has fixed radios. Later on they solve how to allocate the radios within the network by generating another optimization problem. By utilizing multi-radio and multi-channel, they can find how to optimally plan the network to make it perform in the best possible way. The model also considers delay constraints, which could be useful for us to consider if we tried to add these constraints to our model.

In [KWE08], Kompella *et al.* showed a cross-layer optimization formulation that utilizes column generation to minimize the time during which a wireless multi-hop network is actively transmitting data. The pricing part of the column generation problem generates configurations that consist of the links that can be transmitting simultaneously during a time slot, considering the SINR constraint at the receiving nodes, and adapting the transmission power and rate. New configurations are added until transmission time cannot be minimized anymore (an optimal solution has been reached). The authors claim to be using Spatial-TDMA (STDMA) for scheduling, which according to [Amo01] is to divide the network into space slots (areas in the network) and group them into space frames that repeat periodically; something similar to using TDMA but taking into account each section of the network separately and avoiding collisions.

In [ZWZL06], Zhang *et al.* developed a joint routing and scheduling optimization model to minimize the time it takes to transmit data in an ad-hoc, multi-hop, wireless network utilizing multi-channel and multi-radio technologies that will improve the capacity of the network. The formulation utilizes the column generation mechanism because solving this formulation for a network with multi-radio and multi-channel is very complex and the problem needs to be divided into sub-problems. The pricing part of the column generation problem generates patterns of all the links that can transmit simultaneously without interfering other transmissions. In order to avoid interferences, the model uses orthogonal channels. The results obtained from this research show that there is a considerable benefit in transmitting simultaneously through different channels (multi-channel) since several nodes can transmit at the same time without interference, an advantage that becomes clearer in areas of the network that are densely populated by nodes. Also, it is advantageous to utilize multi-radio because the nodes would be able to transmit and receive at any time through different channels, assuming each node had more than one network interface card (NIC).

In [FLH10], Fu *et al.* introduced an optimization formulation with the objective of minimizing the transmission time for a given traffic demand in a wireless network using STDMA, considering SINR constraints and performing power control. To solve this problem efficiently and get integer constraints when allocating time slots, they use the branch-and-price mechanism that combines the column generation method with the branch-and-bound method. Through the pricing part of the column generation problem, they obtain the links that should be active simultaneously in one time slot as well as the power that each node will use when transmitting without causing interferences. Also, an interesting aspect of this research is that, to simplify the complexity of the pricing problem and solve it faster, they used the Perron-Frobenius eigenvalue condition to reduce the running time by 99.86% (when compared to other simulations where this condition is not considered) for networks with 18 links.

In [KWES08], Kompella *et al.* elaborated a joint routing and scheduling optimization model using column generation, where they try to minimize the amount of time it takes to transmit all the traffic demand in a wireless network using STDMA. This model works with the network layer to perform routing, the data link layer (more specifically the MAC layer) to do scheduling, and the physical layer to obtain the SINR constraints. Based on the data provided by these three layers, the pricing part of the column generation problem will select the links that can transmit simultaneously during one time slot. They also compare the results they obtain when having fixed power transmission and when adding power control to each node, which shows that the performance of the network is increased when having power control.

In [MPR08], Molle *et al.* presented a joint routing and scheduling optimization formulation that uses column generation and has as an objective the increase in transport capacity of a wireless mesh network with access to the Internet through gateways. The column generation method helps to generate the sets of transmissions that can be performed during the same time slot, taking into account radio interferences and helping to solve the problem in polynomial time. The authors of this work found through simulations that there should be a certain distance between gateways to avoid interference between them and to improve efficiency in traffic flow.

In [YAS10], Yazdanpanah *et al.* showed a scheduling optimization model for wireless mesh networks, using column generation to minimize the time that the network is active transmitting end-to-end data. They assume that all nodes have smart antennas and use the techniques of beamforming to avoid interference, spatial division multiple access to be able to communicate with more than one node at the same time, and spatial division multiplexing in order to increase the rates of data transmission. Based on the simulation results, the authors observe that using the three techniques we just mentioned and optimizing the scheduling of link transmission helps to improve throughput by increasing the capacity of links as well as the spatial reuse of the spectrum. They also conclude that the time it takes the network to transmit data is reduced by 86.9% when using smart antenna techniques, compared to the same situation but without using smart antennas.

Of the papers we just described, one of the most relevant for our research is [ENAJ09], of which we use the "link-based" routing formulation, where routing paths are calculated during the simulation (and not provided beforehand). We also use an adapted version of [ENAJ09] (Model I in Chapter 4, Section 4.3) to compare to our routing and scheduling model (Model II in Chapter 4, Section 4.4). In addition, [ENAJ09, CC06, CFM08, CCF^{+}10] provided us with the ideas (which we implemented in our model) of utilizing the column generation technique to generate configurations of active links, and adding the SINR constraints to avoid interference between nodes in the network.

Other papers that could be of interest to us are [YW08] and [FLH10]. In [YW08] they consider delay constraints, which could give us some ideas on how to implement those constraints in our model. In [FLH10] they use the branch-and-price mechanism to find an integer solution to the problem, which could be useful for us as well since it involves using column generation with the branch-and-bound method.

Chapter 4

Mathematical Models

In this chapter we present the optimization models that we work with. First, we introduce in Section 4.1 the assumptions we make for the wireless network that we simulate, and in Section 4.2 the notations that are common to Models I and II. Then, we present Model I in Section 4.3 and Model II in Section 4.4. We now briefly describe the purpose of utilizing Model I and Model II.

Model I is based on the optimization model from [ENAJ09] (mentioned in Chapter 3, Section 3.3), where they optimize joint routing and scheduling in a WiMAX mesh network. The model in [ENAJ09] as well as Model I do not truly perform scheduling according to the definition of this term given in Section 3.1. These two models do not provide a way to show how time slots are assigned to nodes throughout time. The authors do not perform scheduling because they do not take into account precedence constraints, and therefore their simulations only obtain a lower bound on the number of required time slots. Note that Model I is a modified version of the original model in [ENAJ09], since we adapted it to work with the 802.16j standard and also made some changes (or minor corrections) that we explain in Section 4.3.1.

We propose Model II as a new optimization model that performs scheduling (according to the definition cited in Section 3.1) within configurations, where each configuration is an ordered sequence of transmission patterns, with one pattern per time slot (which we explain later on). Our Model II is the core part of this thesis and was designed with the purpose of comparing it with Model I, and see how both models differ on the count of the number of time slots it takes to transmit data from end to end in a network (the results of this simulation are shown later, in Chapter 6).

4.1 Assumptions

Before explaining the models in detail, we are going to enumerate the assumptions we make for the network that we simulate. The assumptions include how we allow traffic to flow on the network, as well as other physical aspects that affect data transmission and the operation of the network.

The 802.16j amendment states that the SSs cannot be directly connected with each other, but only to the RS or directly to the BS [StIMTS09b]. Notwithstanding, we assume that there are only links between the SS and the RS, as well as between the RS and the BS, and between a RS and other RSs. That way, we forbid having links directly between the SS and the BS, something that would slow down the data transfer rate of the network because the SS would be occupying the time slot in which a RS could be transmitting to the BS. A RS connected to the BS can make better use of the speed of the link by sending bigger amounts of information contained in its buffer (with this information coming from several SSs).

We consider that a bidirectional link exists between two nodes only if the node of lower hierarchy is within the coverage radius of the node of higher hierarchy (with the BS being the most important, followed by the RS, and lastly by the SS). That is, a link between a BS and a RS only exists if the RS is within the coverage radius of the BS. The same way, a link between a RS and another RS or a SS only exists if the two nodes are within the coverage radius distance of the RS.

We also assume constant rate r_{ij} for each link $(v_i, v_j) \in L$ (defined on Table 2), which is not necessarily the same rate for all links, and transmission power P_i for node v_i. The reason why we specify the link on the rate r_{ij} is that links between the BS and RS (as well as those between two RSs) have a much faster rate than links between the RS and SS. In addition, we specify the node v_i for transmission power P_i because each node has a different power level depending on its type. For example, the BS has more transmission power than the RS, and the RS has in turn more transmission power than the SS.

Other assumptions that we make are the use of only one carrier to transmit, and the consideration of the same transmission tree during all time slots that the transmission lasts (having one tree for uplink and another for downlink). In addition, we use TDMA as channel access method (assigning time slots to each node in the network so that they can transmit data), and transmit with the Time Division Duplexing (TDD) technique (explained in Section 2.4). Based on [ENAJ09], we select a link-based formulation instead of a path-based formulation (the reasons for this choice are explained in Section 4.5.3).

Finally, for Models I and II, we take into account the Signal to Interference-plus-Noise Ratio (SINR) formula described in [CC06, CCF+10]. This formula represents the clarity of a signal received by a node v_j when a transmitter v_i sends data to it, considering interference and noise. In both models we use a derivation of this formula which can be seen in constraints (15) for Model I and (34) for Model II. We cite the SINR formula in (1) and describe its different parameters on Table 1:

$$\mathrm{SINR}_j = \frac{P_i G_{ij}}{\eta_j + \sum\limits_{v_\ell \neq v_i, v_j} P_\ell G_{\ell j}} \geq \gamma_r. \tag{1}$$

Name	**Definition**
P_i	Maximum power at which information can be transmitted from node v_i.
G_{ij}	Environment gain between nodes v_i and v_j.
η_j	Thermal noise at receiving node v_j.
γ_r	SINR threshold associated with rate r.

Table 1: SINR formula parameters

4.2 Common Notations in Model I and Model II

There are some notations that are used in both Model I and Model II. Therefore, to simplify the understanding of both models, we define these notations on Table 2.

4.3 Routing and Scheduling Model I

As we mentioned previously, Model I was taken from [ENAJ09] and adapted by us. Model I uses the column generation method to solve the problem of finding the configurations (a configuration is a

Name	**Definition**
V	Set of nodes in the network.
V_{SS}	Set of all SS in the network.
V_{RS}	Set of all RS in the network.
L	Set of links that can transmit uplink (e.g. from SS to RS), downlink (e.g. from RS to SS), or in both directions (from RS to RS).
L^{UL}	Set of links that can transmit in uplink direction.
L^{DL}	Set of links that can transmit in downlink direction.
$\mathcal{SD}$	Set of source and destination node pairs between which there is some traffic. Note that $\mathcal{SD} \subseteq (\{\text{BS}\} \times V_{\text{SS}}) \cup (V_{\text{SS}} \times \{\text{BS}\})$.
D_{sd}	Demand of data (in bits) from source v_s to destination v_d, with $(v_s, v_d) \in \mathcal{SD}$.
r_{ij}	Flow rate on link (v_i, v_j).
ℓ	Length in time units of a time slot.
P_i	Maximum power at which information can be transmitted from node v_i.
$P_{\max}$	Power at which the "strongest" node in the network (the BS in our case) can transmit.
G_{ij}	Environment gain between nodes v_i and v_j.
$G_{\max}$	Maximum value of G_{ij} for all links $(v_i, v_j) \in L$.
η	Thermal noise.
γ_r	SINR threshold associated with rate r.
M_r	A value equal to $\gamma_r \left(\eta + G_{\max} P_{\max} \frac{\lvert V \rvert}{2}\right)$, for each $r \in R$.
C	Set of configurations c.

Table 2: Parameters used in Model I and Model II

set of active links that are transmitting within one time slot) that will help it to reduce transmission time in a network.

The column generation mechanism is explained in [Chv83], and we show how we utilize it in Chapter 5. But for now, all we need to know is that the column generation method consists of two parts: the Master Problem (in Section 4.3.1) that selects the configurations to minimize transmission time, and the Pricing Problem (in Section 4.3.2) that generates transmission configurations for the Master Problem to choose from.

4.3.1 Master Problem

We now describe the Master Problem and show all the constraints it uses to select the best combination of configurations to reduce transmission time on the network. Let us emphasize once again that a configuration is a set of links that transmit simultaneously during one time slot.

Notations

First, we show the notation that is utilized to make it simpler to follow the explanation of the Master Problem. We present the parameters used on Table 3, and the decision variables on Table 4.

Name	Definition
t_{ij}^{c}	$\in \{0,1\}$, such that $t_{ij}^{c} = 1$ if link $(v_i, v_j) \in L$ is active in configuration $c \in C$, 0 otherwise.
$t_{ij}^{c,\text{UL}}$	$\in \{0,1\}$, such that $t_{ij}^{c,\text{UL}} = 1$ if link $(v_i, v_j) \in L^{\text{UL}}$ is active for uplink in configuration $c \in C$, 0 otherwise.
$t_{ij}^{c,\text{DL}}$	$\in \{0,1\}$, such that $t_{ij}^{c,\text{DL}} = 1$ if link $(v_i, v_j) \in L^{\text{DL}}$ is active for downlink in configuration $c \in C$, 0 otherwise.
W	A value greater than the maximum throughput that can be reached by the network.
M'	A large number used in constraints (11) and (12).

Table 3: Model I. Master Problem parameters

Name	Definition
y_{ij}^{UL}	$\in \{0,1\}$, such that $y_{ij}^{\text{UL}} = 1$ if link $(v_i, v_j) \in L^{\text{UL}}$ is active for uplink, 0 otherwise.
y_{ij}^{DL}	$\in \{0,1\}$, such that $y_{ij}^{\text{DL}} = 1$ if link $(v_i, v_j) \in L^{\text{DL}}$ is active for downlink, 0 otherwise.
z^{c}	Number of times that configuration $c \in C$ is selected.
w_{ij}^{UL}	Amount of data that is sent uplink through link $(v_i, v_j) \in L$.
w_{ij}^{DL}	Amount of data that is sent downlink through link $(v_i, v_j) \in L$.

Table 4: Model I. Master Problem variables

Objective

The objective of this model (reflected in (2)) is to reduce the number of configurations used to transmit all data from end to end. By minimizing the number of configurations used we will also increase throughput (the rate delivery of data) and reduce the number of time slots utilized for transmission, since only one configuration can be used per time slot.

$$\min \sum_{c \in C} z^{c}. \tag{2}$$

Uplink and Downlink Trees Constraints

This model uses a tree structure for its transmission links. Meaning that a node v_i can only have one parent node in the uplink (UL) and downlink (DL) directions. This is reflected in the following constraints:

$$(\textsc{ul}) \sum_{v_j \in V_i^+ \cap (V \setminus V_{\mathrm{SS}})} y_{ij}^{\textsc{ul}} \leq 1 \qquad v_i \in V \setminus \{\mathrm{BS}\}, \tag{3}$$

$$(\textsc{dl}) \sum_{v_j \in V_i^- \cap (V \setminus V_{\mathrm{SS}})} y_{ji}^{\textsc{dl}} \leq 1 \qquad v_i \in V \setminus \{\mathrm{BS}\}. \tag{4}$$

The next two sets of constraints make the connection between the tree structure constraints and the scheduling constraints in this model. Constraint (5) (resp. (6)) indicates that each $y_{ij}^{\textsc{ul}}$ (resp. $y_{ij}^{\textsc{dl}}$) has to be greater or equal than $w_{ij}^{\textsc{ul}}$ (resp. $w_{ij}^{\textsc{dl}}$) divided by W if we have data to send uplink (resp. downlink) through link $(v_i, v_j) \in L^{\textsc{ul}}$ (resp. $(v_i, v_j) \in L^{\textsc{dl}}$). In both constraints, W is a value greater than the maximum throughput that can be reached by the network.

$$y_{ij}^{\textsc{ul}} \geq \frac{w_{ij}^{\textsc{ul}}}{W} \qquad (v_i, v_j) \in L^{\textsc{ul}}, \tag{5}$$

$$y_{ij}^{\textsc{dl}} \geq \frac{w_{ij}^{\textsc{dl}}}{W} \qquad (v_i, v_j) \in L^{\textsc{dl}}. \tag{6}$$

Flow Conservation Constraints

The flow conservation constraints are presented in (7) for uplink and in (8) for downlink. If we observe the left side of constraint (7) (resp. (8)), we can see that the result on the right is based on the amount of bandwidth $w_{ji}^{\textsc{ul}}$ (resp. $w_{ji}^{\textsc{dl}}$) that is sent uplink (resp. downlink). On the right side, both constraints indicate that the total data net flow has to be equal to the negative amount of demand D_{v_i,v_j} for source nodes, to 0 for intermediate RS nodes, and to the positive amount of demand D_{v_i,v_j} for destination nodes. The source nodes are the BS (for DL) or one of the SS (for UL), and the destination nodes are the BS (for UL) or one of the SS (for DL).

$$\sum_{j \in V_i^-} w_{ji}^{\textsc{ul}} - \sum_{j \in V_i^+} w_{ij}^{\textsc{ul}} = \begin{cases} 0 & \text{if } v_i \in V_{\mathrm{RS}}, \\ -D_{v_i,\mathrm{BS}} & \text{if } v_i \in V_{\mathrm{SS}}, \\ D_{\mathrm{BS}}^{\textsc{ul}} & \text{if } v_i = \mathrm{BS}, \end{cases} \tag{7}$$

where $D_{\mathrm{BS}}^{\textsc{ul}} = \sum_{v \in V_{\mathrm{SS}}} D_{v,\mathrm{BS}}$.

$$\sum_{j \in V_i^-} w_{ji}^{\text{DL}} - \sum_{j \in V_i^+} w_{ij}^{\text{DL}} = \begin{cases} 0 & \text{if } v_i \in V_{\text{RS}}, \\ D_{\text{BS},v_i} & \text{if } v_i \in V_{\text{SS}}, \\ -D_{\text{BS}}^{\text{DL}} & \text{if } v_i = \text{BS}, \end{cases} \tag{8}$$

where $D_{\text{BS}}^{\text{DL}} = \sum_{v \in V_{\text{SS}}} D_{\text{BS},v}$.

Bandwidth Constraints

The bandwidth constraints force the model to have the configurations needed to satisfy the traffic demand for uplink (9) going through a link $(v_i, v_j) \in L^{\text{UL}}$, and for downlink (10) going through a link $(v_i, v_j) \in L^{\text{DL}}$. If there were no configurations to satisfy all traffic demands, the model would not be valid, which is why we have to input a set of "dummy" configurations to validate the model and start running its simulation.

$$\ell \sum_{c \in C} t_{ij}^{c,\text{UL}} r_{ij} z^c \geq w_{ij}^{\text{UL}} \qquad (v_i, v_j) \in L^{\text{UL}}, \tag{9}$$

$$\ell \sum_{c \in C} t_{ij}^{c,\text{DL}} r_{ij} z^c \geq w_{ij}^{\text{DL}} \qquad (v_i, v_j) \in L^{\text{DL}}, \tag{10}$$

where ℓ is the length of the time slot, r_{ij} is the rate at which data is transmitted in link $(v_i, v_j) \in L^{\text{UL}}$ (resp. $(v_i, v_j) \in L^{\text{DL}}$), $t_{ij}^{c,\text{UL}}$ (resp. $t_{ij}^{c,\text{DL}}$) is equal to 1 if link (v_i, v_j) is active for uplink (resp. downlink) in configuration $c \in C$.

Correction of Model

We discovered that, in the original version of this model in [ENAJ09], the Pricing Problem could return a configuration with most of its links being part of the transmission tree, making it a valid selection for the Master Problem even if that configuration contained a link that would violate the tree constraints. This is why we added constraints (11) for uplink and (12) for downlink. These new constraints ensure that all the selected configurations respect the tree structure of the network.

$$\sum_{c \in C} t_{ij}^{c,\text{UL}} z^c \leq M' y_{ij}^{\text{UL}} \qquad (v_i, v_j) \in L^{\text{UL}}, \tag{11}$$

$$\sum_{c \in C} t_{ij}^{c,\text{DL}} z^c \leq M' y_{ij}^{\text{DL}} \qquad (v_i, v_j) \in L^{\text{DL}}, \tag{12}$$

where M' is an integer number large enough so that it will make the right side of constraints (11) and (12) be greater or equal than the left side.

Differences with the Original Model

Constraints (9) and (10) differ from the original bandwidth constraint in [ENAJ09] because we added constraints (11) and (12), which demand us to separate the bandwidth constraint in uplink and downlink constraints.

The reason of this division is that in constraints (11) and (12) we need to deal with the uplink and downlink trees separately, because some links (v_i, v_j) may be able to transmit information uplink, downlink, or both. Therefore, if we had an uplink transmission that would make variable $t_{ij}^c = 1$ (such as the variable in the model of [ENAJ09]), we would have to validate that this link belongs to the uplink or downlink tree that is being used. However, since we do not know the direction of transmission, we could incorrectly mark that link as active for the downlink tree (not knowing it is actually an uplink transmission).

That is why we had to introduce the $t_{ij}^{c,\text{UL}}$ and $t_{ij}^{c,\text{DL}}$ variables, to replace the original t_{ij}^c variable and accommodate the need of specifying the direction of transmission, dividing the bandwidth constraint in uplink and downlink. As a consequence, we also had to adapt the Pricing Problem to generate configurations specifying in which direction (or directions) each link would be transmitting.

4.3.2 Pricing Problem - Generation of Configurations

The Pricing Problem generates configurations $c \in C$ based on the dual values that the Master Problem in Section 4.3.1 sends to it. After that, the Pricing Problem returns those configurations to the Master Problem. The cycle, in which the Master Problem sends dual values to the Pricing Problem and then the Pricing Problem returns configurations to the Master Problem, is the column generation mechanism (explained in Chapter 5, Section 5.1). This iteration stops once the optimal solution for the linear relaxation of the Master Problem has been found.

Notation

Before explaining the Pricing Problem's constraints, we give the notation used in its formulation. The parameters are defined on Table 2 and the decision variables on Table 5.

Name	Definition
t_{ij}	$\in \{0,1\}$ such that $t_{ij} = 1$ if link $(v_i, v_j) \in L$ is active, 0 otherwise.
t_{ij}^{UL}	$\in \{0,1\}$ such that $t_{ij}^{\text{UL}} = 1$ if link $(v_i, v_j) \in L^{\text{UL}}$ is active, 0 otherwise.
t_{ij}^{DL}	$\in \{0,1\}$ such that $t_{ij}^{\text{DL}} = 1$ if link $(v_i, v_j) \in L^{\text{DL}}$ is active, 0 otherwise.

Table 5: Model I. Pricing Problem variables

Objective or Reduced Cost

We minimize the objective function of the Pricing Problem, also called reduced cost, which contains the dual values sent by the Master Problem. These dual values are obtained from the constraints containing the decision variable z^c, which in this case are constraints (9), (10), (11), and (12).

$$\begin{aligned} \min = & 1 - \ell \sum_{(v_i,v_j) \in L^{\text{UL}}} u_{ij} t_{ij}^{\text{UL}} r_{ij} - \ell \sum_{(v_i,v_j) \in L^{\text{DL}}} u_{ij} t_{ij}^{\text{DL}} r_{ij} \\ & + \sum_{(v_i,v_j) \in L^{\text{UL}}} u_{ij} t_{ij}^{\text{UL}} + \sum_{(v_i,v_j) \in L^{\text{DL}}} u_{ij} t_{ij}^{\text{DL}}. \end{aligned} \tag{13}$$

We differ in the reduced cost from the original version in [ENAJ09] because we wrote constraints (9) and (10) for uplink and downlink respectively, while in the original version these two constraints are represented by only one. Also, we added constraints (11) and (12), which did not exist in the original version. As we explained in the previous section, this separation of constraints in downlink and uplink was due to the necessity of knowing the direction of transmission of each link in a configuration.

No simultaneous node receiving/transmitting status

The first constraint to create a configuration $c \in C$ is that a node $v_i \in V$ cannot simultaneously transmit to and receive from more than one node [ENAJ09, CCF+10].

$$\left(\sum_{v_j:(v_i,v_j) \in L} t_{ij} + \sum_{v_j:(v_j,v_i) \in L} t_{ji} \right) \leq 1 \qquad v_i \in V. \tag{14}$$

Power Constraint

The next constraint we need is one that forces the sending node $v_i \in V$ to transmit with enough power so that its signal will reach the receiving node $v_j \in V$, considering interference of neighboring nodes as well as other physical factors such as thermal noise. The power constraint (15) is derived from the SINR formula, defined in (1) [CC06, CCF$^+$10].

$$P_i G_{ij} + M_r(1 - t_{ij}) \geq \gamma_r \left(\eta + \sum_{(v_\ell, v_k) \in L; v_\ell \neq v_i} P_\ell G_{\ell j} t_{\ell k} \right) \qquad \forall (v_i, v_j) \in L, r \in R, \tag{15}$$

where the constant M_r is

$$M_r \geq \gamma_r \left(\eta + \sum_{(v_\ell, v_k) \in L; v_\ell \neq v_i} P_\ell G_{\ell j} t_{\ell k} \right) \quad \forall (v_i, v_j) \in L, r \in R, \tag{16}$$

γ_r is the SINR threshold at rate r, η is the background thermal noise (note that, for simplicity, we assume the same thermal noise for all nodes), and P_i is the maximum power that is used to transmit from node v_i to node v_j in order to meet the SINR which is proportional to the amount of interference received at v_j from other transmitting nodes at the given time slot σ. In [CC06, CCF$^+$10] we find the definition for M, which we adapted to have one M for each type of rate in our network (rate depends on the type of node). Therefore, the value of M_r is a constant defined as:

$$M_r = \gamma_{\max} \left(\eta + G_{\max} P_{\max} \frac{|V|}{2} \right) \quad r \in R, \tag{17}$$

where $G_{\max} = \max_{v_i, v_j \in V} G_{ij}$, and $P_{\max} = \max_{v_i \in V} P_i$.

Link Direction

We now need to define some constraints to know the direction in which we are transmitting on a link that will be part of the configuration generated. Directions can be uplink, downlink or both.

First, if $t_{ij} = 1$ and the link (v_i, v_j) transmits only uplink (resp. downlink), constraint (18) (resp. (19)) will make variable $t_{ij}^{\text{UL}} = 1$ (resp. $t_{ij}^{\text{DL}} = 1$), indicating that we are transmitting uplink (resp. downlink).

$$t_{ij} \leq t_{ij}^{\text{UL}} \quad \forall (v_i, v_j) \in L^{\text{UL}} \setminus (L^{\text{UL}} \cap L^{\text{DL}}), \tag{18}$$

$$t_{ij} \leq t_{ij}^{\text{DL}} \quad \forall (v_i, v_j) \in L^{\text{DL}} \setminus (L^{\text{UL}} \cap L^{\text{DL}}). \tag{19}$$

The links between two RSs are able to transmit in one or both directions. For those links we could have three situations: that the transmission goes uplink, downlink or for destinations in both directions. Therefore, if $t_{ij} = 1$ and the link (v_i, v_j) is transmitting uplink, then $t_{ij}^{\text{UL}} = 1$. If $t_{ij} = 1$ and the link (v_i, v_j) is transmitting downlink, then $t_{ij}^{\text{DL}} = 1$. Nevertheless, if $t_{ij} = 1$ and the link (v_i, v_j) is transmitting uplink and downlink simultaneously, then $t_{ij}^{\text{UL}} = 1$ and $t_{ij}^{\text{DL}} = 1$.

$$t_{ij} \leq t_{ij}^{\text{UL}} + t_{ij}^{\text{DL}} \quad \forall (v_i, v_j) \in (L^{\text{UL}} \cap L^{\text{DL}}). \tag{20}$$

Finally, we make sure that variables t_{ij}^{UL} and t_{ij}^{DL} are never greater than t_{ij}.

$$t_{ij} \geq t_{ij}^{\text{UL}} \quad \forall (v_i, v_j) \in L^{\text{UL}}, \tag{21}$$

$$t_{ij} \geq t_{ij}^{\text{DL}} \quad \forall (v_i, v_j) \in L^{\text{DL}}. \tag{22}$$

4.4 Routing and Scheduling Model II

Model II is the model that we created considering the definition of scheduling. This model also uses configurations, which are a set of active links and the amount of data that is sent from a transmitting node (the source v_s) to a receiving node (the destination v_d) during a period of time slots. The difference between a configuration in this model and a configuration in Model I, is that in Model I configurations were only for one time slot and sent always the same amount of data. Instead, in Model II configurations reach a given period of time slots and specify the amount of data sent from a source to a destination.

Similarly to Model I, our Model II utilizes column generation. Therefore, we have the Master Problem (in Section 4.4.1) that selects the most convenient transmission configurations to reduce the amount of time in which data is transmitted on the network. We also have the Pricing Problem (in Section 4.4.2), which generates configurations that are added to the pool of configurations from which the Master Problem will be selecting. Once again, the column generation method is explained in [Chv83] and we demonstrate how we use it in Chapter 5.

Note that in this model, we perform scheduling within each configuration. However, we do not schedule when selecting the configurations since we do not give an order in which they should be used.

4.4.1 Master Problem

As we just mentioned, in the Master Problem we choose among different configurations to find the optimal ones that will increase the network's throughput by reducing the amount of time that it takes to transmit all data from source to destination.

Notations

As we did for Model I, we now define the notations of the parameters and variables used in the Master Problem of Model II. The parameters are defined on Table 6, and the decision variables on Table 7.

Name	**Definition**
Σ	Set of time slots spanned by the scheduling and resource allocation model.
a^c_{sd}	Traffic (in bits) carried from source v_s to destination v_d in configuration $c \in C$.
b^c	Transmission delay (in time slots) in configuration $c \in C$.
Σ^c	Set of time slots spanned by configuration $c \in C$, with $\Sigma^c \subseteq \Sigma$. This parameter is not utilized in the constraints of the Master Problem. Σ^c is used in the Pricing Problem as Σ and indicates the maximum number of time slots that can be used in the configuration that is being created.
$x^{c,\text{UL}}_{ij}$	(resp. $x^{c,\text{DL}}_{ij}$) $\in \{0,1\}$, such that $x^{c,\text{UL}}_{ij} = 1$ (resp. $x^{c,\text{DL}}_{ij} = 1$) if link (v_i, v_j) is used for uplink (resp. downlink) transmission in configuration $c \in C$, 0 otherwise.

Table 6: Model II. Master Problem parameters

Name	**Definition**
y^{UL}_{ij}	$\in \{0,1\}$, such that $y^{\text{UL}}_{ij} = 1$ if link $(v_i, v_j) \in L^{\text{UL}}$ is active for uplink, 0 otherwise.
y^{DL}_{ij}	$\in \{0,1\}$, such that $y^{\text{DL}}_{ij} = 1$ if link $(v_i, v_j) \in L^{\text{DL}}$ is active for downlink, 0 otherwise.
z^c	Number of times configuration $c \in C$ is selected.

Table 7: Model II. Master Problem variables

Configuration Definition

As we mentioned earlier, a configuration is a set of active links that transmit during a given period of time slots. The configuration also indicates the amount of data that is sent from each source node

v_s to each destination node v_d. From that definition, we have to explain that a configuration $c \in C$ is defined by the parameters listed here:

- a_{sd}^c is the bandwidth carried from v_s to v_d in configuration c.
- b^c is the number of time slots used by configuration c.
- $x_{ij}^{c,\text{UL}}$ and $x_{ij}^{c,\text{DL}}$ are the parameters that indicate which links are used in configuration c.
- Σ^c indicates the maximum number of time slots that can be used by a configuration c.

The Σ^c parameter is not utilized in any constraint of the Master Problem, but it is used in the Pricing Problem (represented as Σ) to set the maximum number of time slots that can be used in a configuration. All these parameters can also be seen on Table 6.

Objective

The objective in this problem, as we mentioned earlier, is to reduce the amount of time it takes to transmit end-to-end traffic in a network (which can also be stated as increasing throughput). To accomplish this, we have the following objective function:

$$\min \sum_{c \in C} b^c \, z^c, \tag{23}$$

where b^c is the amount of time used by configuration c, and z^c equals 1 if configuration c is used (0 otherwise).

Demand

Each demand of data from a source node v_s to a destination node v_d has to be fulfilled by the amount of data that is transmitted in the configurations that are selected.

$$\sum_{c \in C} a_{sd}^c \, z^c \geq D_{sd} \qquad (v_s, v_d) \in \mathcal{SD}, z^c \in \mathbb{Z}^+, c \in C. \tag{24}$$

Scheduling

We next need to make sure that the number of time slots used by all the selected configurations is never greater than the total number of time slots Σ:

$$\sum_{c \in C} b^c z^c \leq |\Sigma|. \tag{25}$$

Uplink/Downlink Tree Structures

We built two tree structures, one of which is for uplink transmission and the other for downlink transmission. Both of these trees can overlap or not, as we can observe in Figure 4. Having non-overlapping transmission trees may help to reduce the transmission delay, as we will explain soon.

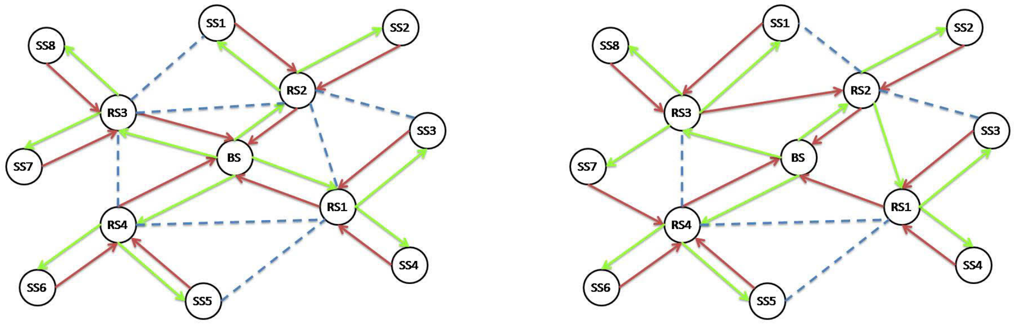
To build the mentioned trees, we have constraints (26) and (27). For the uplink direction (26), only one outgoing link is allowed for each node (except for the base station node, since it does not send anything uplink) in order to guarantee an uplink tree transmission structure. Likewise, for the downlink direction (27), we only allow one incoming link for each node (except for the base station node, since it does not receive any downlink data from any other node) in order to guarantee a downlink tree transmission structure. The tree constraints can be understood in a simple way as allowing each node to have only one parent node in a tree that has the base station as its root.

$$(\text{UL}) \quad \sum_{v_j \in V_i^+ \cap (V \setminus V_{\text{SS}})} y_{ij}^{\text{UL}} \leq 1 \qquad v_i \in V \setminus \{\text{BS}\}, \tag{26}$$

$$(\text{DL}) \quad \sum_{v_j \in V_i^- \cap (V \setminus V_{\text{SS}})} y_{ji}^{\text{DL}} \leq 1 \qquad v_i \in V \setminus \{\text{BS}\}. \tag{27}$$

Consistency: Transmit on the Tree Structures

The following constraints associate the tree structure of the network with the scheduling part of the problem. Constraint (28) indicates that if the link $x_{ij}^{c,\text{UL}} > 0$ and configuration c is selected at least once throughout the transmission of data, then the link (v_i, v_j) must be active forming part of the uplink tree and making $y_{ij}^{\text{UL}} = 1$. Likewise, in constraint (29) if the link $x_{ij}^{c,\text{DL}} > 0$ and configuration c is selected at least once throughout the transmission of data, then the link (v_i, v_j) must be active forming part of the downlink tree and making $y_{ij}^{\text{DL}} = 1$. In both of these constraints we multiply y_{ij}^{UL}



(a) Overlapping UL and DL transmission trees

(b) Partially-overlapping UL and DL transmission trees

Figure 4: Overlapping and partially-overlapping transmission trees

and y_{ij}^{DL} by a number large enough so that each constraint in which it is present will never have both of its sides (on the right and left of the $\geq$ sign) equal. The large number is equal to $(1 + |\Sigma|)$, which is 1 plus the total number of time slots considered by the problem.

$$(1 + |\Sigma|)\; y_{ij}^{\mathrm{UL}} - \sum_{c \in C} x_{ij}^{c,\mathrm{UL}}\, z^c \geq 0 \qquad (v_i, v_j) \in L^{\mathrm{UL}}, \tag{28}$$

$$(1 + |\Sigma|)\; y_{ij}^{\mathrm{DL}} - \sum_{c \in C} x_{ij}^{c,\mathrm{DL}}\, z^c \geq 0 \qquad (v_i, v_j) \in L^{\mathrm{DL}}. \tag{29}$$

In this problem we make the assumption that the UL and DL trees may overlap or not. In Figure 4 we can observe that there are uplink (red, or dark if you have a black and white copy of this thesis) and downlink (green, or clear if color is lacking) transmission links, as well as radio links represented by dotted lines for the nodes that are within radio distance of each other. We can observe that the advantage of the partially-overlapping transmission tree in Figure 4(b) is that it uses more transmission links than the overlapping tree seen in Figure 4(a).

The fact that the partially-overlapping transmission tree uses more links helps to reduce the number of time slots required to do at least one transmission from each source node to each destination node (from the BS to all the SS, and from all the SS to the BS). If each node makes one transmission, the difference in time slots between overlapping and partially-overlapping trees can be seen on Table 8. Therein, we can observe that it takes 12 time slots for the overlapping tree structure to transmit data, while it takes 10 time slots for the partially-overlapping tree structure to perform the same task (16.67% less time slots).

Time	**Links Used for Each Tree Type**	
Slot	**Overlapping**	**Non-Overlapping**
1	(SS1-RS2), (SS3-RS1), (SS5-RS4), (SS7-RS3)	(SS1-RS3), (SS2-RS2), (SS3-RS1), (SS5-RS4)
2	(SS2-RS2), (SS4-RS1), (SS6-RS4), (SS8-RS3)	(SS8-RS3), (SS4-RS1), (SS6-RS4)
3	(RS1-BS)	(RS3-RS2), (RS1-BS), (SS7-RS4)
4	(RS2-BS)	(RS2-BS)
5	(RS3-BS)	(RS4-BS)
6	(RS4-BS)	(BS-RS2)
7	(BS-RS1)	(BS-RS3), (RS2-RS1)
8	(BS-RS2), (RS1-SS3)	(BS-RS4), (RS2-SS2), (RS1-SS3), (RS3-SS1)
9	(BS-RS3), (RS1-SS4), (RS2-SS1)	(RS1-SS4), (RS3-SS7), (RS4-SS5)
10	(BS-RS4), (RS2-SS2), (RS3-SS7)	(RS3-SS8), (RS4-SS6)
11	(RS3-SS8), (RS4-SS5)	
12	(RS4-SS6)	

Table 8: Active links for overlapping and non-overlapping trees

While using a partially-overlapped transmission tree might be beneficial to reduce the number of time slots utilized for transmission, we also have to take into account the amount of traffic to be transmitted by each node. For example, the tree may change its shape if we have a greater amount of traffic sent by some nodes in the network. Therefore, the best overlap combination of links will depend on the set of transmission links, but also on the traffic. Taking these facts into account, our model identifies what the best transmission trees are to reduce the number of time slots it takes to transmit data.

4.4.2 Pricing Problem - Generation of Configurations

The Pricing Problem of Model II creates configurations based on the dual values obtained from the Master Problem of Section 4.4. Here we present the constraints needed to generate a configuration within a time frame, performing real scheduling by assigning time slots to different nodes so that they can transmit.

4.4.3 Notations

Once again, we present the notations used in this problem before introducing the constraints, so that it is easier to find what each parameter and variable are. The parameters for the Pricing Problem are defined on Table 2 and on Table 9, and the decision variables on Table 10.

Name	**Definition**
Σ	Maximum number of time slots that can be used to transmit.

Table 9: Model II. Pricing Problem parameter

Name	**Definition**
a_{sd}	Traffic (in bits) carried from source v_s to destination v_d.
$B_v^{sd,\sigma}$	Load of the buffer located at node v in the uplink ($v_s \in V_{\text{SS}}$)/downlink ($v_s = \text{BS}$) direction during time slot $\sigma \in \Sigma$.
b	$\in \mathbb{Z}^+$. Maximum transmission delay (in time slots) from source to destination.
$\alpha_{ij}^{sd,\sigma}$	$\in \{0,1\}$, such that $\alpha_{ij}^{sd,\sigma} = 1$ if link (v_i, v_j) is transmitting from v_s to v_d, during any time slot $\sigma' \geq \sigma$, 0 otherwise.
t_{ij}^{σ}	$\in \{0,1\}$, such that $t_{ij}^{\sigma} = 1$ if link $(v_i, v_j) \in L$ is transmitting during time slot σ.
$t_{ij}^{sd,\sigma}$	$\in \{0,1\}$, such that $t_{ij}^{sd,\sigma} = 1$ if link (v_i, v_j) is transmitting from v_s to v_d, during time slot σ, 0 otherwise.
x_{ij}^{UL}	(resp. x_{ij}^{DL}) $\in \{0,1\}$, such that $x_{ij}^{\text{UL}} = 1$ (resp. $x_{ij}^{\text{DL}} = 1$) if $(v_i, v_j) \in L^{\text{UL}}$ (resp. $\in L^{\text{DL}}$) is an active link in the uplink (resp. downlink) tree transmission structure, 0 otherwise.
φ_{ij}^{sd}	Amount of bandwidth flow carried on link (v_i, v_j) during the overall time period of the Pricing Problem, for traffic flow from v_s to v_d.
$\varphi_{ij}^{sd,\sigma}$	Amount of bandwidth flow that is carried on link (v_i, v_j) during time slot σ, for traffic flow from v_s to v_d.

Table 10: Model II. Pricing Problem variables

Objective or Reduced Cost

In the Pricing Problem, the objective function corresponds to the so called reduced cost. The reduced cost is built from the dual values of the constraints that contain the decision variable z^c from the Master Problem. For more explanations on how we form the reduced cost, refer to Section 5.1.

The objective here is to minimize the transmission delay (the terms with b) and maximize the amount of bandwidth transmitted (the term with a_{sd}, which is negative).

$$\min \left(b - \sum_{(v_s,v_d) \in \mathcal{SD}} u_{sd} a_{sd} + ub \right), \tag{30}$$

where u_{sd} and u are the non negative values of the dual variables associated with constraints (24) and (25) respectively.

Note that we should have included the dual values associated with constraints (28) and (29) in the reduced cost, but we did not do so because their value is always 0 in the optimal Linear Programming (LP) solution of the Master Problem. This happens because of the complementary slackness property of dual values [NW88], where in the optimal LP solution of the Master Problem we have an optimal value for z^c which implies that the optimal dual value u_{ij}^{UL} multiplied by the left hand side of constraint (28) should equal 0. In our model, constraint (28) will never equal 0 because $(1 + |\Sigma|)\ y_{ij}^{\text{UL}} \neq \sum\limits_{c \in C} x_{ij}^{c,\text{UL}} z^c$. Therefore, to satisfy the complementary slackness property and find the optimal solution for the LP we need to have $u_{ij}^{\text{UL}} \left((1 + |\Sigma|)\ y_{ij}^{\text{UL}} - \sum\limits_{c \in C} x_{ij}^{c,\text{UL}} z^c \right) = 0$. Having the constraint portion of this formula never equal to 0, the dual value u_{ij}^{UL} will be forced to be 0 to comply with the property that we mentioned before. The same occurs for DL in constraint (29).

If constraint (28) were $|\Sigma|\ y_{ij}^{\text{UL}} = \sum\limits_{c \in C} x_{ij}^{c,\text{UL}} z^c$, the formula would reach equality in two situations. The first situation would be if we considered one time slot per configuration $c \in C$ and if we used one configuration throughout all time slots $\sigma \in \Sigma$, which can only happen if we have the BS directly connected to a SS and only exchanging information with it in UL direction. The second situation where this could happen, is if we did not use any link or configuration to transmit data, which would make $y_{ij}^{\text{UL}} = 0$ and $z^c = 0$. In our simulations, the first situation will never happen because we exchange data between more than one SS and the BS, with at least one RS acting as an intermediary node in the transmission. If the second situation occurred, we would not need to calculate the optimal solution at all because no data is being sent. That is why we made it impossible for the two portions of the constraint to be equal in any situation by adding 1 to $|\Sigma|$ and leaving constraint (28) as it looks now. The same situation occurs for DL in constraint (29).

Now that we have explained how the reduced cost is formed, we can introduce the constraints for the Pricing Problem.

Active Links

When data is transmitted through a link (v_i, v_j) during a certain time slot $\sigma \in \Sigma$, that link is considered active. However, since we are simulating how the data moves from node to node during each time slot, we also need to store information on what the source and destination of each data unit are. Therefore, we have to specify which link is active transmitting data that goes from a source v_s to a destination v_d. In one time slot σ there can be data passing through a single link with several sources and destinations, which is why we have to specify in constraint (31) that a link t_{ij}^{σ} is active if it transmits at least once from any source to any destination.

$$t_{ij}^{\sigma} = \max_{(v_s,v_d)\in\mathcal{SD}} \max_{\sigma\in\Sigma} t_{ij}^{sd,\sigma} \qquad (v_i, v_j) \in L. \tag{31}$$

Constraint (31) can then be rewritten in order to reduce it to a linear constraint:

$$t_{ij}^{\sigma} \geq t_{ij}^{sd,\sigma} \qquad (v_i, v_j) \in L, (v_s, v_d) \in \mathcal{SD}, \sigma \in \Sigma. \tag{32}$$

No simultaneous node receiving/transmitting status

In order to be valid, during a given time slot σ, a configuration must guarantee that a node $v_i \in V$ cannot transmit and receive simultaneously [ENAJ09, CCF$^+$10].

$$\sum_{v_j:(v_i,v_j)\in L} t_{ij}^{\sigma} + \sum_{v_j:(v_j,v_i)\in L} t_{ji}^{\sigma} \leq 1 \quad v_i \in V, \sigma \in \Sigma. \tag{33}$$

The first summation on the left-hand side is associated with the links on which v_i sends traffic, while the second summation is associated with the links from which v_i receives traffic.

Power Constraint

Next, we need to make sure that a transmitting node $v_i \in V$ transmits with enough power so that its signal will reach receiving node $v_j \in V$ without interference of other neighboring transmitting nodes. To do this, we utilize a derivation of the SINR formula defined in (1) [CC06, CCF$^+$10]:

$$P_i G_{ij} + M_r(1 - t_{ij}^{\sigma}) \geq \gamma_r \left(\eta + \sum_{(v_\ell,v_k)\in L; v_\ell \neq v_i} P_\ell G_{\ell j} t_{\ell k}^{\sigma} \right) \qquad \forall (v_i, v_j) \in L, r \in R, \tag{34}$$

where

$$M_r \geq \gamma_r \left(\eta + \sum_{(v_\ell, v_k) \in L; v_\ell \neq v_i} P_\ell G_{\ell j} t^{\sigma}_{\ell k} \right) \qquad \forall (v_i, v_j) \in L, r \in R, \tag{35}$$

γ_r is the threshold of the SINR at rate r, η is the background thermal noise, and P_i is the maximum power that is used to transmit from node v_i to node v_j in order to meet the SINR which is proportional to the amount of interference received at v_j from other transmitting nodes at the given time slot σ. The constant M_r is defined in formula (17).

Flow Conservation

The flow conservation constraint (36) indicates how much data has to remain in each node at the end of the configuration. In our case, the source nodes will remain with $-a_{sd}$ because they are sending that amount of data. The destination nodes will remain with a_{sd} because that is the data they are demanding. The intermediary nodes (the RSs) remain in 0 because they send any data that they receive so that the data arrives at its destination.

$$\sum_{v_j \in V_i^+} \varphi^{sd}_{ji} - \sum_{v_j \in V_i^-} \varphi^{sd}_{ij} = \begin{cases} 0 & \text{if } v_i \in V_{\text{RS}}, \\ -a_{sd} & \text{if } v_i = v_s, \\ a_{sd} & \text{if } v_i = v_d, \end{cases} \qquad v_i \in V, \{v_s, v_d\} \in \mathcal{SD}. \tag{36}$$

The summation on the left-hand side is the data being received, while the summation on the right of it is the data sent by node v_i. Note that if $v_s \in V_{\text{SS}}$ and $v_d = \text{BS}$, constraint (36) is associated with uplink flows; while if $v_s = \text{BS}$ and $v_d \in \text{SS}$, constraint (36) is associated with downlink flows.

Flow Decompositions and Limitations

Next, we specify in (37) that the total amount of data being transmitted on each link (v_i, v_j) is equal to the sum throughout all time slots $\sigma \in \Sigma$ of data transmitted through that same link, with any source and destination pairs $(v_s, v_d) \in \mathcal{SD}$.

$$\varphi^{sd}_{ij} = \sum_{\sigma \in \Sigma} \varphi^{sd,\sigma}_{ij} \quad (v_i, v_j) \in L, (v_s, v_d) \in \mathcal{SD}. \tag{37}$$

Also, as we can see in (38), the amount of data $\varphi_{ij}^{sd,\sigma}$ sent during one time slot σ through link (v_i, v_j) cannot be greater than the rate r_{ij} allowed for that link during the duration ℓ of the time slot.

$$\sum_{(v_s,v_d)\in\mathcal{SD}} \varphi_{ij}^{sd,\sigma} \leq \ell r_{ij} \quad (v_i, v_j) \in L, \sigma \in \Sigma. \tag{38}$$

Similarly, in (39) we specify that we cannot send more data through a link than what the rate allows us (such as stated in (38)), but with the addition that data from a specific source v_s to a destination v_d can only be transmitted if the decision variable $t_{ij}^{sd,\sigma}$ is equal to 1, meaning that the link has to be active for transmission for the same source and destination.

$$\varphi_{ij}^{sd,\sigma} \leq \ell r_{ij}\, t_{ij}^{sd,\sigma} \quad (v_i, v_j) \in L, (v_s, v_d) \in \mathcal{SD}, \sigma \in \Sigma. \tag{39}$$

We state as well in (40) that no data is transmitted during time slot 0 because that is the time slot in which all the nodes specify the data with which they begin the configuration, only being able to transmit (subtract from their buffer) or receive (add to their buffer) data starting from time slot 1 onwards.

$$\varphi_{ij}^{sd,0} = 0 \quad (v_i, v_j) \in L, sd \in \mathcal{SD}. \tag{40}$$

Counting the number of time slots used

One configuration uses a maximum of b time slots. However, it can also use less than that number of time slots to transmit data from a source to a destination on the network. To keep track of how many time slots are really used by the network to transmit data from a specific source v_s to a destination v_d, we use the decision variable $\alpha_{id}^{sd,\sigma}$. This variable will equal 1 only when data is being transmitted from a source v_s to a destination v_d, and will equal 0 otherwise. That way, if we sum the values of $\alpha_{id}^{sd,\sigma}$ for one v_s and v_d pair, we will know the time slots it takes to transmit data from that source to its destination.

To calculate the value of $\alpha_{id}^{sd,\sigma}$ we need to elaborate some constraints. First, $\alpha_{id}^{sd,\sigma}$ can never be greater than the number of time slots b used by the configuration, as we can see in constraint (41).

$$b \geq \sum_{\sigma\in\Sigma} \sum_{v_i\in V_{\mathrm{RS}}:(v_i,v_d)\in L^{sd}} \alpha_{id}^{sd,\sigma} \quad (v_s, v_d) \in \mathcal{SD}. \tag{41}$$

Next, the value of $\alpha_{id}^{sd,\sigma}$ has to be greater than or equal to the value of the active link (v_i, v_j) for a source v_s and destination v_d in a time slot σ, as well as greater than or equal to $\alpha_{id}^{sd,\sigma+1}$ (that is, the same α variable but for the following time slot). These last two conditions are explained with an example on Table 11, and expressed in constraint (42) which is linearized in constraints (43) and (44).

$$\alpha_{id}^{sd,\sigma} \geq \max\{t_{id}^{sd,\sigma}; \alpha_{id}^{sd,(\sigma+1)}\} \quad \sigma \in \Sigma, v_i \in V_{\text{RS}} : (v_i, v_d) \in L^{sd}, (v_s, v_d) \in \mathcal{SD}, \tag{42}$$

$$\alpha_{id}^{sd,\sigma} \geq t_{id}^{sd,\sigma} \quad \sigma \in \Sigma, v_i \in V_{\text{RS}} : (v_i, v_d) \in L^{sd}, (v_s, v_d) \in \mathcal{SD}, \tag{43}$$

$$\alpha_{id}^{sd,\sigma} \geq \alpha_{id}^{sd,(\sigma+1)} \quad \sigma \in \Sigma, v_i \in V_{\text{RS}} : (v_i, v_d) \in L^{sd}, (v_s, v_d) \in \mathcal{SD}. \tag{44}$$

	Time Slot σ								$\vert\Sigma\vert$	$\vert\Sigma\vert + 1$
t_{ij}^{σ}	0	0	1	0	1	0	1	1	0	-
α_{ij}^{σ}	1	1	1	1	1	1	1	1	0	0

Table 11: Illustration of the use of the α variables

As we show in constraint (44), $\alpha_{id}^{sd,\sigma}$ has to be greater than the same variable but in the following time slot. However, when we arrive at the last time slot that the configuration will be able to handle, we have to add one more time slot for making $\alpha_{id}^{sd,(|\Sigma|+1)}$ a valid variable. This extra time slot is not included in the configuration and will always give the value of 0 to this last α variable. We can see this on the right-hand side of Table 11, and add constraint (45) to enforce the situation that was just mentioned:

$$\alpha_{id}^{sd,(|\Sigma|+1)} = 0 \qquad v_i \in V_{\text{RS}} : (v_i, v_d) \in L^{sd}, (v_s, v_d) \in \mathcal{SD}, \tag{45}$$

where $L^{sd} = L^{\text{UL}}$ if $v_s \in V_{\text{SS}}, L^{\text{DL}}$ otherwise.

Basically, we can summarize these constraints saying that the $\alpha_{id}^{sd,\sigma}$ variables are equal to 1 as soon as one $t_{id}^{sd,\sigma'}$ is equal to 1 for any $\sigma' \geq \sigma$. Also, as long as we are minimizing b in the reduced cost, those constraints are sufficient to force variables $\alpha_{id}^{sd,\sigma}$ to take their smallest possible value by satisfying constraint (42).

Buffer Constraints

To simulate the transmission of data from node to node in the network we need to have buffers which are represented by the $B_v^{sd,\sigma}$ variables. We have to remark that each node $v \in V$ does not only have one but several buffers. The reason for this is that we have to keep track of the source v_s and destination v_d of the data going through node v, in order to push it (the data) through the network and help to deliver it as fast as possible to avoid delays. The variable for the buffer has to specify as well the time slot σ on which it is showing information because a node might send or receive data and the buffer variable has to keep this information current.

Initialization. To show proper information on the data they contain, buffers have to be initialized at time slot $s = 0$. As we demonstrate in constraint (46), the buffers of source nodes are initialized containing the data D_{sd} that they will send to destination v_d. On the other hand, in constraint (47) we specify that the intermediate and destination nodes will be initialized with the value 0 since they do not have any data in them. Note that $B_v^{sd,0}$ denotes a downlink buffer if $v_s = \text{BS}$ and an uplink buffer if $v_s \in V_{\text{SS}}$.

$$B_v^{sd,0} = D_{sd} \qquad v = v_s \in V_{\text{SS}} \cup \{\text{BS}\}, (v_s, v_d) \in \mathcal{SD}, \tag{46}$$

$$B_v^{sd,0} = 0 \qquad v \neq v_s, v \in V, (v_s, v_d) \in \mathcal{SD}. \tag{47}$$

Buffer capacity. To make our model more realistic, we have to put a limit on the capacity of the buffers (otherwise if they are infinite the model would not be too credible). We assign a very high capacity to the end nodes (the BS and all the SS) since they have to store all the data that they will be sending and receiving. The RS nodes are intermediary and their capacity will be much smaller than that of the end nodes we just mentioned.

The reason for the RS buffers being smaller is that they are only there to forward data and not to store it. If we have a buffer that is too big for the RS, they might take more time to forward data since they are able to store it, causing delay. If we have a buffer that is smaller than the rates of the links connected to the node, the speed of transmission will never reach the maximum rate because not enough information is stored to reach it. That is, if we have a rate of 100 units and the buffer's capacity is 50 units, the transmission speed will only reach 50 units because there is no more data to be transmitted than the one stored. For that reason, it is necessary to choose a good buffer size for the RS in order to push data as soon as it arrives and take advantage of the maximum rates allowed.

The way to limit the capacity of buffers is ruled by constraint (48). This constraint indicates that the sum of all the buffers of one node v cannot be greater than the total capacity $\overline{B}_v$ of that node for time slot σ.

$$\sum_{(v_s,v_d)\in\mathcal{SD}} B_v^{sd,\sigma} \leq \overline{B}_v \qquad v \in V, \sigma \in \Sigma. \tag{48}$$

Buffer load updating. When scheduling, every movement of data in the network has to be registered. That is why the buffers have to be updated every time slot with the amount of data they have sent and received, to know the amount of data that they currently have. The following constraint indicates how the variable $B_v^{sd,s}$ obtains its values for time slots greater than 0 (the initial time slot). Constraint (49) states that the variable $B_v^{sd,s}$ is equal to the value of the same buffer in the previous time slot (that is why this constraint only applies to time slots that are greater than the first time slot), plus what node v receives, minus what it sends. The change in each buffer is due to what it receives or sends, denoted by the φ decision variable that indicates the amount of bandwidth flow transmitted.

$$\begin{aligned} B_v^{sd,\sigma} =& B_v^{sd,\sigma-1} + \sum_{v'\in V:(v',v)\in L} \varphi_{v'v}^{sd,\sigma} - \sum_{v'\in V:(v,v')\in L} \varphi_{vv'}^{sd,\sigma} \\ & v \in V, (v_s,v_d) \in \mathcal{SD}, \sigma \in \Sigma : \sigma > 0. \end{aligned} \tag{49}$$

Tree structure

We include a tree structure in the Pricing Problem similar to the one we have in the Master Problem, in order to obtain configurations with valid trees that can be utilized in the Master Problem. To do this, we need to introduce constraints (50) and (51) which state that a node v_i (except for the BS) can have at most one parent node in the transmission tree.

$$\sum_{v_j \in V_i^+ \cap (V \setminus V_{\mathrm{SS}})} x_{ij}^{\mathrm{UL}} \leq 1 \qquad v_i \in V \setminus \{\mathrm{BS}\}, \tag{50}$$

$$\sum_{v_j \in V_i^- \cap (V \setminus V_{\mathrm{SS}})} x_{ji}^{\mathrm{DL}} \leq 1 \qquad v_i \in V \setminus \{\mathrm{BS}\}. \tag{51}$$

Also, we include constraints (52) and (53) to limit the variable $t_{ij}^{sd,\sigma}$ which should be equal to 1 if the link (v_i, v_j) belongs to the tree and 0 if it doesn't. Without constraints (52) and (53), variable $t_{ij}^{sd,\sigma}$ could take the value of 1 for links that would not be part of the tree.

$$|\Sigma| \, x_{ij}^{\mathrm{UL}} \geq \sum_{\sigma \in \Sigma} t_{ij}^{sd,\sigma} \qquad (v_i, v_j) \in L^{\mathrm{UL}}, (v_s, v_d) \in \mathcal{SD} : v_s \in V_{\mathrm{SS}}, \tag{52}$$

$$|\Sigma| \, x_{ij}^{\mathrm{DL}} \geq \sum_{\sigma \in \Sigma} t_{ij}^{sd,\sigma} \qquad (v_i, v_j) \in L^{\mathrm{DL}}, (v_s, v_d) \in \mathcal{SD} : v_s = \{\mathrm{BS}\}. \tag{53}$$

4.5 Comments

4.5.1 Delay

Delay is a constraint that is not directly taken into account in our model. We address it indirectly by including buffers which contain the source and destination of data, and by using configurations that send data from end to end and last a small number of time slots. This way, we try to push data through the network so that it reaches its destination as fast as possible. However, we should include actual delay constraints because they would influence scheduling decisions and are part of providing QoS.

To introduce delay constraints at this moment would make our model (even more) non-scalable because we would have to keep track of time limits of the data packets for each node, depending on the type of traffic. The way our model is developed makes it difficult to simulate scheduling with configurations that have a maximum of 6 or 7 time slots. If on top of that we took into account delay, simulations might take a very long time to run and it is not certain whether the computers we have access to would be able to process such a big model.

Nevertheless, instead of using an optimization model, we could use a heuristic to introduce delay constraints. A heuristic would not provide security in obtaining guaranteed near optimal solutions, but it would be much faster to run and would require less computer memory. In addition, addressing the delay constraints through a heuristic would allow us to choose the order in which configurations are selected, making our model (with the heuristic) a complete scheduling model (based on the scheduling definition).

4.5.2 Buffer

The buffer constraints in our model help to make it non-scalable, because they are directly or indirectly associated with variables that grow as we increase the number of time slots per configuration. We could also say that taking into account the number of time slots (enabling real scheduling) in the mentioned variables is the factor responsible for making our model non-scalable. However, considering the different time slots and having buffer constraints are two concepts that in our model go together. Therefore, in our model, in order to perform real scheduling we need to have buffers.

As we can see on Table 12, the variables in the Pricing Problem that grow as we increase the number of time slots per configuration are: $B_v^{sd,\sigma}$, $\varphi_{ij}^{sd,\sigma}$, t_{ij}^{σ}, $t_{ij}^{sd,\sigma}$, and $\alpha_{ij}^{sd,\sigma}$. Of those variables, $B_v^{sd,\sigma}$ and $\varphi_{ij}^{sd,\sigma}$ are directly related to the buffer constraints. The other variables are indirectly related to them, and deal more with the real scheduling portion of the problem.

The results seen on Table 12 were obtained from the Pricing Problem after running the simulation of a network shown in Section 6.2 of Chapter 6. Looking at how the total number of variables increases from time slot to time slot, we can see that the increase is always 725. The same way, the number of constraints will increase by 1198 if we increase the number of time slots per configuration. This constant increase in variables and constraints might not mean too much as it is presented. But once we observe the results obtained in Chapter 6, we will see how computer memory requirements and CPU time increase rapidly as the number of time slots per configuration increases.

Even though buffers and enabling real scheduling make the program difficult to run for big instances, they allow us to show the status of the network in each time slot. This characteristic of our model is what makes it fit partly the definition of scheduling. We say ”partly” because even

Time Slots per Configuration	**4**	**5**	**6**	**7**
a_{sd}	40	40	40	40
$\alpha_{ij}^{sd,\sigma}$	830	996	1162	1328
b	1	1	1	1
$B_{v}^{sd,\sigma}$	632	790	948	1106
φ_{ij}^{sd}	166	166	166	166
$\varphi_{ij}^{sd,\sigma}$	664	830	996	1162
t_{ij}^{σ}	664	830	996	1162
$t_{ij}^{sd,\sigma}$	272	340	408	476
x_{ij}^{DL}	35	35	35	35
x_{ij}^{UL}	35	35	35	35
Total Variables	3343	4068	4793	5518
Increase in Variables	N/A	725	725	725
Total Constraints	5694	6892	8090	9288
Increase in Constraints	N/A	1198	1198	1198

Table 12: Model II. Number of variables and constraints for the Pricing Problem

though the state of the buffers is shown during the time slots a configuration lasts, we cannot see in which order these configurations are selected throughout the whole period of time analyzed (not only the time of a configuration). However, as we mentioned earlier, we could make our program fully comply with the scheduling definition by adding a heuristic that would give an order in which configurations should be used.

4.5.3 Path vs. Link Models

In the scheduling model formulated in [ENAJ09], the authors analyzed the efficiency of running a model based on paths compared to a model based on links to build the transmission tree. The authors concluded that the path-based formulation was more efficient because it took less time and used fewer variables to provide the same solution as the link-based formulation. The fact that the path-based formulation used fewer variables is because of not considering all possible paths.

However, we used the link-based formulation in our model because it considers all paths on the network. The path-based formulation only considers the k shortest paths (3 in the case of [ENAJ09]) and does not evaluate all alternatives. There might not be a significant difference in simulation results if we select the link or path-based formulation for a small network. Notwithstanding, in a network with 3 or more hops and several possible paths in it, we could find a more exact solution with the link-based formulation than with the path-based formulation (which might not consider several paths that could be useful to improve the solution).

Using the link-based formulation in our model, we make sure that we select the optimal paths. Nevertheless, if we wanted to use the path-based formulation and find the optimal solution for the problem, we could add a second Pricing Problem that would be in charge of generating optimal transmission paths. That way, we would not have to analyze all possible paths, since the Pricing Problem would only generate those paths that would help to obtain a better solution.

4.5.4 Comparing Model II to Model I

As we can see in Chapter 6, if we compared Models I and II, having Model II with a limited number of time slots per configuration, we would be able to observe that Model I gets as a result a considerably lower number of time slots for transmission than Model II. This happens because we are not able to make configurations in Model II bigger than a certain number of time slots. If we were able to assign more time slots to each configuration, the results of Model II would probably approach more those of Model I.

Therefore, to compare these models on more similar grounds, we have to run a simulation in our model (Model II) first, and then based on the output and results, we will feed that information into Model I. Model II depends on us to give it the number of time slots we want each configuration to last as a maximum to transmit a given amount of information. On the other hand, Model I only gets as an input the amount of data we want to transmit. That is why, from Model II we will obtain the configurations that are selected and the information they transmit from each source to destination, and we will feed them as input to Model I.

That way, we will run Model I once for each configuration selected in Model II, considering the amount of data sent from each source to destination in the configuration. Once we have run one instance of Model I for each configuration of Model II, we will know how many time slots it takes to transmit them in Model I (and also in Model II).

Chapter 5

Solving the Models

In this chapter we talk about how we solve the large scale integer linear programming (ILP) Models I and II. To solve these two models we use the column generation tool described in [Chv83] (see also [GG61] and [GG63]). Note that we only show how we use the tool to solve both models. Therefore, this chapter is by no means an explanation of the column generation method, but only a description of how we use it.

When trying to solve large scale models, we usually have to consider a big number of variables (maybe millions), and we have to explore all the possible solutions given by choosing among all possible columns to arrive at the optimal solution. In our case, a column is associated to a vector that includes the decision variables which form a configuration. With the column generation tool, instead of considering all different possibilities when solving the problem, we only generate columns if they improve the objective value of the solution of the linear programming (LP) model. That way, we arrive at the optimal solution faster, using less computational resources, and only taking into account some solutions (as opposed to all).

By using the column generation tool, we will first solve the linear relaxation (LP) of the two models (I and II). However, once we have found the optimal LP solution, we will have to find an ILP solution that has an objective value tentatively close to that of the objective of the optimal LP solution. To find such an ILP solution we can round the values of variables (as mentioned in

[Chv83]). We could also use other techniques such as branch-and-price, explained in [Van94]. In our case, we use our own heuristic and round the values of specific configurations as we explain later in this chapter.

We use Figure 5 to aid us in our explanation on how we solve both models (I and II). If we observe carefully that figure, we see that it has two cycles: a small cycle within a bigger cycle. The inner cycle is the column generation process, which we describe on Section 5.1. The outer cycle, described in Section 5.2, is the process by which we try to reduce the gap percentage to get the ILP objective as close as possible to the optimal LP objective (obtained from the column generation cycle).

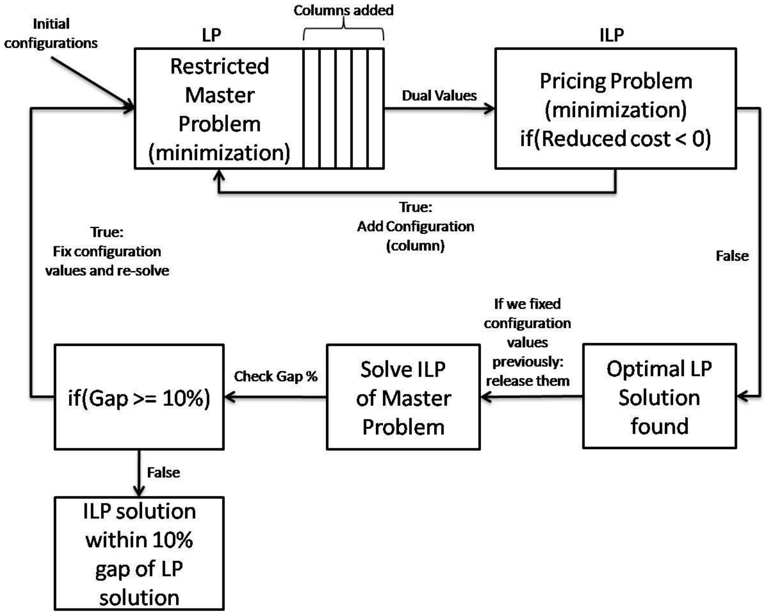

Initial configurations
LP
Columns added
ILP
Restricted Master Problem (minimization)
Dual Values
Pricing Problem (minimization) if(Reduced cost < 0)
True: Add Configuration (column)
True: Fix configuration values and re-solve
False
If we fixed configuration values previously: release them
if(Gap >= 10%)
Check Gap %
Solve ILP of Master Problem
Optimal LP Solution found
False
ILP solution within 10% gap of LP solution

Figure 5: Column generation method and gap percentage reduction

5.1 Column Generation

As we mentioned earlier, in Figure 5 we show how we solve Models I and II. In this section we explain how we utilize column generation (the smaller inner cycle on that figure) to obtain an optimal LP solution.

The first step is to solve the restricted master problem (RMP) by selecting configurations from a given initial configuration set. We call it RMP because it is restricted to a subset of configurations, as opposed to the master problem (MP), which considers all possible configurations. However, for simplicity, we refer to the RMP as ”master problem” in other chapters of this thesis.

As we said, we solve the RMP with an initial configuration set. This set is formed by ”dummy” configurations which we create especially so that they do not provide an optimal solution to the model. The main purpose of the ”dummy” configuration set is to help getting the column generation mechanism started by giving the RMP a valid solution when solved for the first time.

The objective of the MP (in Models I and II) is to minimize the number of configurations (and therefore minimize the amount of time) used to transmit data from end to end in a WiMAX network. Also, note that in column generation we solve the MP as a LP.

After solving the RMP, we will obtain dual values from the constraints that are related to the decision variable that is located in the objective function. In both Models I and II this decision variable is z^c, for $c \in C$ being the configuration.

The dual values that were obtained after solving the RMP are then used by the pricing problem (PP) to be part of its objective function (called reduced cost). The PP is the problem that will provide new columns to the RMP in the column generation method.

To form the objective function of the PP, we have to take into account that if we are minimizing in the RMP, we will also minimize in the PP. In addition, the reduced cost is composed by different terms that consist of dual values (obtained from the RMP) multiplied by a decision variable. The decision variable that accompanies each dual value in the objective function of the PP, is used as a parameter in the constraint in the RMP of which the dual value was obtained. Also, if in the objective of the RMP the decision variable z^c is accompanied by a parameter on the same term, this parameter will be added as a decision variable in the reduced cost of the PP.

To define the sign of each term in the reduced cost of the PP we have to consider two factors. First, we have to look at the position of the decision variable z^c in the constraint where it is present in the RMP and on the sign of its term in that constraint. That is, if the decision variable is on the left hand side of the $\geq$ sign its term in the reduced cost will be negative. On the other hand, if the decision variable is on the left of the $\leq$ sign its term in the reduced cost will be positive. However, if the sign of the decision variable's term in its constraint of the RMP is negative, the variables on the left hand side of the $\geq$ sign will have a positive sign and those on the left of the $\leq$ sign will have a negative sign.

In theory, after solving the PP we should obtain its optimal solution. However, in practice, it is common to obtain the first feasible solution of the PP. In our case, we consider the best of the first 3 feasible solutions of the PP, since it helps us to arrive faster at an optimal solution than considering only the first feasible solution (which would generate too many configurations for the RMP).

Since we are minimizing, if the objective of the PP is less than 0 the decision variables of that solution will form a configuration $c \in C$ that will be added as a column to the RMP. This new configuration helps to ameliorate the objective of the RMP. After this, the cycle begins once again and the RMP is solved, sending its dual values to the PP. The cycle will repeat until the objective function of the PP is not any longer less than 0, which will mean that we have reached the optimal solution for the LP in the RMP.

5.2 Reducing the Gap Percentage

Now we explain the outer cycle in Figure 5. This cycle represents the way we obtain an ILP solution closer to the optimal LP solution of the RMP, after utilizing the column generation tool.

Once the program has found the optimal LP solution of the RMP, we have to find an ILP solution, which we do by solving the same mathematical program as an ILP with the configurations obtained during the column generation process. After solving the ILP, we have to compare its value with the LP optimal value of the RMP. The way to compare these two values is by checking the gap percentage, where the gap percentage is equal to $\frac{|\mathrm{OBJ}_{\mathrm{ILP}} - \mathrm{OBJ}^*_{\mathrm{LP}}| \times 100}{\mathrm{OBJ}^*_{\mathrm{LP}}}$ (with $\mathrm{OBJ}_{\mathrm{ILP}}$ being the value of the objective of the ILP, and $\mathrm{OBJ}^*_{\mathrm{LP}}$ being the optimal value of the solution of the LP). If the value of the gap represents a lower percentage than 10%, we can say that we have obtained a guaranteed accurate solution. That being said, if our gap is 10% or more, we will have to perform again the column generation cycle to hopefully obtain a better ILP solution.

As we said, if our gap is 10% or more, we will once again try to obtain new configurations through the column generation method in order to narrow the gap between OBJ_{ILP} and OBJ^*_{LP}. To accomplish our goal, we can fix the values of some of the decision variables that represent configurations. In this situation, we use first Algorithm 1, and if we still need to reduce the gap we apply Algorithm 2.

In Algorithm 1, we fix the value of the first configuration variable z^c that has a value greater than 0 in the ILP solution (we fix it to its value in the ILP solution). Then, we solve the column generation problem and, once we found an optimal LP solution, we release the variable of the configuration we had fixed previously. After this, we measure the gap between OBJ_{ILP} and OBJ^*_{LP}, and if the new gap is smaller than the gap we had before fixing the configuration (previous gap), we will select once again the first configuration we find in the ILP solution (it could be a different configuration than the one we fixed when starting this algorithm), and perform once again the cycle of fixing its value, solving the LP relaxation of the current RMP, and releasing this configuration's fixed value. However, if the new gap is not better than the previous gap, then we will select the next configuration that has a value in the ILP solution, fix its value and solve again the LP relaxation of the current RMP. If there are no more configurations with a value to be selected in the ILP solution, that means we have already fixed one time all the configurations in the ILP solution and we were not able to reduce further the gap using Algorithm 1.

The whole iteration cycle in Algorithm 1 consists of fixing the configuration's value, solving the LP relaxation of the current RMP, releasing the configuration's fixed value, measuring the gap, and finding the next configuration to be fixed. This cycle is performed until we have found a gap smaller than 10%, or until we do not have any more configurations to fix.

If after applying Algorithm 1 we were still not able to find a small gap percentage, we will have to use Algorithm 2. In Algorithm 2, we fix the value of variable z^c for each configuration that has a value greater than 0 in the LP as well as in the first ILP solution of the pool of solutions (the configuration has to have a nonzero value in both LP and ILP, if it does not, it will not be fixed). The ILP has a pool of several solutions with the same objective value but with different values

Algorithm 1 Fix one configuration

```
$c =$ First configuration with a value $> 0$ in the ILP solution
$c_{flag} = 0 \leftarrow$ Flag that indicates if there are no more configurations to be fixed
$Gap \leftarrow$ Gap percentage between the LP and ILP objectives
while $Gap > 10\%$ and $c_{flag} = 0$ do
  Fix the value of variable $z^c$ with configuration $c$ to its value in the ILP solution
  Solve the column generation problem
  Release the fixed value of configuration $c$
  if $Gap_{new} < Gap$ then
    $Gap_{new} = Gap$
    $c =$ first configuration with a value $> 0$ in the ILP solution
  else
    if $c$ is not the last configuration with a value in the ILP solution then
      $c =$ next configuration that has a value in the ILP solution
    else
      Set $c_{flag} = 1$ to end the algorithm
    end if
  end if
end while
```

for its decision variables. Then we solve the LP relaxation of the current RMP, release the fixed configurations, and check for the gap percentage. If the new gap is smaller than the previous gap, we fix once again the configurations present in both the LP and the first ILP solution of the pool of solutions. However, if the new gap is not smaller than the previous gap, we will fix the configurations of the next ILP solution in the solution pool (again, the configurations have to be selected in both LP and ILP solutions). If there are no more solutions to be selected in the ILP solution pool, then we stop this algorithm.

Similarly to Algorithm 1, the iteration cycle in Algorithm 2 fixes the values of the selected configurations, solves the LP relaxation of the current RMP, releases the fixed configurations, measures the gap, and finds the next ILP solution pool to fix its corresponding configurations (those that are also selected in the LP solution). The cycle is performed until the gap is smaller than 10%, or until we do not have any more ILP solution pools to choose from (since we have used all of them already to fix their corresponding configurations).

After fixing the configuration values individually (Algorithm 1) or as a group (Algorithm 2), we always obtained, in practice, a solution with a gap lower than 10%. However, there is no theoretical guarantee that this will always be the case. If after applying Algorithm 2, we cannot reduce the gap below the target value, then we would need to use another technique such as generating cuts. This

Algorithm 2 Fix all configurations that have a nonzero value in both LP and ILP solutions

```
$ILP_{sol}$ = First solution selected from the pool of ILP solutions
$c_{flag} = 0 \leftarrow$ Flag that indicates if there are no more configurations to be fixed
$Gap \leftarrow$ Gap percentage between the LP and ILP objectives
while $Gap > 10\%$ and $c_{flag} = 0$ do
  $Both[c]$ = Array of configurations $c$ with a value $> 0$ in both LP and $ILP_{sol}$ solutions
  Fix the value of variable $z^c$ for all configurations $c$ that are in $Both[c]$
  Solve the column generation problem
  Release the value of the $c$ configurations that we fixed previously
  if $Gap_{new} < Gap$ then
    $Gap_{new} = Gap$
    $ILP_{sol}$ = First solution selected from the pool of ILP solutions
  else
    if $ILP_{sol}$ is not the last solution in the pool of ILP solutions then
      $ILP_{sol}$ = Next solution in the pool of ILP solutions
    else
      Set $c_{flag} = 1$ to end the algorithm
    end if
  end if
end while
```

is how we obtain a guaranteed accurate solution for our simulation of Models I and II.

Chapter 6

Numerical Results

In this chapter we explain the numerical results obtained when simulating Models I and II. We first present the data sets in Section 6.1, where we talk about the parameters, the formulas used to obtain various parameters, and we show the network scenario that we use for the simulation. In Section 6.2, we explain the experiments that we performed using Model II, such as using different numbers of time slots per configuration, having non-homogeneous and unbalanced traffic on the network, changing the buffer sizes of the RSs, and comparing Model I and Model II.

Before we start explaining the data sets and the results of the experiments, we have to give some background on how the simulation was programmed. The network and its optimization models were developed using IBM ILOG OPL IDE, version 6.3. The computer utilized to program the simulation is a Dell PC running on Windows XP, with an Intel Core 2 Duo 2.33 GHz CPU, and 2 GB of RAM. The results of the simulation were obtained by running it on the Cirrus cluster at Concordia University, which runs on Linux Redhat and has 608 2.2 GHz AMD processor cores, of which we were able to use 1 node with 4 cores and 8 GB of memory for each simulation instance submitted. The software used to run the simulations on the cluster was OPL (Optimization Programming Language), which is part of the software package IBM ILOG CPLEX Optimization Studio, version 12.2.

6.1 Data Sets

To simulate a WiMAX network we need to have some realistic parameters that will help us to obtain results that are as close as possible to reality. The parameters we use, their values, and the source we obtained them from are shown on Table 13. These parameters that are used to build the network can also be seen in some of the constraints of Model I and II.

Parameter	**Value**	**Source**
Standard	802.16j	[MNP+06]
Rate	BS = 100 Mbps, RS = 20 Mbps	[MNP+06]
Coverage Radius	BS = 1 km., RS = 0.6 km.	[MNP+06]
Power	BS = 20 Watts, RS = 10 Watts, SS = 0.1 Watt	[MNP+06]
Frequency	2.5 GHz.	[MNP+06]
Total Bandwidth	10 MHz	[MNP+06]
Frame duration (ms.)	10 ms.	[CRK06]
Frame duration (time slots)	256 time slots	[CRK06]
Multiple Access	TDMA	[MNP+06, ENAJ09]
Duplex Mode	TDD	[SKK+10]
Sub-Carriers	1	[ENAJ10]
Links	Bidirectional links	[ENAJ09]
Antenna Height	BS = 50 m., RS = 30 m., SS = 1.5 m.	[MNP+06]

Table 13: Parameters assumed for our network

We obtained several parameters from [MNP+06], which is an unpublished presentation made by WiMAX industry experts for an IEEE 802.16 session. The parameters that we found in that source, we were also able to find them in a variety of other sources with very similar values. However, we decided to utilize as many parameters as possible from [MNP+06] to make our network have more uniform values (something that would not occur if we obtained each parameter from a different source).

Note that the rate values on Table 13 were transformed into bits per time slot and rounded up in order to allow our simulation to run faster. That is why, we actually use the following values for rate: BS $= 100Mbps = \frac{100{,}000{,}000bits}{1{,}000ms} = \frac{100{,}000bits}{ms} \times \frac{1frame}{1frame} = \frac{100{,}000bits}{ms} \times \frac{10ms}{256slots} = \frac{3906.25bits}{slot} \sim \frac{4{,}000bits}{slot}$, RS $= 20Mbps = \frac{20{,}000{,}000bits}{1{,}000ms} = \frac{20{,}000bits}{ms} \times \frac{1frame}{1frame} = \frac{20{,}000bits}{ms} \times \frac{10ms}{256slots} = \frac{781.25bits}{slot} \sim \frac{800bits}{slot}$. In this case, the actual rates that we are using are $\frac{4{,}000bits}{slot} = 102.4Mbps$ for BS, and $\frac{800bits}{slot} = 20.48Mbps$ for RS.

In order to obtain some of the parameters we need for the constraints in Model I and II, we need to use some formulas. Table 14 shows the parameters we need for the mentioned constraints, as well as the formulas to obtain them, and the source where we can find these formulas.

Parameter	**Value**	**Source**
Thermal Noise	η is obtained from the formula $\eta = -174 + 10 \times log_{10}(B)$, where B is the bandwidth. This value gives us the thermal noise in dB, but we transform it to mW to work with the same type of unit throughout the whole simulation.	[CdM08]
Path loss factor	To obtain the path loss factor p we use the formula $p = a - bh_s + \frac{c}{h_s}$, where a, b, and c are predetermined values of a standard terrain type. In our case we chose terrain type C, which is flat with light tree density and makes $a = 3.6$, $b = 0.005$, and $c = 20$. Parameter h_s is the height of the s station, which is the BS, or can also be the RS if we are dealing with a link that connects two RSs or a RS to a SS.	[CdM08]
Gain	To obtain the gain between two nodes (v_i, v_j) we use the formula $G_{ij} = d_{ij}^{-p}$, where d_{ij} is the distance between nodes v_i and v_j, and p is the path loss factor.	[ENAJ10, ENAJ09, CCF+10]
Required SINR	Obtained using the Shannon capacity theorem and assuming a channel with white Gaussian noise, with the formula: $\gamma_r = 2^{\frac{r}{B}} - 1$, where γ_r is the SINR threshold for rate r, and B is the bandwidth used.	[ENAJ10, EN11, Ned02]
M Parameter	Used in constraints (16) and (35), and defined in (17).	[CC06, CCF+10]
SINR	SINR is given by formula (1) in Chapter 4.	[ENAJ10, ENAJ09, CCF+10]

Table 14: Formulas used for our network

To simulate data flow, we use the network pictured on Figure 6. In that figure, we can see that the triangle is the base station (BS), the squares are the relay stations (RS), and the diamonds are the subscriber stations (SS). The scale in the X and Y axis in the figure are units in meters and represent the geographical location of each station.

Note that we also did experiments with other network topologies. Notwithstanding, the results of simulations with other topologies did not show different characteristics when comparing them to the results of the topology in Figure 6. That is why we did not include in this chapter any results of simulations with any other networks.

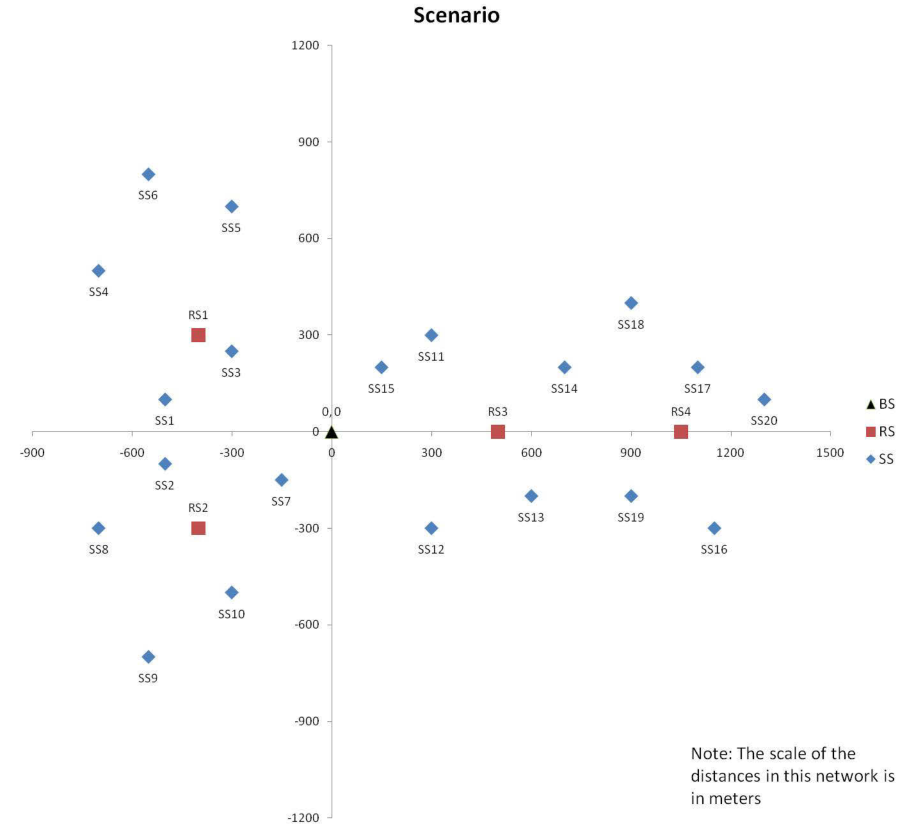


Figure 6: WiMAX network scenario

The possible links are only between the SS and the RS, between the RS and the BS, and between two RS. This means that we do not consider links between the BS and SS (to force the use of RS), or between two SS (the 802.16j amendment states that this is not allowed [StIMTS09b]). Also, links can only exist if the two nodes involved are within the distance that can be reached by the coverage radius of the BS or the RS (see coverage radius values on Table 13). With this said, we assume that if one node can reach the other one, there is a bidirectional link.

6.2 Experiments

6.2.1 Instance 1 - Time Slots Per Configuration

The first experiment we did was to try different numbers of time slots for configurations. Within each scenario, we can only assign one value for the number of time slots that the configurations are allowed to have. Therefore, we have one scenario for each number of time slots.

The issue with assigning the number of time slots is that we were only able to simulate it for values between 4 and 7 slots per configuration. We did not perform experiments with 3 time slots because it would not provide an interesting solution, since only a small number of data packets would be transmitted by few nodes. If we assign less than 3 time slots per configuration, the simulation will not run correctly since for some configurations we need at least 3 time slots to transmit from end to end (for example, a transmission following the path $SS20-RS4-RS3-BS$). Also, if we go beyond 7 time slots per configuration, the simulation will use too much time and too many resources to calculate a solution. For 8 time slots for example, to calculate the final ILP solution, we will need a much greater amount of time and memory than what we need to calculate the solution for 7 time slots.

The computer that calculates the solution has only 8 GB of memory per node, and with 8 or more time slots per configuration we pass that amount of memory, which crashes the program as well as the node that runs the simulation. These computer limitations could be solved by improving our algorithm and making it consume fewer resources.

The results we obtained for this experiment are presented on Table 15. We explain these results with the graphs in Figures 7, 8, 9, 10, and 11.

Scenario	**S1-4**	**S1-5**	**S1-6**	**S1-7**
Time Slots	4	5	6	7
LP Solution	720.0	450.0	432.0	360.0
ILP Solution	780	485	468	392
Final Gap %	8.3	7.8	8.3	8.9
Time to find LP Solution (sec.)	59.5	191.6	273.6	1640.1
Time to find ILP Solution (sec.)	78.7	2034.2	5496.0	15767.3
Starting Configurations	40	40	40	40
Configurations generated	41	116	100	157
Total Configurations	81	156	140	197
Sum of the configuration occurrences (ILP)	195	97	78	56
Number of distinct configurations (ILP)	15	31	13	25
First Gap % Before Gap Reduction	19.4	33.3	38.9	43.9
Configs. Generated before Gap Reduction	35	45	43	56
Max. Memory (MB)	207	952	1739	1178
Max. Swap (MB)	292	1086	1883	1272

Table 15: Results for different numbers of time slots per configuration

Solutions

The first observation we can make about these results is how the solutions become better as we increase the number of time slots, as we can see on Figure 7. By ”becoming better”, we mean that the total amount of time slots needed to transmit data from end to end diminishes as we increase the number of time slots that we can use in a configuration. This happens because if we have a low number of time slots in a configuration and have it send data from end to end, the last time slot will be ”wasted”.

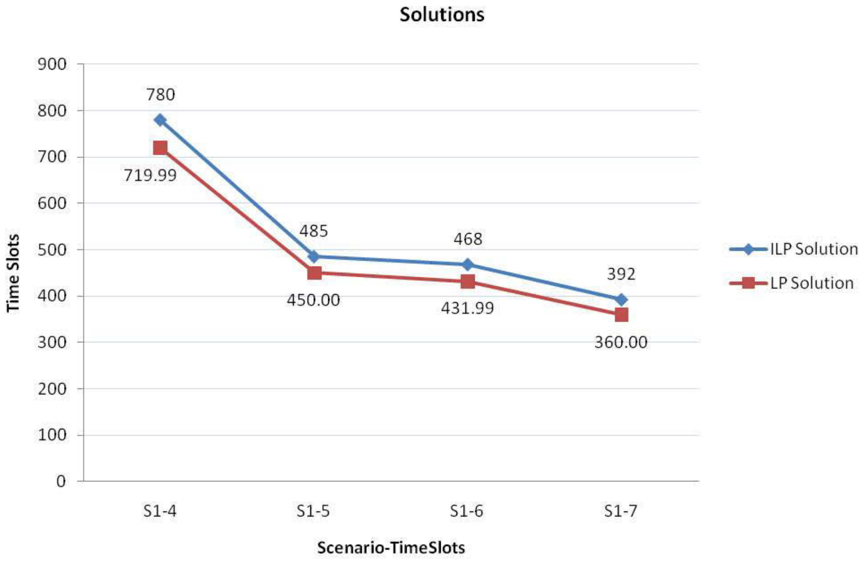


Figure 7: Instance 1. LP and ILP solutions

For example, if we have a configuration that can only use 4 time slots, all the remaining data in intermediate nodes will have to be arriving at its destination during the 4^{th} (and last) time slot. That means that during the 4^{th} time slot, the SSs or the BS will not be transmitting data but only receiving it. If we use configurations that are allowed to use 5 time slots, these configurations may have SSs or the BS transmitting data during the 4^{th} time slot, but not during the 5^{th} one.

This implies that the last time slot is always a slot that is not fully used, because it is only for end nodes to receive data. If we have a small number of time slots per configuration, we will need to use several configurations to transmit data, repeating the last time slot as many times as we use those configurations. Whereas if we used a greater number of time slots per configuration, we would have to use less configurations to transmit the same amount of data and we would therefore have less repetitions of the last time slot, making the transmission faster.

On Figure 7, we can see that the ILP solutions are always greater than or equal to the LP solutions. This happens because the ILP has to have all its variables with integer values, whereas the LP solution does not have this restriction and can have its variables with floating point (real) numbers. Therefore, in the LP solution there might be some configurations that are used 2.5 times for example. In the ILP solution, that same configuration might be used 2 or 3 times only, since only integer values are accepted. This difference between LP and ILP is what helps the LP problem to generally obtain better solutions than the ILP problem.

Gap percentage

Next, we analyze the gap percentage between the objective obtained by the LP and ILP solutions. The gap percentage between the objective of both solutions has to be less than 10% in order to be a guaranteed accurate solution, and it is calculated by using the objective of the optimal LP solution ($\mathrm{OBJ}^*_{\mathrm{LP}}$) and the ILP objective ($\mathrm{OBJ}_{\mathrm{ILP}}$) with the formula $\frac{|\mathrm{OBJ}_{\mathrm{ILP}} - \mathrm{OBJ}^*_{\mathrm{LP}}| \times 100}{\mathrm{OBJ}^*_{\mathrm{LP}}}$. However, when we find the optimal LP solution and solve the ILP for the first time we might get gap percentages higher than 10%, something that we can observe on Figure 8 on the line named "First Gap % before reduction". Therefore, the program will have to iterate and create more solutions until it is able to reduce the gap percentage below 10%.

Notice also that as we increase the number of time slots per configuration, the "First Gap % before reduction" increases as well. A possible interpretation of this increase in the gap percentage might be that it is due to the fact that with the increase of time slots per configuration, we are also increasing the number of decision variables in the program. This will in turn make it more difficult for the ILP to obtain a solution with a small gap percentage, with the configurations that were initially generated.

The solution that we obtain finally has a gap represented by the line "Final Gap %". All the results in this line are below 10%, which was our objective to make the solutions accurate enough.

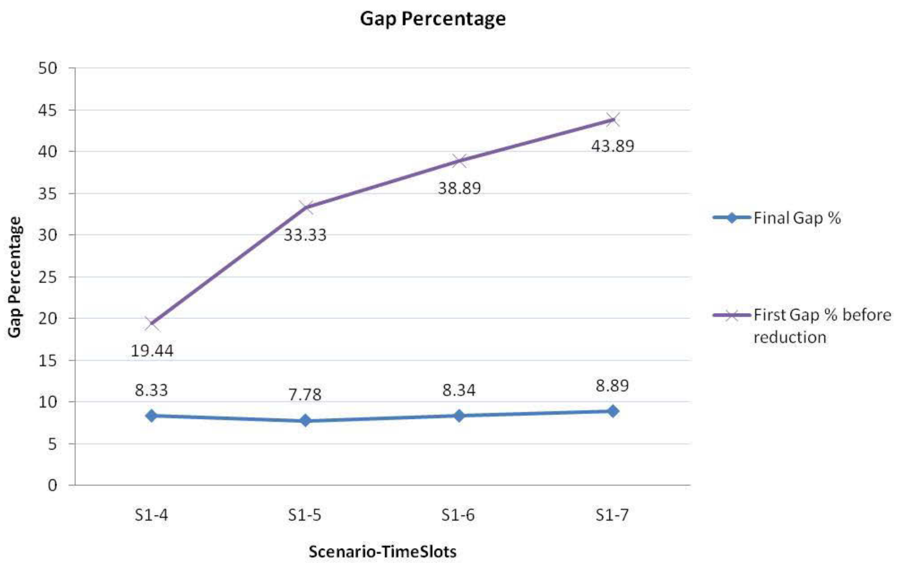


Figure 8: Instance 1. Gap percentage of the solutions

Time

Now we discuss the results based on the running time (in seconds) it takes the simulation to find the LP and ILP solutions. As we can observe on Figure 9, the time to find both the optimal LP and ILP solutions increases exponentially as we increase the number of time slots per configuration. Also, note that the time it takes to find the ILP solution is always higher than the one to find the LP solution. This occurs because we first solve the LP solution, and then we find the ILP solution. If the time it takes to solve the ILP is much higher than the time it takes to find the LP, then we probably have a gap percentage to reduce between the LP and ILP solutions as we can see on Figure 8, which is what delays the program in finding the appropriate ILP solution.

Configurations

In Figure 10, we can see the number of configurations generated, the sum of the configuration occurrences, and the number of distinct configurations utilized in the final ILP solution. The number of configurations generated becomes bigger as the number of time slots per configuration increases. This is probably due to the fact that the gap percentage increases as we increase the number of time slots per configuration. Because of that, the gap has to be reduced by creating new configurations

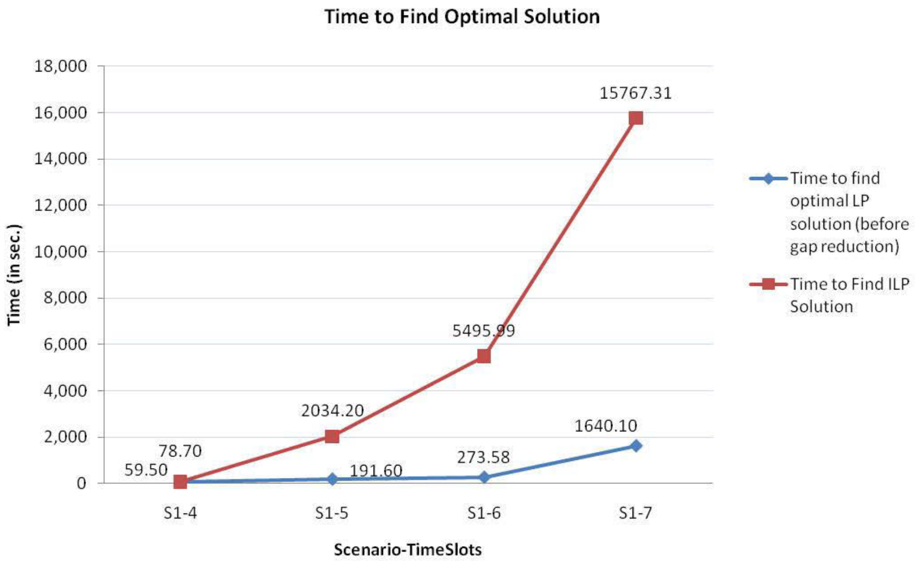


Figure 9: Instance 1. Time to find the optimal LP solution and the ILP solution

which make the number in the ”Configurations Generated” line increase. We can also see how powerful the column generation method is, because the number of configurations generated is much less than the total possible number of configurations that exist (which could be thousands, or even millions).

The sum of the configuration occurrences selected in the ILP final solution is the total number of times configurations were selected, and its value decreases as the number of time slots per configuration increases. This happens because an increasing number of time slots per configuration can transmit the same amount of data in less time.

If we look at the line of the number of distinct configurations, we can see that its values are always much smaller than those of the line of the configurations generated. We should remark that the final solution only selects a small group of the configurations generated for its final ILP solution. For example, in S1-6 we have 100 different configurations generated but only 13 of those were selected in the final ILP solution. This fact shows that of all possible configurations that could exist, we generate only a small number, but even a smaller number of configurations are chosen in the solution.

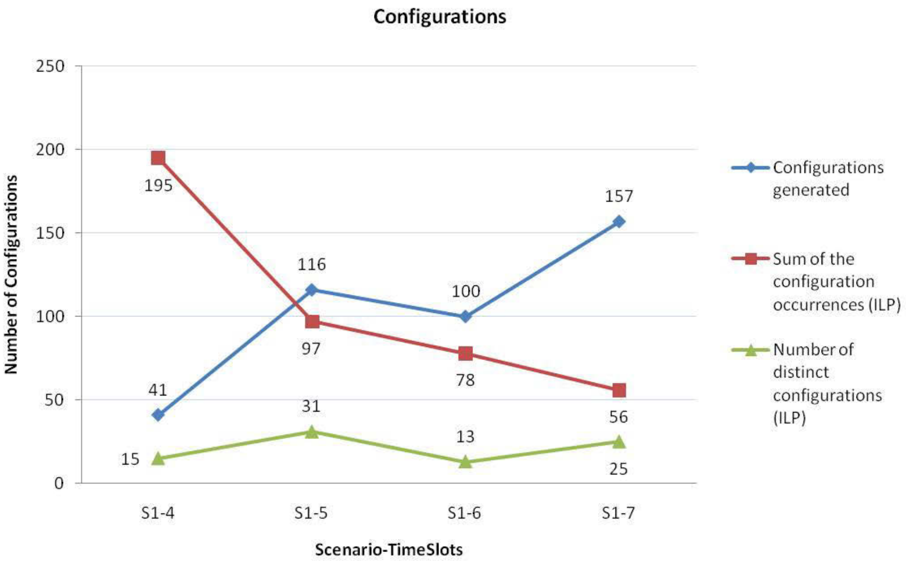


Figure 10: Instance 1. Number of configurations generated and used

Memory

On Figure 11 we can see that the maximum memory usage reached by the program increases as the number of time slots per configuration increases. This happens because of the increase in decision variables as we increase the number of time slots per configuration. The increasing number of decision variables creates a bigger tree of possibilities to choose from, and the simulation has to try these possibilities to find a solution. That is why this increasing tree occupies more memory in the computer that is running the simulation.

However, in S1-7 the memory usage decreases because we use the opportunistic solving method, a solving mode in the IBM ILOG CPLEX software which finds an optimal solution going through different solving paths each time the simulation is run (as opposed to the deterministic method, another mode in CPLEX that finds an optimal solution going always through the same solving path). The fact that the simulation goes through a different solving path every time the program is run made it possible to go through a solution path that used less memory than other possible solution paths.

Note that the two types of memory described in the graph are the physical random access memory (RAM) for the ”Max Memory” line, and the swap memory for the ”Max Swap” line. The

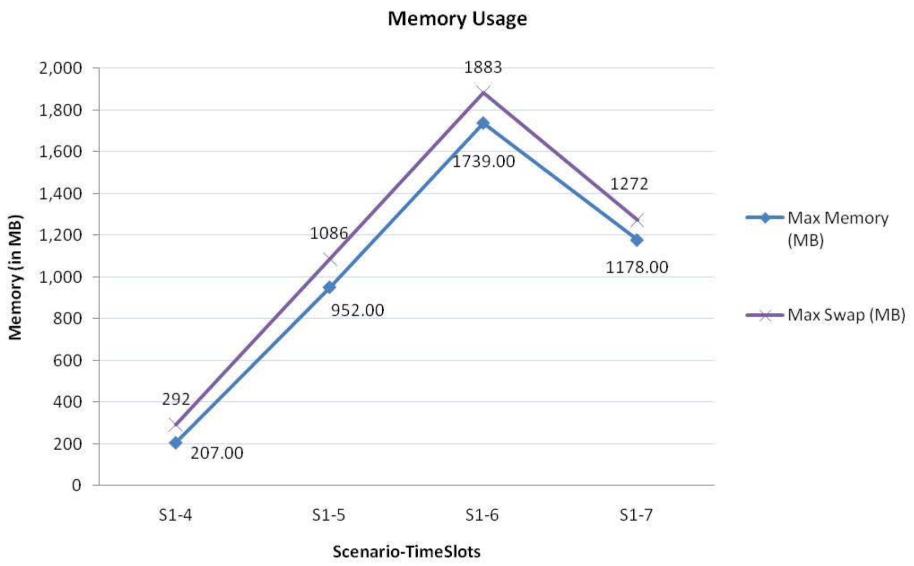


Figure 11: Instance 1. Amount of memory used to simulate each scenario

swap memory is the hard disk space utilized by the system to free space in the RAM memory.

Transmission trees

According to what we discussed earlier in Section 4.4.1, non-overlapping transmission trees are better to transmit information than overlapping transmission trees because they make use of more links than their counterpart (overlapping trees). Our simulation is programmed to choose the tree shape that will help it to reduce transmission time. And, as we predicted, the results of our simulation show that the use of non-overlapping trees is chosen rather than overlapping trees to transmit in less time.

Figure 12 is a rough representation of Figure 6, in which the nodes are located in a similar but not exact geographical position since the purpose of the figure is only to show the transmission trees. We show the transmission trees that resulted from the scenario with 6 time slots per configuration (S1-6), since it is the scenario we use as a template in the following experiments and comparisons. Note that the uplink (UL) links are colored in red (or dark for black and white copies of this thesis), whereas the downlink (DL) links are green (or clear).

If we look carefully at the links, we can see that those that go UL are different than those that

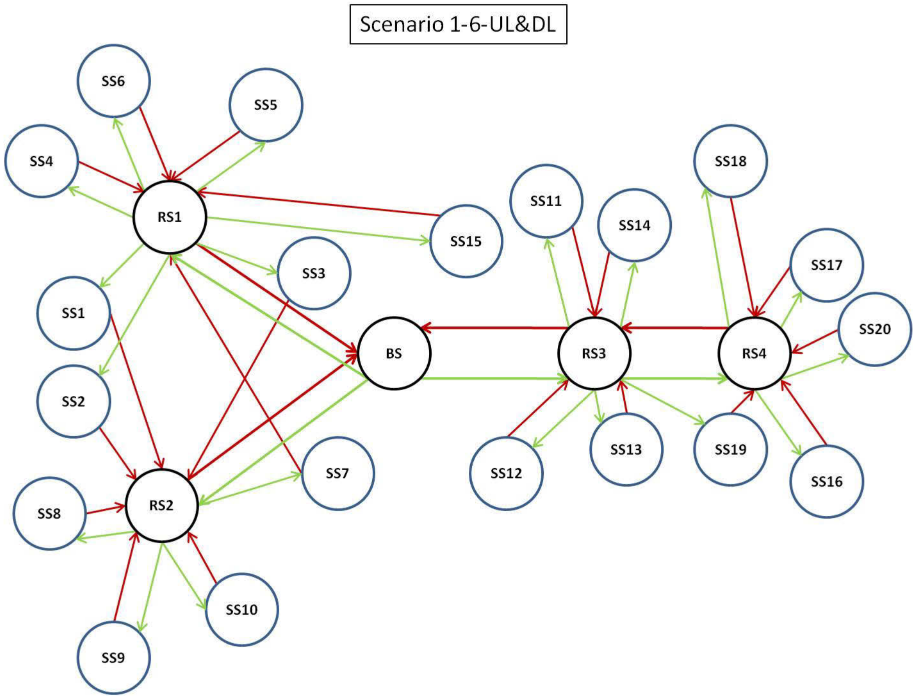


Figure 12: Instance 1, 6 time slots per configuration. Transmission trees.

go DL. In the UL direction, we can see that nodes $SS1$ and $SS2$ are linked to $RS2$, while in the DL direction they receive transmissions directly from $RS1$. Likewise, nodes $SS3$, $SS7$, and $SS19$ utilize different links for UL and DL transmission. We can say that our simulation has chosen the best non-overlapping trees to transmit data in the least amount of time possible.

6.2.2 Instance 2 - Non-Homogeneous Traffic

In Instance 2 we deal with non-homogeneous traffic on the network. By this we mean that nodes do not all transmit and receive the same amount of data as we assumed in the previous experiment. In this case, the nodes that are close to $RS1$ and to $RS3$ will be transmitting and receiving more traffic, whereas the nodes in the proximity of $RS2$ and $RS4$ transmit and receive less traffic.

We compare the different scenarios with S1-6 (scenario instance 1 with 6 time slots) from the previous section. The scenarios that we created for this experiment have always the same nodes that are transmitting more or less amount of traffic. Nevertheless, the variation from one scenario to the other is the increase of the difference in traffic between those nodes that transmit more data and those that transmit less. For example, in scenario S2c the difference between the nodes that transmit more data and those that transmit less is greater than when we compare it to scenario S2b. Therefore, the difference increases in this order starting with S1-6 (no difference in traffic), S2a, S2b, and S2c. As the difference in transmission increases, so does the weight on each node that has to transmit the data of a SS with more traffic.

Having explained the difference between each scenario, we now have to say that the data that is more important to analyze in this experiment is how many links connected to a SS does a RS serve and for what percentage of the total traffic in the network. That is why for scenarios S1-6, S2a, S2b, and S2c we count the active links connected to each RS on Table 16 and see if the traffic is shared by the relay stations. We also look at Figure 13, which shows the UL and DL trees for scenario S2c, as well as which nodes are transmitting more and less traffic.

Scenario	**S1-6**		**S2a**		**S2b**		**S2c**	
	Links	Traffic %	Links	Traffic %	Links	Traffic %	Links	Traffic %
RS1	12	30.00	10	26.00	10	27.13	10	29.00
RS2	10	25.00	12	29.25	12	28.37	12	27.00
RS3	9	22.50	9	23.13	7	22.75	9	23.75
RS4	9	22.50	9	21.62	11	21.75	9	20.75

Table 16: Instance 2. Links between RS and SS for UL and DL

As we can see on Table 16, as we increase the amount of data transmitted from the nodes close to $RS1$, the number of links that are connected to it diminish, while those connected to $RS2$ increase. In addition, we can see how the total traffic percentage of these two nodes oscillates between values of 25% and 30%. This data indicates that $RS1$ and $RS2$ are sharing their traffic weight so that transmission speed will not decay.

However, when looking at the links connected to $RS3$ and $RS4$, we can see that they both have 9 active links. If we observe Figure 13, we will be able to see that the reason for these two RS to have the same number of links even though the nodes closer to $RS3$ are those with more traffic, is that the SSs that have more traffic and are close to $RS4$ will connect to it to relieve $RS3$. To balance this, the SSs that have less traffic to transmit and are close to both RSs will connect to $RS3$ so that $RS4$ does not have to carry all the traffic weight. We can also see in the traffic percentage numbers how both $RS3$ and $RS4$ transport similar amounts of data.

Note that the percentage of data transmitted by $RS3$ on Table 16 does not include the total data that it will transmit, since it acts as an intermediary node between the BS and $RS4$ and therefore has to transport the percentage of data that belongs to $RS4$ as well. However, our model also takes that factor into account at the time of distributing the load that each RS has to transmit.

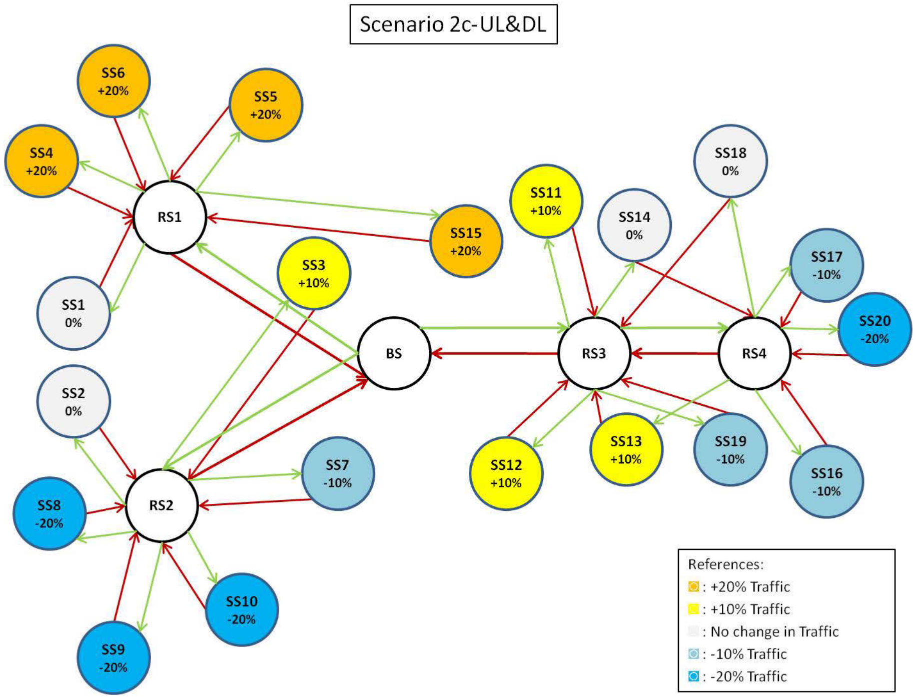


Figure 13: Scenario 2c. Transmission trees.

The conclusion at which we can arrive with these observations is that our model is effective at balancing transmissions so that the RSs in an area of the network (such as neighbors $RS1$ and $RS2$, and neighbors $RS3$ and $RS4$) share the weight of transmitting traffic with their neighboring RSs. This balance of traffic is due to the objective of our model, which is to reduce the amount of transmission time on the network.

6.2.3 Instance 3 - Unbalanced Traffic

In this instance we made experiments having unbalanced traffic. By unbalanced traffic we mean that we moved the SSs in the network to concentrate them in a certain area. We took SSs that were close to $RS3$ and $RS4$, as well as some nodes that were only able to connect to $RS2$, and moved these nodes in between $RS1$ and $RS2$. This situation simulates in a way when a group of people concentrate in a certain location for an event that takes place. In this case, the big event is located between nodes $RS1$ and $RS2$.

We use once again scenario S1-6 as the template to compare scenarios S3a, S3b, and S3c which we created for this experiment. Scenario S3a moves 3 nodes between $RS1$ and $RS2$, while scenario S3b is the same as S3a but with 2 more nodes in the concentration area. Scenario S3c is almost equal to S2b, except that it adds 2 more nodes to the area of the big event.

Analyzing the data obtained on Table 17, we can see that as the number of nodes in the concentration area between $RS1$ and $RS2$ increases, so do the number of links connected to $RS1$ and $RS2$ and the traffic percentage transported by these two RSs. However, note that in S1-6 the percentage of traffic that goes through $RS1$ is more than that of $RS2$, but in S3a, S3b, and S3c the traffic percentage is even for both RSs. This is an indication that our model is distributing data among the RSs to avoid delays in transmission as much as possible.

Scenario	**S1-6**		**S3a**		**S3b**		**S3c**	
	Links	Traffic %	Links	Traffic %	Links	Traffic %	Links	Traffic %
RS1	12	30.0	12	30.0	14	35.0	14	35.0
RS2	10	25.0	12	30.0	14	35.0	14	35.0
RS3	9	22.5	8	20.0	6	15.0	7	17.5
RS4	9	22.5	8	20.0	6	15.0	5	12.5

Table 17: Instance 3. Links between RS and SS for UL and DL

In addition, if we observe the numbers on Table 17 for S3b and S3c, we can see that the number of links as well as the traffic percentage for $RS1$ and $RS2$ does not increase. If we look at Figure 14, we can see that this is due to the fact that the nodes that were approximated to the event area in S3c ($SS11$ and $SS16$) were close enough to $RS3$ to use it as their relay to connect to the BS. That way, our model avoided saturating $RS1$ and $RS2$, and took advantage of the proximity of $RS3$ that was not using its full capacity.

Now, if we look at Figure 15 (the line named "ILP Solution"), we can notice that comparing S1-6, S3a, S3b, and S3c we obtain the best transmission time in S3a (438 time slots). Considering this and seeing the results on Table 17, we could think that in S1-6, $RS3$ is saturated by the amount of traffic it has to transport; but since in S3a $RS3$ is relieved from some traffic, the total transmission

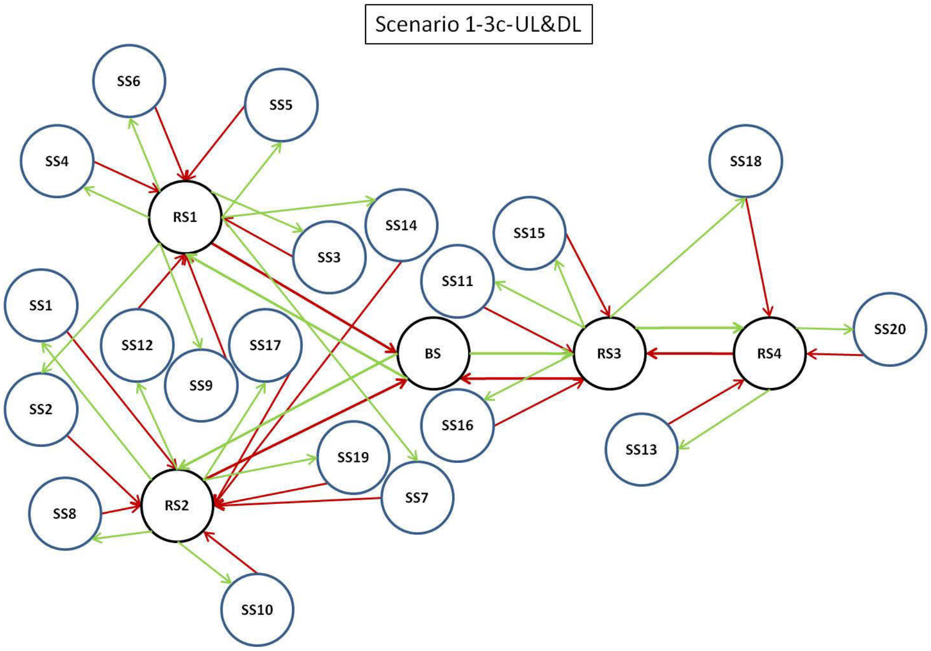


Figure 14: Scenario 3c. Transmission trees.

time in the network is reduced. We could also suppose that $RS1$ and $RS2$ get saturated only with a percentage higher than 30% of total network traffic (as it occurs in S3b and S3c), delaying the transmission of data on the network.

Once more, we could see that our model can deal perfectly with traffic changes on the network. In this case, it was able to distribute unbalanced traffic among the RSs, always with the objective of reducing the amount of total transmission time on the network.

6.2.4 Instance 4 - Change in RS Buffer Sizes

We have also performed experiments changing the buffer size of the RSs. The buffer size we use in S1-6 is 4,000 bits; while we have a transmission rate of 4,000 bits per time slot for links between the BS and a RS, or between two RSs, and a rate of 800 bits per time slot for links between a RS and a SS.

Taking the mentioned data into account, we increased the size of the buffers over 4,000 bits. As we can see in Figure 16 increasing the buffer size did not reduce or increase the number of time slots it takes to transmit data on the network (the ”LP Solution” line marked always the same

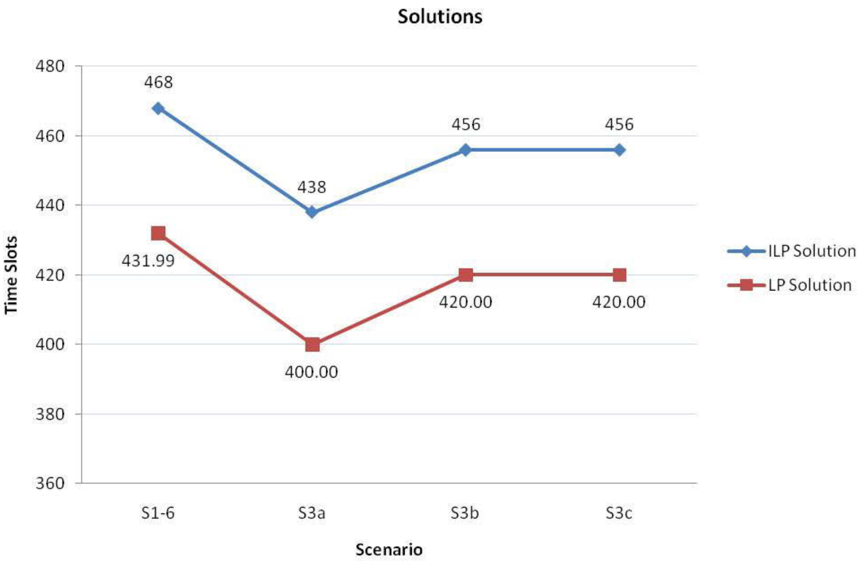


Figure 15: Instance 3. LP and ILP solutions

number of time slots). Nevertheless, when we decreased the buffer size to a number smaller than the maximum rate in the network (4,000 bits per time slot), we obtained results showing an increase in transmission time. The cause of this increase was that the network could no longer transmit at its maximum capacity through all its links, meaning that the maximum rate of its links connecting the BS to the RS (and the RSs amongst themselves) could never be reached.

6.2.5 Comparison Between Model I and Model II

As we mentioned in previous chapters, the model we developed (Model II) performs scheduling within a configuration because we can clearly see how data flows through each node during the period that the configuration lasts. However, our model chooses those configurations and performs them in no certain order, which is not exactly what the definition of scheduling states, since we should know in which order those configurations should go. This problem could be solved with a heuristic that gives an order in time to the use of configurations, performing scheduling and addressing delay constraints at the same time.

Model I on the other hand, does not perform any scheduling since it creates configurations on a time slot by time slot basis, and then chooses which configurations to use without having any

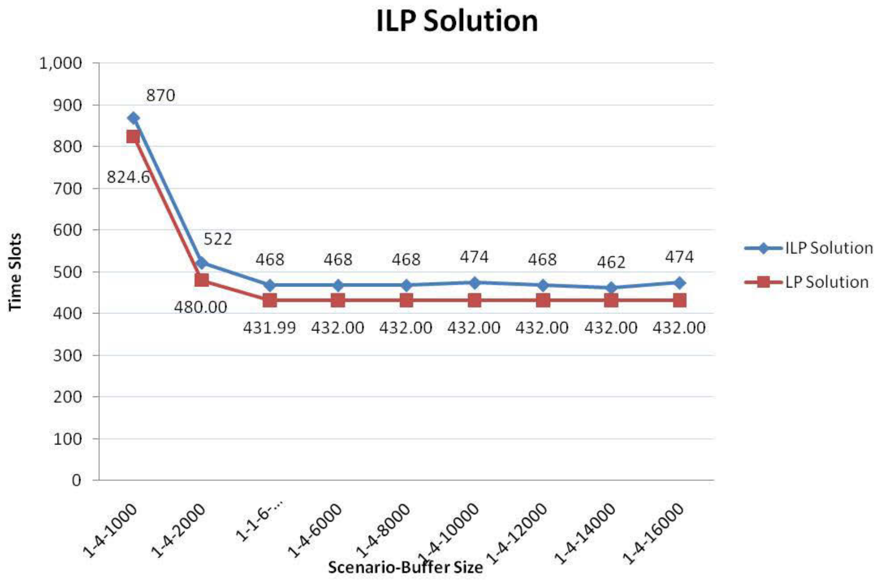


Figure 16: Instance 4. LP and ILP solutions

defined order. Model I also selects configurations that can transmit data even if the node does not contain data to send (for example, an intermediate RS node transmitting at the beginning of the simulation without having any data to transmit).

In this section, we compare the results obtained by these two models and see their differences in experiment results. First, we compare the two models as they transmit the same amount of data. Next, we compare them on a configuration by configuration basis, looking at smaller examples of scheduling.

Transmission of the same amount of data

We now compare our model using once again scenario S1-6 as the template, and see the results obtained by running a simulation with Model I with the same inputs that were used for our model. The results are shown on Table 18.

In Table 18 we can see that the LP and ILP solutions are much better in Model I, due to the fact that we cannot analyze many time slots per configuration in our model. However, if we were able to have more time slots per configuration, this difference in solutions between Model I and Model II would be smaller.

Scenario	**S1-6**	**Model I**
Time Slots per Configuration	6	1
LP Solution	432.0	261.8
ILP Solution	468	283
Final Gap %	8.3	8.1
Time to find LP Solution (sec.)	273.6	6.2
Time to find ILP Solution (sec.)	5496.0	23.4
Max. Memory (MB)	1739	4
Max. Swap (MB)	1883	29

Table 18: Results for S1-6 and Model I

Nevertheless, to have more time slots per configuration in our model would not be feasible yet, since Model II is not scalable and cannot be properly run (with more than 7 time slots per configuration) by the computers we currently have access to. A very good example of the non-scalability of our model can be seen when we look at the time it takes to find the LP and ILP solutions. Model II takes much longer time to run than Model I does. In addition, if we look at the maximum RAM memory and maximum swap memory used, our model requires many more resources than Model I. Even though our model is not scalable, we could improve it by adding heuristics to speed up the work it performs.

Comparing Configurations

To compare Model I and Model II in an equilibrated way, we have to compare the results of their simulation after knowing the maximum number of time slots we can use in Model II (our model). Therefore, to do this comparison we have to make it at the level of configurations in which we already know how many time slots per configuration our model was able to handle to obtain the optimal solution.

To make this comparison, we use the results obtained by simulating our model with configurations of a maximum size of 6 time slots. After obtaining the results of scenario S1-6, we saved the information of the configurations selected by the ILP, creating an input for Model I. This input contained the number of packets transmitted by each node in the configuration, as well as the active transmission links. Each active transmission link was transformed into a starting configuration for Model I.

We can see on Figure 17 the results obtained from simulating Model I in a configuration by configuration basis. The line named "Model II" is the amount of time slots used by each configuration according to the results obtained from S1-6. That line always has the value 6, since all these configurations utilized 6 time slots to transmit data.

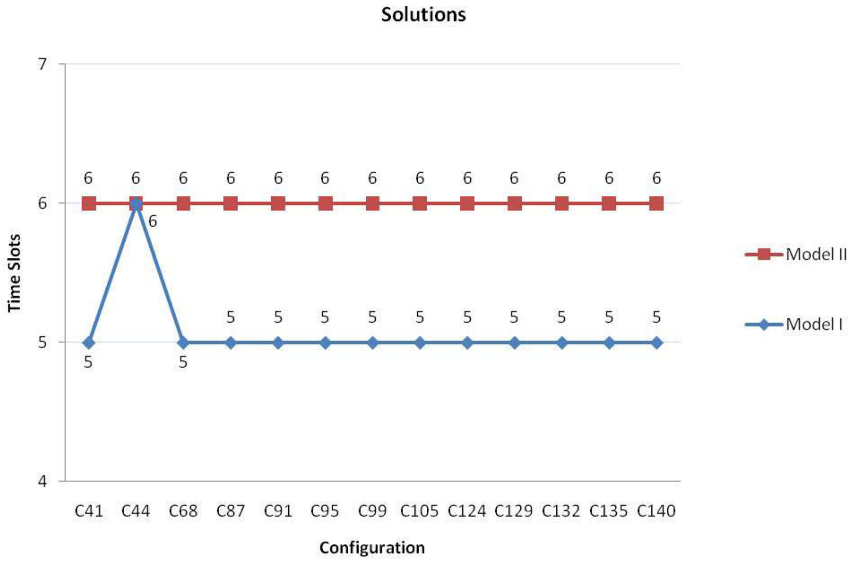


Figure 17: Model I. Configuration solutions

On the line called "Model I" we can see the results obtained by Model I. All the values are 5, except for the value of configuration C44 which is 6. The reason why these values are 5 is because the configurations used by the ILP solution take 5 time slots to transmit information. However, none of these configurations considers that there should be a "starting configuration", in which only origin nodes that contain data can submit, without having any intermediate nodes transmitting (since they do not contain any data in the beginning). This difference in selecting configurations is what makes Model I have better solutions than Model II.

As we could see, the difference in time slots between Model I and Model II is almost always one time slot less for Model I. It would be interesting to know how much the difference would be between these two models if we could use configurations with more time slots in our model. Nevertheless, at this point in time and with the current model we have, it would be impossible for us to find a solution for a higher number of time slots due to the memory and time requirements to run the simulation.

Considering the non-scalability of Model II, we could design a heuristic to create a so called ”warm start” solution to feed our model. This solution should be good enough to help our model to arrive at a near optimal solution much faster. And as we mentioned earlier, we could implement another heuristic to order in time the use of configurations output by our model, helping us to accomplish scheduling and to address delay constraints. If these heuristics were not enough to speed up our model, we could develop another heuristic to allow configurations with a higher number of time slots to be compared with Model I.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

There were two objectives in our work. The first objective was to design a routing and scheduling optimization model for a WiMAX IEEE 802.16j network utilizing the column generation technique and to perform scheduling as defined in Section 3.1. The second objective was to compare Model II (which we designed) to Model I (which does not exactly fit the proper definition of scheduling) and see how different the results obtained by both models were.

We accomplished partly our first objective in the sense that we performed scheduling within each configuration, but we did not provide a way to state in which order the selected configurations should be used; an issue that could be addressed by a heuristic that would give an order in time to the use of configurations (and provide delay constraints as well). Nevertheless, we were able to obtain useful results in our experiments showing how using different routing trees for uplink and downlink would be more efficient compared to using the same tree for both directions, and how using configurations with a greater number of time slots would help to reduce transmission time. We also showed how our model responded to non-homogeneous and unbalanced traffic by routing traffic through the different RSs in the network, and how using different buffer sizes would only affect transmission times if the buffer size were less than the maximum transmission rate in the network.

Our second objective was partly accomplished as well. We compared the results obtained after simulating Models I and II. By comparing these two models and their results, on how many configurations and time slots each one used to transmit data from end to end, we could see that Model I used a considerably lower number of time slots than Model II, in which several time slots were wasted because of having configurations with a small number of time slots.

This measurement did not seem fair because of the limitation of the configurations in Model II to a small number of time slots. That is why we decided to obtain the configurations output by the optimal solution of Model II and run each one individually on Model I. This way, we were able to see on a configuration-by-configuration basis how Model I at almost all times utilized one less time slot than Model II because of not performing proper scheduling. Nevertheless, this comparison would be more interesting if it would be performed in configurations of 50 or more time slots, something that is impossible at this time with our model and the computational resources we have access to.

To make our model fully compatible with the scheduling definition, we should add constraints that would give an order in time in which each configuration should be used. Or we could perform scheduling in two phases, by first using our model to obtain the configurations to be selected, and then including a heuristic that would determine how configurations should be ordered in time.

In addition, to compare our model to Model I on a more similar ground, we should find a way to create configurations with more time slots, without needing more computational resources than those that we currently have access to. To make this possible, we could utilize the power and speed of a heuristic method, even though using heuristics does not provide a known optimal solution. Otherwise, we could utilize both a column generation model and a heuristic, and have them solve this problem.

In conclusion, the model we developed works as a planning tool and performs scheduling within a configuration, but the way it is modeled makes it very ”heavy” to solve due to the growing number of decision variables it contains as we increase the number of time slots per configuration. The direction to take would probably be to combine the use of the column generation method with a heuristic, since using only our model with column generation will not be efficient for scheduling during several time slots.

7.2 Future Work

Our future work in the scheduling area for WiMAX would still consist of making our models and simulations adjust to the scheduling definition cited in Section 3.1. As a continuation to our model, to make it be completely a scheduling model we should add constraints in the master problem ordering the use of configurations in time. This could give place to adding delay constraints as well. However, if adding more constraints to our scheduling model will make it run slower and consume more memory, we should utilize another strategy to schedule traffic.

Another idea, as mentioned in the conclusion, would be to simulate scheduling in an IEEE 802.16j network utilizing a heuristic combined with the column generation method, or a heuristic by itself. Even though we will not have complete certainty that the solutions we obtain are optimal (or near optimal), we will be able to perform scheduling and compare the output to Model I. With a heuristic we would also be able to add delay constraints and perhaps different types of traffic to the network, making scheduling more realistic.

Finally, we could use simulation packages such as NS2 or OPNET to simulate scheduling in WiMAX. The advantage of these simulators is that they can be very accurate in their results, and perhaps be faster and more efficient than using an optimization model. Such a simulation tool would also enable us to compare our results to models (such as Model I) where scheduling is not performed according to the definition.

Bibliography

[ACH10] Z. Abichar, J.M. Chang, and C.Y. Hsu. WiMAX or LTE: Who will Lead the Broadband Mobile Internet? *IT Professional*, 12(3):26–32, 2010.

[AI08] S. Ahson and M. Ilyas. *WiMAX: applications.* CRC, 2008.

[Amo01] K. Amouris. Space-time division multiple access (stdma) and coordinated, power-aware maca for mobile ad hoc networks. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2890–2895. IEEE, 2001.

[BBT$^+$07] L. Badia, A. Baiocchi, A. Todini, S. Merlin, S. Pupolin, A. Zanella, and M. Zorzi. On the impact of physical layer awareness on scheduling and resource allocation in broadband multicellular IEEE 802.16 systems [Radio Resource Management and Protocol Engineering for IEEE 802.16]. *Wireless Communications, IEEE*, 14(1):36–43, 2007.

[CC06] A. Capone and G. Carello. Scheduling optimization in wireless mesh networks with power control and rate adaptation. In *IEEE SECON 2006*, pages 138–147, 2006.

[CCF$^+$10] A. Capone, G. Carello, I. Filippini, S. Gualandi, and F. Malucelli. Routing, scheduling and channel assignment in Wireless Mesh Networks: Optimization models and algorithms. *Ad Hoc Networks*, 8(6):545–563, 2010.

[CdM08] K.C. Chen and J.R.B. de Marca. *Mobile WiMAX.* Wiley Online Library, 2008.

[CFM08] A. Capone, I. Filippini, and F. Martignon. Joint routing and scheduling optimization in wireless mesh networks with directional antennas. In *IEEE International Conference on Communications - ICC*, pages 2951–2957, 2008.

[CGN+09] V. Corvino, L. Giupponi, A. Perez Neira, V. Tralli, and R. Verdone. Cross-Layer Radio Resource Allocation: The Journey so Far and the Road Ahead. *Second International Workshop on Cross Layer Design, 2009. IWCLD '09.*, pages 1–6, June 2009.

[Chv83] V. Chvatal. *Linear Programming.* Freeman, 1983.

[CLY07] Y. Cao, Z. Liu, and Y. Yang. A centralized scheduling algorithm based on multipath routing in WiMAX mesh network. In *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006. International Conference on*, pages 1–4. IEEE, 2007.

[Con10] J.P. Conti. LTE vs WiMax: the battle continues [COMMS WiMax vs LTE]. *Engineering & Technology*, 5(14):63–65, 2010.

[CRH+08] M. Cao, V. Raghunathan, S. Hanly, V. Sharma, and PR Kumar. Power control and transmission scheduling for network utility maximization in wireless networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5215–5221. IEEE, 2008.

[CRK06] M. Cao, V. Raghunathan, and P.R. Kumar. A tractable algorithm for fair and efficient uplink scheduling of multi-hop WiMax mesh networks. In *2nd IEEE Workshop on Wireless Mesh Networks - WiMesh*, pages 101–108, 2006.

[CTV09] V. Corvino, V. Tralli, and R. Verdone. Cross-layer radio resource allocation for multicarrier air interfaces in multicell multiuser environments. *Vehicular Technology, IEEE Transactions on*, 58(4):1864–1875, 2009.

[Duf10] J. Duffy. LTE vs WiMax: Has mobile WiMAX been permanently crippled in the 4G technology battle? Network World, June 7 2010. http://www.networkworld.com/news/2010/060710-tech-argument-lte-wimax.html.

[EN11] J. El-Najjar. *Efficient design of WiMAX/802.16 mesh networks.* PhD thesis, CONCORDIA UNIVERSITY, 2011.

[ENAJ09] J. El-Najjar, C. Assi, and B. Jaumard. Joint routing and scheduling in WiMAX-based mesh networks: A column generation approach. In *10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks - WoWMoM*, 2009.

[ENAJ10] J. El-Najjar, C. Assi, and B. Jaumard. Joint routing and scheduling in WiMAX-based mesh networks. *Wireless Communications, IEEE Transactions on*, 9(7):2371–2381, 2010.

[FLH10] L. Fu, S.C. Liew, and J. Huang. Fast algorithms for joint power control and scheduling in wireless networks. *Wireless Communications, IEEE Transactions on*, 9(3):1186–1197, 2010.

[For10a] WiMAX Forum. About the WiMAX Forum. Online, July 28 2010. http://www.wimaxforum.org/about.

[For10b] WiMAX Forum. WiMAX Maps - WiMAX Deployments. Online, October 6 2010. http://www.wimaxmaps.org/.

[GG61] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, 1961.

[GG63] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem: Part ii. *Operations Research*, 11:863–888, 1963.

[GGM09] D. Ghosh, A. Gupta, and P. Mohapatra. Adaptive Scheduling of Prioritized Traffic in IEEE 802.16 j Wireless Networks. In *2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 307–313. IEEE, 2009.

[HP09] C.Y. Hong and A.C. Pang. 3-approximation algorithm for joint routing and link scheduling in wireless relay networks. *Wireless Communications, IEEE Transactions on*, 8(2):856–861, 2009.

[JX05] M. Johansson and L. Xiao. Scheduling, routing and power allocation for fairness in wireless networks. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 3, pages 1355–1360. IEEE, 2005.

[KSK10] K.R. Krishnan, D. Shallcross, and L. Kant. Joint optimization of scheduling and multicast trees by column-generation. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, pages 578–583. IEEE, 2010.

[KWE08] S. Kompella, J.E. Wieselthier, and A. Ephremides. Multi-hop routing and scheduling in wireless networks subject to sinr constraints. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5690–5695. IEEE, 2008.

[KWES08] S. Kompella, J.E. Wieselthier, A. Ephremides, and H.D. Sherali. A cross-layer approach to end-to-end routing and SINR-based scheduling in multi-hop wireless networks. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pages 261–266. IEEE, 2008.

[LKD10] W.H. Liao, S.P. Kedia, and A.K. Dubey. A centralized scheduling algorithm for WiMAX mesh network. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 858–861. IEEE, 2010.

[LO09] S.C. Lo and L.C. Ou. Efficient Algorithms for Routing and Centralized Scheduling for IEEE 802.16 Mesh Networks. In *Scalable Computing and Communications; Eighth International Conference on Embedded Computing, 2009. SCALCOM-EMBEDDEDCOM'09. International Conference on*, pages 212–217. IEEE, 2009.

[MNP$^+$06] R. Marks, M. Nohara, J. Puthenkulam, M. Hart, and I. Fu. IEEE 802 Tutorial: 802.16 Mobile Multihop Relay (presentation at IEEE 802.16 Session 42). Unpublished, March 2006. Collection of four presentations. http://www.ieee802.org/16/tutorial/index.html.

[MPR08] C. Molle, F. Peix, and H. Rivano. An optimization framework for the joint routing and scheduling in wireless mesh networks. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–5. IEEE, 2008.

[Ned02] P. Nedeltchev. *Troubleshooting remote access networks.* 2002.

[Nua07] L. Nuaymi. *WiMAX Technology for Broadband Wireless Access.* Wiley, 2007.

[NW88] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, New York, 1988.

[OZV08] M. Okuda, C. Zhu, and D. Viorel. Multihop Relay Extension for WiMAX NetworksOverview and Benefits of IEEE 802.16 j Standard. *Fujitsu Sci. Tech. J*, 44(3):292–302, 2008.

[Par06] D. Pareek. *WiMAX: Taking wireless to the max.* CRC Press, 2006.

[PH09] S.W. Peters and R.W Heath. The Future of WiMAX: Multihop Relaying with IEEE 802.16j. *IEEE Communications Magazine*, 47:104–111, January 2009.

[SIJT09] C. So-In, R. Jain, and A.K. Tamimi. Scheduling in IEEE 802.16e Mobile WiMAX Networks: Key Issues and a Survey. *IEEE Journal on Selected Areas in Communications*, 27:156–171, February 2009.

[SKK$^+$10] S.O. Seo, S.J. Kim, S.Y. Kim, Y.I. Kim, H.W. Lee, S. Ryu, and C.H. Cho. Relay Performance Analysis of TTR and STR Relay Modes in IEEE 802.16 j MMR System. *ETRI journal*, 32(2), 2010.

[Sta10] The IEEE 802.16 Working Group On Broadband Wireless Access Standards. IEEE Wireless MAN 802.16. Online, July 28 2010. http://www.ieee802.org/16/.

[StIMTS09a] IEEE Computer Society, the IEEE Microwave Theory, and Techniques Society. IEEE Std 802.16-2009, IEEE Standard for Local and metropolitan area networks. Part 16: Air Interface for Broadband Wireless Access Systems. May 2009.

[StIMTS09b] IEEE Computer Society, the IEEE Microwave Theory, and Techniques Society. IEEE Std 802.16j-2009, IEEE Standard for Local and metropolitan area networks. Part 16: Air Interface for Broadband Wireless Access Systems. Amendment 1: Multiple Relay Specification. May 2009.

[Tan03] A.S. Tanenbaum. *Computer Networks - Fourth Edition*. Prentice Hall, 2003.

[Van94] F. Vanderbeck. *Decomposition and Column Generation for Integer Programs*. PhD thesis, Universite Catholique de Louvain, 1994.

[WIS07] WISELAB. WiMAX Activity in Canada. Technical report, Communications Research Centre (CRC) Canada, 3701 Carling, Ottawa, Canada, February 2007.

[YAS10] M. Yazdanpanah, C. Assi, and Y. Shayan. Optimal joint routing and scheduling in wireless mesh networks with smart antennas. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pages 1–7. IEEE, 2010.

[YHXM09] Y. Yang, H. Hu, J. Xu, and G. Mao. Relay Technologies for WiMAX and LTE-Advanced Mobile Systems. *IEEE Communications Magazine*, pages 100–105, October 2009.

[YW08] K. Yang and X. Wang. Cross-layer network planning for multi-radio multi-channel cognitive wireless networks. *Communications, IEEE Transactions on*, 56(10):1705–1714, 2008.

[ZWZL06] J. Zhang, H. Wu, Q. Zhang, and B. Li. Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, pages 631–640. IEEE, 2006.