Adaptive load balancing routing algorithms for the next generation wireless telecommunications networks.

A thesis submitted for the degree of
Doctor of Philosophy

by
Panagiotis Tsiakas

School of Engineering and Design,
Brunel University

September 2009

Abstract

With the rapid development of wireless networks, mesh networks are evolving as a new important technology, presenting a high research and commercial interest. Additionally, wireless mesh networks have a wide variety of applications, offering the ability to provide network access in both rural and urban areas with low cost of maintenance.

One of the main functionalities of a wireless mesh network is load-balancing routing, which is the procedure of finding the best, according to some criteria, routes that data need to follow to transfer from one node to another. Routing is one of the state-of-the-art areas of research because the current algorithms and protocols are not efficient and effective due to the diversity of the characteristics of these networks.

In this thesis, two new routing algorithms have been developed for No Intra-Cell Interference (NICI) and Limited Intra-Cell Interference (LICI) networks based on WiMAX, the most advanced wireless technology ready for deployment. The algorithms created are based on the classical Dijkstra and Ford-Fulkerson algorithms and can be implemented in the cases of unicast and multicast transmission respectively.

TOC

List of tables

List of figures

Acknowledgments

There are a number of people I would like to thank for their support and help in this research. First of all, my project’s supervisors, Dr. Marios Hadjinicolaou from Brunel University and Dr. Konstantinos Voudouris from the Technological Educational Institution of Athens, offered their knowledge and excellent guidance in the course of this thesis. Secondly, Professor Dimos Triantis, my State Scholarship foundation supervisor, provided me with his valuable consulting for the completion of my studies. Finally, I would like to thank the consortium members of the FP7-REWIND project and especially Dr. Avner Dor for their full and continuous collaboration. Last but not least, I want to express my full gratitude to my family for their psychological and sentimental support and, especially to my lovely wife, Vicky, for her endless and tireless support and encouragement throughout these years.

This research has received funding from the State Scholarships Foundation of Greece.

1 Introduction

1.1 Overview

Wireless communications nowadays are considered to be a “hot” topic in the field of Information and Communication Technologies (ICT). Both the introduction of innovative and demanding services and the exponential growth in the volume of numerous handheld devices, such as laptops, PDAs etc, have now increased the need for ubiquitous connectivity and coverage. Wireless technology has the potential to be an important component of future converged (or ubiquitous) networks because of its range and the relatively high-speed connectivity and service availability [European Commission, 2007].

The simplicity of wireless network deployment, especially after hot-spot exploitation, has led to the existence of millions of Wi-Fi networks on the planet, many of which are connected in a mesh topology [Held, 2005 Hossain, 2008]. Nevertheless, Wi-Fi as a technology has specific drawbacks such as limited range, power demand and interference from other wireless devices. Thus, the needs for more reliable wireless broadband technology for Internet access have grown to a great extent [Ohrtman, 2003].

The new technology introduced to meet these needs is called “WiMAX (Worldwide Interoperability for Microwave Access)”. WiMAX is currently the most advanced wireless technology available for deployment, and many of its aspects are likely to be implemented in any 4G wireless technology. WiMAX with the IEEE 802.16-2004 or 802.16e standard, which includes support for optional mesh topology, enables the creation of mesh networks. It allows the deployments of nodes distributed in a network in an arbitrary manner, operating either on licensed or unlicensed bands, with built-in Quality of Service (QOS) support. In addition, it is optimized for longer distance and higher data rates than Wi-Fi is. The latest release of the WiMAX standard, 802.16j-2009, introduces the concept of wireless Multi-hop Relay Stations (MRSs), which should be small, cost effective and easy to install in order to enable mass deployment in indoor and outdoor environments. Additionally, MRSs create relatively small areas with excellent coverage and high capacity availability [Senza Fili Consulting, 2007, Chochliouros *et al.*, 2009b].

MRSs can become for WiMAX what hot spots have been for Wi-Fi technology [Agapiou, 2009]. The low cost and ease of installation of relays can lead to an exponential growth in the number of nodes in WiMAX networks. However, being able to exploit fully the potentials of a mesh network, the furtherance of significant research activities is required. The major limitation in the existing routing protocols regarding throughput is that, whenever the population of nodes grows or the number of hops increases, a major reduction is introduced [Kyungtae & Hong, 2006]. Hence, the development of new, fast and efficient load balancing algorithms is essential.

The main functionality of load balancing is routing of data. In this thesis, two main routing algorithms have been developed for two different cases and the mathematical models have been thoroughly presented. The first algorithm based on Dijkstra [1959] focuses on unicast transmission in No Intra-Cell Interference (NICI) networks, while it has been extended to support also unicast transmission in Limited Intra-Cell Interference (LICI) networks. The second algorithm based on the Ford-Fulkerson [1956] focuses on multicast transmission in Limited Intra-Cell Interference (LICI) networks.

However, simulation results for the developed algorithms could not be provided. Current versions of simulators support neither the IEEE 802.16j standard nor the concept of relay stations, while the notion of extending the WiMAX mesh network architecture to include relay stations is also not supported by the current standards yet.

1.2 Motivation

The IEEE 802.16™ standard and the WiMAX system profiles provide only the outline and the requirements for functionalities that should be supported, but allow the implementation of algorithms to be developed by each vendor, without any restraints. There are many research groups in industry and in academia working on these issues, while much of this effort is aimed at developing load balancing schemes and especially routing algorithms. WiMAX was the stimulating use case and gave the instigation for this study as well.

The notion of WiMAX mesh networks using relay stations as nodes is pioneering and is considered as a state-of-the-art topic in wireless networks. Thus, there hasn’t been any work published towards this direction yet, since the IEEE 802.16j standard for Relay Stations has just been released. Relay stations are based on a highly integrated System on Chip (SoC) device, which incorporates all baseband, networking and control processes required for its functionality. Therefore, its software shall run all the PHY, MAC, scheduler and networking tasks required to operate a complete BS with relay functionality [Chochliouros *et al.*, 2009a].

The main benefit gained by adding relay stations in WiMAX mesh networks is the creation of relatively small areas with excellent coverage, increased throughput and high capacity availability without the need of any dedicated backhaul equipment.

The main goal of this research is to study and evaluate how load-balancing routing could be implemented in a WiMAX mesh network integrating relay stations. The load balancing logic, which includes routing schemes, can reside a) in the Customer-Premises Equipment (CPE), b) in the wireless Multi-hop Relay Base Station (MRBS) and c) in the wireless Multi-hop Relay Station (MRS). Although in the case of Wi-Fi mesh networks the decisions are mostly made in a distributed manner, in this research load balancing and therefore routing, are controlled and initiated by the wireless Multi-hop Relay Base Station (MRBS).

The results are expected to affect routing schemes used in 4G technology networks, since the algorithms produced can be used accordingly in a LTE/LTE-Advanced network or even in a Wi-Fi mesh network. Finally, since the importance of mesh networks is taken for granted and in this study the ability to use them in 4G and especially WiMAX is presented, the results will hopefully affect future releases of WiMAX.

However, limitations for conducting the current research exist. These are mainly the lack of simulators for demonstrating the performance of the designed algorithms and the current frame structure of the 802.16j standard that limits the number of hops to a maximum of two within a path. Therefore, this research will be fully exploited with the new version of WiMAX where the frame structure will enable the use of more relays in a route.

1.3 Scope of the thesis

The goal of this study was the design of novel routing algorithms for WiMAX mesh networks. Throughout the research, two cases have been identified, therefore two routing algorithms have been produced; one for each case. For the first algorithm, unicast transmission in a No Intra-Cell Interference (NICI) WiMAX mesh network is being studied, and has also been extended to support Limited Intra-Cell Interference (LICI) mesh networks allowing the simultaneous unicast transmission of tiers of nodes. For the second one, multicast transmission in a LICI WiMAX mesh network is being investigated.

The algorithms have been evaluated to the maximum possible extent, since the concept of a WiMAX mesh network with relay stations is not standardised yet. Network-level simulations of the algorithms remains outside the scope of the thesis, given that there are currently no software packages that support the design of WiMAX mesh networks integrating relay stations.

The four steps for the completion of the research are presented in Figure 1-1. The first step was the background study on the subject. The second one was the modelling of the system for both cases mentioned above and the third one was the design and analysis of the algorithms. Finally, conclusions have been extracted and the course of future work has been identified.

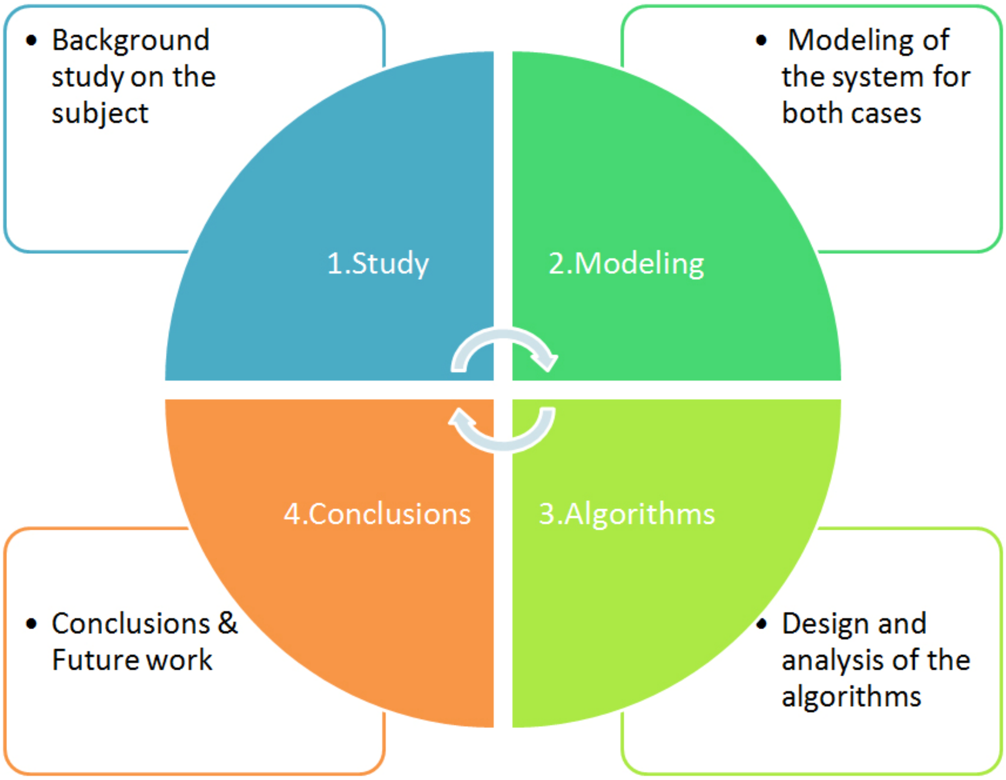


Figure 1-1 *Course of the thesis*

1.4 Contribution to knowledge

The distinct contributions of this research to the field of wireless telecommunications are:

- An algorithm for unicast transmission in No Intra-Cell Interference (NICI) WiMAX mesh networks based on Dijkstra has been designed and mathematically proved to be working.
- Additionally, it has been extended to support Limited Intra-Cell Interference (LICI) networks created when simultaneous unicast transmission of nodes is allowed.
- Another algorithm for Limited Intra-Cell Interference (LICI) multicast WiMAX mesh networks based on the Ford-Fulkerson algorithm and its Edmonds-Karp version has been designed and mathematically proved to be working.
- This research was performed within the context of the FP7-REWIND project and has led to significant results adopted by the consortium members. Afterwards, the consortium expanded both the results and the scope of the research and contributed a part of them to the standardisation bodies. The accepted contributions are listed in appendix I of the thesis.
- Many publications have been made to journals and conferences as an outcome of this research, while many more are prepared for submission. These are also listed in appendix I.
- An attempt to extend the routing algorithms so as to lead to load-balancing decisions has been made in the last chapter of the thesis.

1.5 Thesis outline

The thesis is outlined in a way to reflect the course of the work done so far and concludes with the two novel routing algorithms designed for WiMAX mesh networks.

Chapter 2 provides the main concepts and technologies used for the completion of the thesis. At first, WiMAX is introduced, along with its basic advantages and its importance as a technology, in order to explain why it was selected over other alternatives. After that, basic concepts of mesh networks used are presented. Next, a comparison is made between distributed and centralized algorithms explaining why centralized algorithms have been designed. Finally, routing algorithms are analyzed in order to identify for which cases algorithms are designed.

Chapter 3 performs an analysis of related research performed in the field and identifies important work done by other researchers. This research has helped to identify the field for which no work has been done so far, thus it ensures the originality of the ideas presented in the following chapters.

Chapter 4 introduces basic notions necessary to comprehend the ideas described in the following chapters. The proposed WiMAX mesh network architecture is presented and depicted schematically. The next section supports the centralized choice for the routing algorithms designed and clarifies in which network entity they are implemented. The last section provides a brief description of the algorithms designed and analysed in the next chapters.

Chapter 5 describes the first algorithm developed for unicast transmission, in NICI networks, based on the classic Dijkstra algorithm. In the beginning, the composite rate of each subscriber, that enables utilization of the standard Dijkstra algorithm, is calculated. After that, and in order to maximize this composite rate, the network is represented as a directed graph, while a formula for assigning weights is provided. The next section describes how the utilization of the algorithm in Limited Intra-Cell Interference (LICI) networks is enabled. Finally, the dynamic delta end-user optimization is analyzed.

Chapter 6 describes the algorithm developed for multicast transmission in LICI networks based on the Ford-Fulkerson algorithm and its Edmonds-Karp version. The first section outlines the idea and the relation of flow networks with multicast transmission in LICI networks providing the required definitions. After that, a description of the standard Ford-Fulkerson algorithm is provided and the designed algorithm is analyzed.

Chapter 7, the last chapter of the thesis, recapitulates the algorithms designed. One of the sections describes the load balancing aspect of the algorithms created, explains how load balancing decisions can be affected and gives insight of how it can be extended to a more general scheme. Moreover, it provides direction for future work and a lead on how the work presented can be further developed. Finally, examples of other networks in which the algorithms designed can be used are provided.

2 Overview of WiMAX, mesh networks and routing

2.1 Introduction

This chapter provides a summary of the concepts used in this research and an insight of the ideas supported.

Section 2.2 presents the outline of WiMAX; what it is, how it was created, its purpose and the importance of the technology. Subsection 2.2.1 presents the basic notion for wireless Multi-hop Relay Stations (MRSs). The subsequent paragraph 2.2.2 describes the IEEE 802.16™ standard, while subsection 2.2.3 states the benefits of WiMAX systems.

Section 2.3 briefly describes the basic ideas behind mesh networks and their functionality. It also outlines the network topologies used for the design of the algorithms.

Section 2.4 makes a brief comparison between centralised and distributed algorithms stating the advantages and disadvantages of both implementations. Based on this short analysis, the decision for the design of the two routing algorithms presented in the thesis has been made.

Section 2.5 identifies the various routing schemes and types of algorithms used in communication networks. Explanation is provided for selecting unicast and multicast transmission models.

2.2 WiMAX

WiMAX™ was named by the WiMAX Forum®, an industry-led, non-profit organization, assembled in June 2001. Its main purpose is to espouse and support WiMAX™ so as to be adopted by vendors and operators as the future technology trend. The WiMAX Forum® also performs tests to certify implementations based on the IEEE 802.16™ standard, which was first adopted by IEEE in 2003 in order to meet the requirements of the market for Broadband Wireless Access (BWA). The aim of WiMAX depicted in Figure 2-1 is to combine cheap, quick and flexible network deployments, to support portability/mobility and to provide high capacity, wide coverage and secure and qualitative communication at the same time under all conditions.

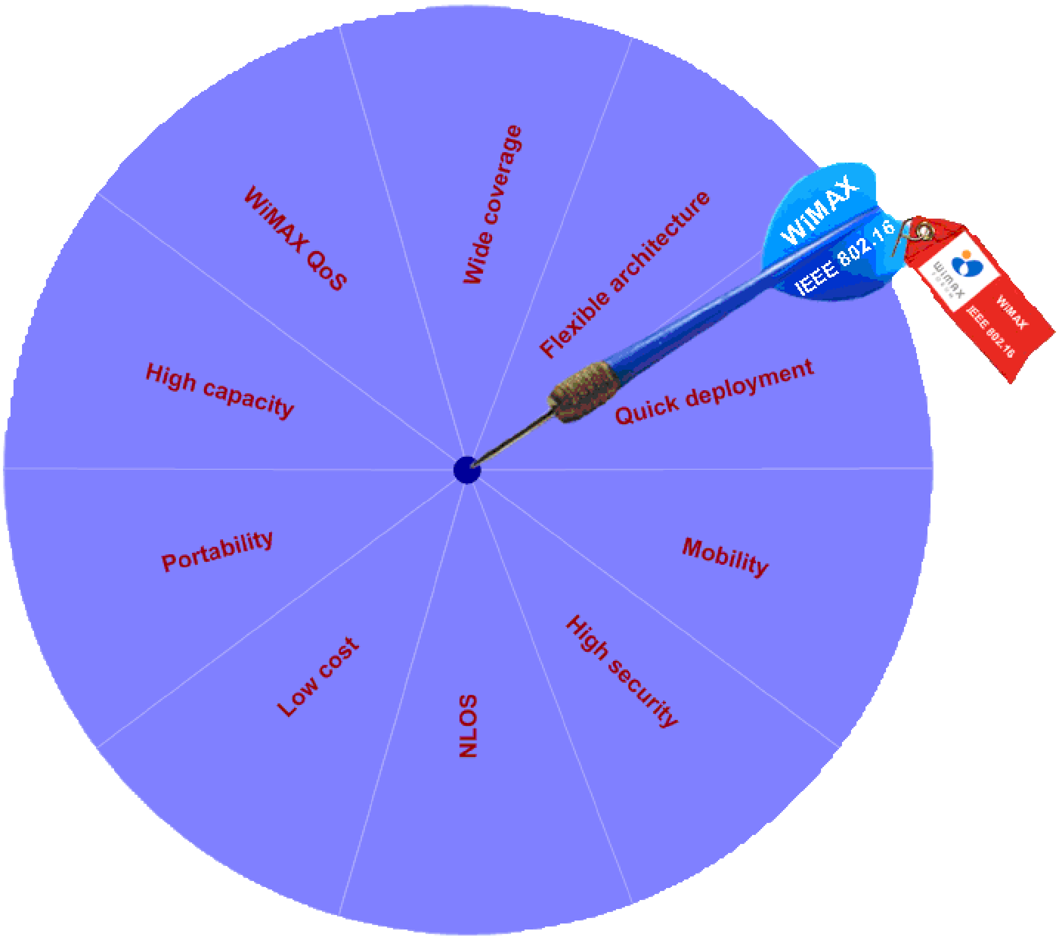


Figure 2-1 *WiMAX objectives*

WiMAX has been designed [Chen & Marca, 2008] to perform in the range of 2-66GHz, to support high data bit rates up to 75Mbps and to provide service in distances up to 50 km for static installations. It is also expected to support data bit rates up to 25Mbps to a distance of up to 5-15 km for mobile stations. All these values refer to Line-Of-Site (LOS) conditions. WiMAX operates on both licensed and non-licensed bands and can be used for deploying wireless networks but over longer distances and with less interference problems than Wi-Fi.

The technical specifications of the communications protocol are defined by the IEEE 802.16™ standard which is described in section 2.2.2. The technical features of WiMAX include Multiple Input/Multiple Output (MIMO) smart antenna technology and, either the Orthogonal Frequency Division Multiplexing (OFDM) or the Orthogonal Frequency Division Multiple Access (OFDMA).

MIMO uses multiple antennas at both ends of the wireless link to enable data transmission along multiple paths [Xiao, 2007]. This means that, when a 2x2 setting is mentioned, there are two transmit antennas on the base station and two receive antennas on the subscriber’s device, while in a 2x4 setting, there are two transmit antennas on the base station and four receive antennas on the subscriber’s device.

Both OFDM and OFDMA provide high spectral efficiency and the ability to deal with severe channel conditions. Additionally, OFDM is used in the IEEE 802.16-2004 standard, while OFDMA is used in the IEEE 802.16e-2005 and in subsequent releases.

The WiMAX Forum® offers the framework of testing the compatibility of manufacturers’ equipment and promotes both the advancement and the commercialization of the technology. WiMAX supports a variety of applications with different features, as described in Table 2-1 [WiMAX Forum, 2005].

Table 2-1 *WiMAX Service Classes*

Class Description	Real Time	Application Type	Bandwidth
Interactive gaming	Yes	Interactive gaming	50-85 kbps
VoIP, Video Conference	Yes	VoIP	4-64 kbps
		Video Phone	32-384 kbps
Streaming Media	Yes	Music/Speech	5-128 kbps
		Video Clips	20-384 kbps
		Movies Streaming	> 2 Mbps
Information Technology	No	Instant Messaging	< 250 byte messages
		Web browsing	> 500 kbps
		Email (with attachments)	>500 kbps
Media Content Download (Store and Forward)	No	Bulk data, Movie download	> 1 Mbps
		Peer-to-Peer	> 500 kbps

Within the context of 4G, WiMAX has to compete with systems such as UMTS and CDMA2000, both of which can provide DSL-level Internet access and phone services at the same time. UMTS has recently been upgraded and renamed UMTS-TDD. On the other hand, CDMA2000 has been based on Ultra Mobile Broadband. The main standards for mobile telephony developed comprise the 4G technology having as basic characteristics high bandwidth and short delays.

Figure 2-2 schematically presents a comparison between various prevailing wireless broadband technologies regarding the data rate and mobility they can offer. Samsung’s view (2009) is that Mobile WiMAX high-speed data services offered to mobile users are closer to 4G, going beyond 3G and this is presented in Figure 2-3.

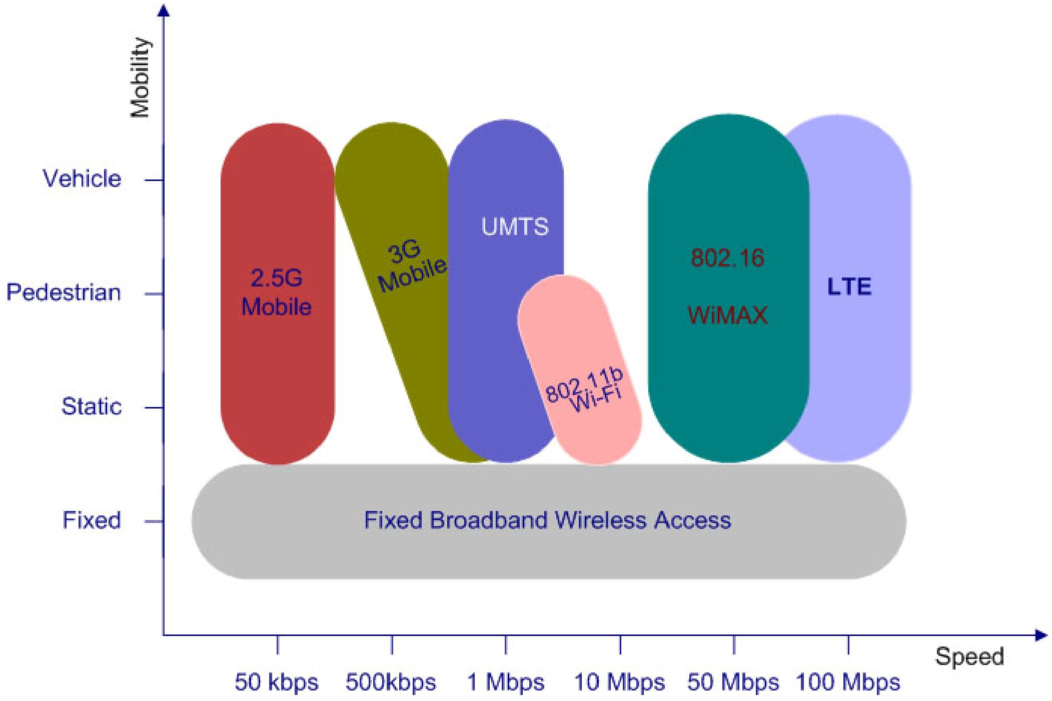


Figure 2-2 *Speed vs. Mobility for wireless technologies*

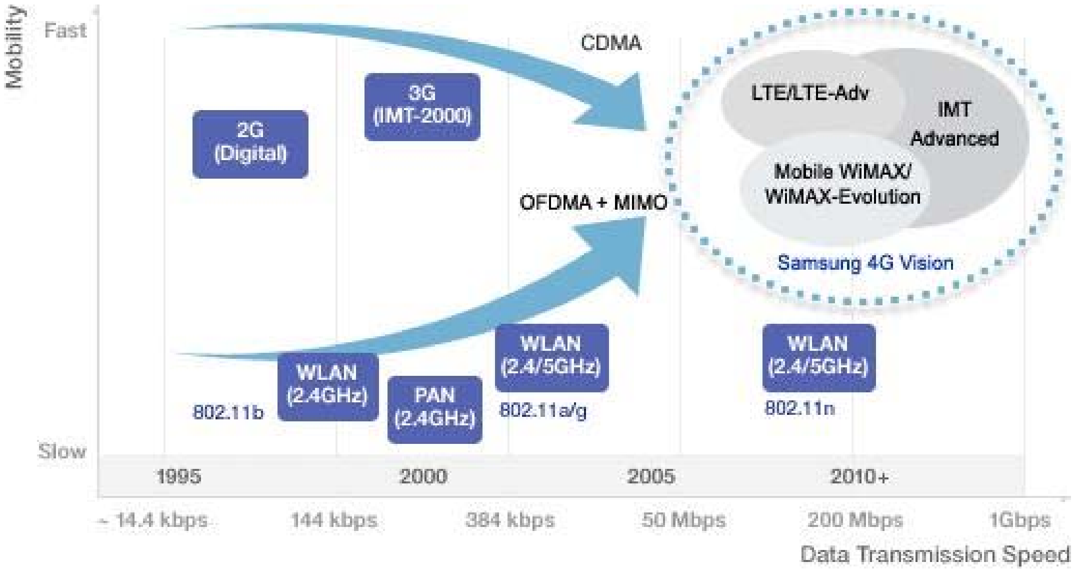


Figure 2-3 *Samsung's view for 4G*
(Resource: Samsung)

2.2.1 Introduction of wireless Multi-hop Relay Stations (MRSs)

It has become apparent in the recent years that in order for the next generation of wireless technology (whether this is WiMAX, LTE or any other 4G implementation) to be able to deliver ubiquitous broadband content, the network is required to provide excellent coverage, both outdoor and indoor, and significantly higher bandwidth per subscriber [Voudouris *et al.*, 2009]. In order to achieve that at frequencies above 2 and 3 GHz, which are targeted for future wireless technologies, network architecture must reduce significantly the cell size or the distance between the network and subscribers’ antennas.

While micro, pico and femto Base Transceiver Station (BTS) technologies reduce the cost of base-station equipment, they still rely on a dedicated backhaul. One solution introduced with the WiMAX 802.16j standard is the wireless Multi-hop Relay Station (MRS), intended to overcome these challenges. On one hand, it should be small, cost-effective and easy to install for enabling mass deployment in indoor and outdoor environments and creating relatively small areas with excellent coverage and high capacity availability. On the other hand, it does not require any dedicated backhaul equipment as it receives its capacity from centralized base-stations via the same resources used for the access service. The network topology of applications integrating MRSs is shown in Figure 2-4.

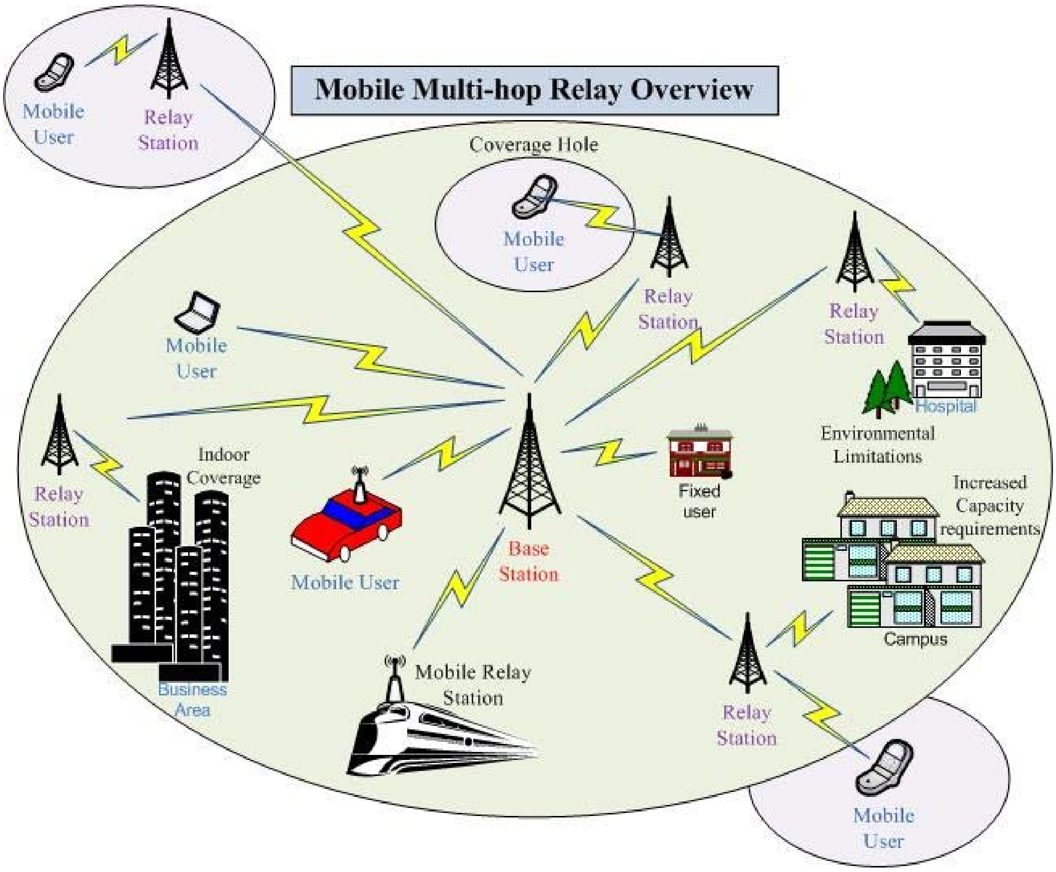


Figure 2-4 *Network topology of Relay Station applications*

In a setting where a MRS exists, enabling MIMO transmission, the link referred to needs to be specified. This means that, when a 2x2 setting is mentioned, there can be either two transmit antennas on the base station and two receive antennas on the relay station, or two transmit antennas on the relay station and two receive antennas on the subscriber’s device [Chochliouros *et al.,* 2009b].

Wireless Multi-hop Relay Stations (MRSs), when are deployed in various sights, result in increased throughput or coverage. Such scenarios are described in Table 2-2, indicating the essential attributes that need to be met for the successful operation of WiMAX in those events [WiMAX Forum, 2005].

Table 2-2 *Scenarios of usage and requirements*

	Flexible Architecture	High security	WiMAX QoS	Quick deployment	Interoperability	Portability	Mobility	Cost-effective	Wide coverage	NLOS	High capacity
Cellular Backhaul				✓				✓			✓
Banking networks	✓	✓	✓					✓		✓	
Educational networks	✓		✓					✓	✓		
Public safety	✓	✓	✓	✓		✓	✓			✓	
Offshore communications	✓		✓			✓	✓		✓	✓	
Temporary construction			✓	✓		✓				✓	
Rural connectivity			✓		✓			✓	✓		✓
Military operations	✓	✓		✓		✓	✓				
Emergency situations	✓			✓		✓	✓			✓	

A general case, where a relay station can be used, is in situations with coverage constraints such as areas where there is presence of physical obstacles (e.g. buildings, forests), or in indoor coverage cases. Some examples are large office buildings, University campuses, and villages in unreachable areas on rockier uplands etc.

Another scenario, where MRSs can be used, is for high mobility users with increased bandwidth requirements, such as trains with a great number of wireless users. Such a mobile subscriber will more likely have data rate degradations due to non-fixed position. In this case, a relay station can be considered as the most feasible solution in terms of cost and easiness of installation in every public transport vehicle, providing increased coverage and throughput to mobile WiMAX users.

In order to achieve certain bit error rate levels on the data transmitted to the subscribers, WiMAX uses adaptive modulation. In case the subscriber is far from the base station or the environment introduces a lot of interference, the modulation used will be adapted accordingly, reducing the available data rate of the user. The use of a relay station can improve the provided service to the end-user, since it can transcode the received signal from the base station increasing the data rate provided to that user. This scenario is applicable in suburban environments, where users are usually away from the base stations, as well as in environments with increased interference.

These scenarios demonstrate that by using relay stations in WiMAX networks you gain:

- Increased Coverage
- Increased Throughput/QoS
- Support of Mobility
- Decreased Cost with respect to base station installation
- Improved frequency planning

2.2.2 The IEEE 802.16™ standard

The technical specifications of the WiMAX™ communications protocol are defined by the IEEE 802.16™ standard. The IEEE 802 LAN/MAN Standards Committee sets international standards for Local Area Networks (LANs) and Metropolitan Area Networks (MANs).

The family of IEEE 802 standards separates the Data Link Layer to 2 sub-layers. The first is the Logical Link Control (LLC) and the second is the Medium Access Control (MAC). The LLC was introduced in the 802.2™ and is common for all 802 MACs. IEEE 802 projects generally work on the PHYsical (PHY) and Medium Access Control (MAC) layers.

IEEE 802.16™ is a group of specifications for wireless broadband networks. The evolution of the IEEE 802.16 projects is shown in Table 2-3. In 2003, the 802.16a standard was released, including Orthogonal Frequency Division Multiplex (OFDM) and allowing transmission of data through non-line of sight conditions. In 2004, the 802.16-2004 version was released combining the updates from previous versions and extending the range of WiMAX service to 50 km for fixed access. In 2005, 802.16e, the first Mobile WiMAX system was released, using the Scalable Orthogonal Frequency-Division Multiple Access (SOFDMA) modulation and including better support for QoS.

The current version is IEEE 802.16-2009, amended by the IEEE 802.16j-2009, which adds relaying functionality in WiMAX networks based on the IEEE 802.16e-2005 standard, being fully compatible with it. The aim for the future is the 802.16m release with the goal to increase data rates to 1Gbps for fixed access and up to 100Mbps for mobile access.

Table 2-3 *The IEEE 802.16 advancement*

Standard	Description	Status
802.16-2001	Fixed Broadband Wireless Access (10–66 Ghz)	Superseded
802.16.2-2001	Recommended practice for coexistence	Superseded
802.16c-2002	System profiles for 10–66 Ghz	Superseded
802.16a-2003	Physical layer and MAC definitions for 2–11 Ghz	Superseded
P802.16b	License-exempt frequencies	Withdrawn
P802.16d	Maintenance and System profiles for 2–11 Ghz (Project merged into 802.16-2004)	Merged
802.16-2004	Air Interface for Fixed Broadband Wireless Access System (rollup of 802.16-2001, 802.16a, 802.16c and P802.16d)	Superseded
P802.16.2a	Coexistence with 2–11 Ghz and 23.5–43.5 GHz (Project merged into 802.16.2-2004)	Merged
802.16.2-2004	Recommended practice for coexistence (Maintenance and rollup of 802.16.2-2001 and P802.16.2a)	Current
802.16f-2005	Management Information Base (MIB) for 802.16-2004	Superseded
802.16-2004/Cor 1-2005	Corrections for fixed operations (co-published with 802.16e-2005)	Superseded
802.16e-2005	Mobile Broadband Wireless Access System	Superseded
802.16k-2007	Bridging of 802.16 (an amendment to IEEE 802.1D)	Current
802.16g-2007	Management Plane Procedures and Services	Superseded
P802.16i	Mobile Management Information Base (Project merged into 802.16-2009)	Merged
802.16-2009	Air Interface for Fixed and Mobile Broadband Wireless Access System (rollup of 802.16-2004, 802.16-2004/Cor 1, 802.16e, 802.16f, 802.16g and P802.16i)	Current
802.16j-2009	Multihop relay	Current
P802.16h	Improved Coexistence Mechanisms for License-Exempt Operation	in progress
P802.16m	Advanced Air Interface with data rates of 100 Mbit/s mobile & 1 Gbit/s fixed	in progress

The IEEE 802.16j Mobile Multi-hop Relay (MMR) specifications aim to extend base station reach and coverage for WiMAX networks, while minimizing wireline backhaul requirements. The relay architecture will allow operators to use in-band wireless backhaul while retaining all the standard WiMAX functionality and performance [Chochliouros *et al.*, 2009c].

The IEEE 802.16j working group have defined the following:

- Definition and terminology used in IEEE 802.16j environment.
- A set of guidelines, focused on channel models, traffic models and performance metrics, for the evaluation and comparison of technology proposals for IEEE 802.16j.
- A set of use-case scenarios involving Relay Stations.
- Description of technical requirements for Relay Stations.
- New frame structure to support Relay Stations.
- OFDMA physical and MAC layer enhancements to IEEE 802.16 specifications to support Relay Stations.
- Centralized vs. distributed network control.
- Centralized vs. distributed Scheduling.
- Radio Resource management.
- Power Control mechanism.
- Call Admission and Traffic Shaping Policies.
- QoS based on network wide load balancing and congestion control.
- Security issues.

The notion of a service flow is also specified by WiMAX. This is a unidirectional data stream with specifically defined QoS parameters such as traffic priority, scheduling type, maximum delay etc. Service flows are either created dynamically or assigned through a network management system. The MAC scheduler of the base station must support the five service flows that WiMAX has identified, to meet the QOS requirements of a wide variety of applications, described in Table 2-4 [WiMAX Forum, 2006a, WiMAX Forum, 2006b].

Table 2-4 *WiMAX service flows*

Service Flow Designation	Description	Qos parameters defined	Applications
Unsolicited grant services (UGS):	Supports fixed-size data packets at a Constant Bit Rate (CBR).	Maximum sustained traffic rate and latency Tolerated jitter Request/transmission policy.	Voice over IP (VoIP) without silence suppression
Real-time polling services (rtPS):	Supports real-time service flows that generate variable-size data packets on a periodic basis.	Minimum reserved traffic rate Maximum sustained traffic rate Maximum latency Request/transmission policy.	Streaming audio and video, Motion Picture Experts Group (MPEG) encoded
Non-real-time polling service (nrtPS):	Supports delay-tolerant data streams that require variable-size data grants at a minimum guaranteed rate.	Minimum reserved traffic rate Maximum sustained traffic rate Traffic priority Request/transmission policy.	File Transfer Protocol (FTP),
Best-effort (BE) service:	Supports data streams that do not require a minimum service-level guarantee.	Maximum sustained traffic rate Traffic priority Request/transmission policy.	Web browsing Data transfer
Extended real-time variable rate (ERT-VR) service:	Supports real-time applications that have variable data rates but require guaranteed data rate and delay. Defined only in IEEE 802.16e-2005. It is also referred to as extended real-time polling service (ErtPS).	Minimum reserved traffic rate Maximum sustained traffic rate Maximum latency Jitter tolerance Request/transmission policy.	Voice over IP (VoIP) with silence suppression

2.2.3 Benefits from WiMAX

According to WiMAX Forum®, the benefits of using WiMAX are:

- *WiMAX QoS*. WiMAX, with the use of service flows, can be dynamically optimized for its network traffic.
- *Interoperability*. Network devices are standard-based implementations leading to interoperable solutions from multiple vendors.
- *Security*. Two protocols supported by WiMAX are Advanced Encryption Standard (AES) and Triple Data Encryption Standard (3DES), while minimum encryption of the network traffic is also required.
- *Portability*. Once the WiMAX Subscriber Station is switched on, it identifies itself, resolves the quality and features of the link with the Base Station and, finally, negotiates its transmission characteristics accordingly.
- *Mobility*. MIMO, Scalable OFDMA, NLOS performance and support for (hard and soft) hand-off, extend the support of devices and services in a mobile environment.
- *Long Range*. WiMAX has a range of up to 50 km for fixed stations and up to 5-15 km for mobile stations.
- *Wide coverage*. BPSK, QPSK, 16-QAM, and 64-QAM are supported by WiMAX and can be dynamically assigned. When operating with a low-level modulation, WiMAX systems have a wide range, under LOS conditions.
- *High capacity*. When operating with a higher modulation, WiMAX systems can serve end-users with increased bandwidth.

- *Service*. WiMAX can provide users with service under two environments:
    - *Non Line-of-Sight*: OFDM technology enables WiMAX to deliver broad bandwidth under NLOS conditions, usually at 2-11 GHz, where it has the ability to overcome obstacles more easily.
    - *Line-of-Sight*: Under such conditions, the transmission can go up to 66 GHz, since the signal is stronger and more stable, providing users with greater bandwidth.
- *Quick, Flexible & Scalable Deployments*. WiMAX supports several network topologies like Point-to-Point and Point-to- Multipoint. Along with the interoperability support, operators can rapidly deploy their networks and easily scale to any size they need.

2.3 Mesh networks

Mesh networks are those whose nodes are interconnected either directly with each other or through other nodes, but always with more than one path and in such a way that closed loops are created [Held, 2005, Hossain, 2008]. The reliability of mesh networks lies in the fact that they remain operational even when a node stops working or a link is broken. Data sent over a mesh network can take any of several possible paths connecting the node that has initiated the transmission to the target node.

Depending on the number of existing connections among nodes, mesh networks can be divided into two categories, as shown in Figure 2-5:

- *Fully connected;* every node is connected to each other node. It is a quite complex and expensive topology that requires maintenance on all links, but provides the maximum number of routes among nodes.

- *Partially connected;* every node is connected to all others not necessarily directly but through other nodes. It is a simpler and cheaper topology, where the network administrator can select the number of links per node or even a different number of links for each node. Therefore, it is upon the administrator to define the complexity of the network, requiring a deliberate design of it.

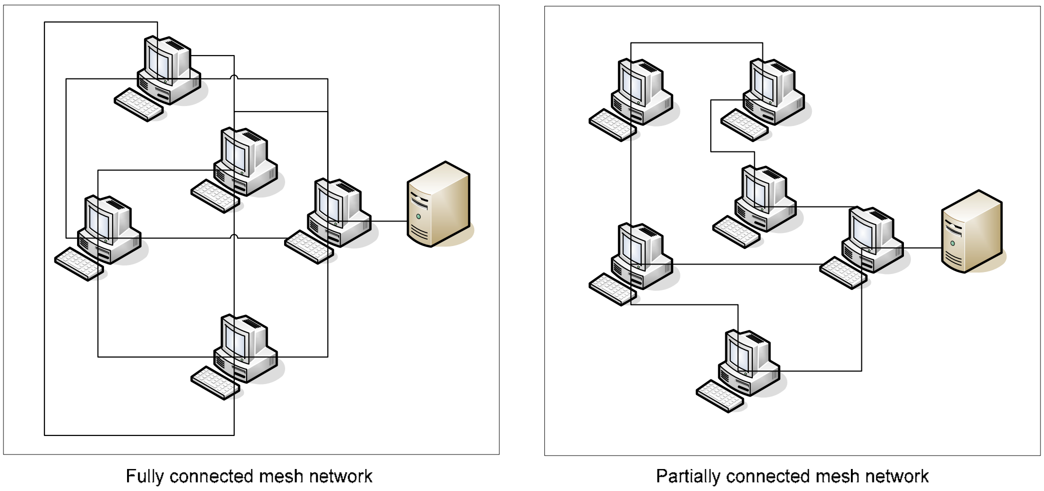
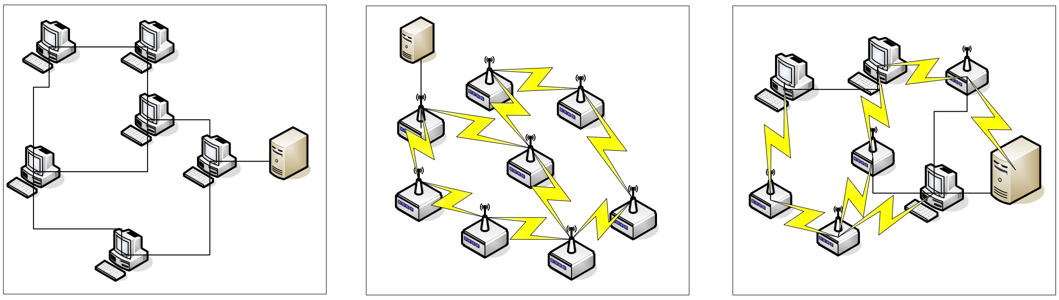

Fully connected mesh network
Partially connected mesh network

Figure 2-5 *Fully and partially connected mesh networks*

Nodes in mesh networks can be PCs, PDAs, laptops, sensors, modems, hubs, switches, routers, repeaters and almost anything that can transmit or retransmit data. As shown in Figure 2-6, according to the interfaces of the links among the nodes, there are three types of mesh networks:

- *Wired;* all nodes are interconnected wired to each other.
- *Wireless;* all nodes are interconnected wirelessly to each other.
- *Mixed;* all nodes are interconnected either wired or wireless to each other.



Wired mesh network Wireless mesh network Mixed mesh network

Figure 2-6 *Wired, wireless and mixed mesh networks*

A Wireless Mesh Network (WMN) consists of wireless nodes interconnected in a mesh style. Various communication protocols - including Wi-Fi and WiMAX - can be used for deploying such a network [Akyildiz *et al., 2005*]. Originally, WMNs had been developed for military applications, but have greatly evolved during the last years. Applications for which wireless mesh networks are used nowadays include:

- Internet access for cities, municipalities and isolated areas
- Educational networks
- Healthcare systems
- Temporary venues
- Warehouses
- Military applications

Current research in the field of WMNs is focused on improving performance, extending coverage and increasing flexibility of network deployment and management. Due to the endless demand for portable handheld devices and the requirement for Internet connection to support modern applications and services, changes in the topology of the network are very often, making routing a challenging issue. In addition, nodes may have limited capabilities and, therefore, require a control on how to forward data to avoid overloading.

2.4 Centralized vs distributed algorithms

One major classification regarding algorithms can be made as to centralized or distributed (decentralized) ones according to the place where computations are made and decisions are taken [McConnell, 2007]:

- *Centralized algorithms:* A central point (coordinator) is used to perform all necessary calculations. The coordinator makes all the necessary decisions imposing them to other elements of the network. Sometimes, these algorithms include the management of other nodes, while calculations and other processes occur sequentially. The main advantages of centralized algorithms are:
    - *Hardware costs;* there are usually lower operational and infrastructure costs.
    - *Complexity of infrastructure;* the complexity of infrastructure is reduced and more reliable systems are used to improve and ensure the integrity of data. In addition, services and data are more easily restored when a failure occurs.
    - *Security;* easier security management, thus, a greater degree of control is applicable. Additionally, the physical security of nodes is achieved more easily.
    - *Upgrades;* hardware and software upgrades can be achieved much more easily.

- *Distributed (decentralized) algorithms:* In this case, a coordinator does not exist, while processes occur in parallel (simultaneously) and there is minimum interaction among nodes. These algorithms can be implemented in two ways. In the first method, parts of an algorithm are executed concurrently on independent nodes. When they finish, one of the nodes receives the results and constructs the total outcome or makes the necessary decisions based on the separate results. In the second approach, each node runs the complete algorithm independently and performs the necessary actions on its own.

  One of the major challenges in developing and implementing distributed algorithms is the successful coordination of nodes, along with the effective management of failures and loss of communications links. The choice of an appropriate distributed algorithm depends a) on the characteristics of the problem, b) on the characteristics of the system, c) the type of nodes, d) the probability of link failures and e) the desired level of synchronization between separate processes and nodes. Distributed algorithms have the following advantages:

  - *Computational efficiency;* computational load is distributed among nodes.
  - *Communication efficiency;* communication overheads regarding the links between nodes are minimized.
  - *Stability;* unpredictable changes to network conditions up to one level do not produce significant changes.
  - *Fairness;* every node is treated in the same way defined by network parameters. Rules and their changes apply to all nodes.

2.5 Routing

Routing is the course of actions followed for identifying routes within a network along which the source node(s) will transmit information to the target node(s) [Osterloh, 2002]. The purpose is to find the optimal path taking into account parameters such as the distance between nodes, time delay and communication cost for the transmission, affecting the performance of the network and its QoS level.

In Figure 2-7, the four routing schemes are illustrated; the blue nodes are the source and the green ones are the target nodes. Routing schemes differ in the way information is transmitted.

- *Anycast* is the scheme where data are transmitted to any node, usually to the one nearest to the source.
- *Broadcast* is the scheme where data are transmitted to all nodes in the network.
- *Unicast* is the scheme where data are transmitted to one preselected node.
- *Multicast* is the scheme where data are transmitted to a predefined group of nodes.

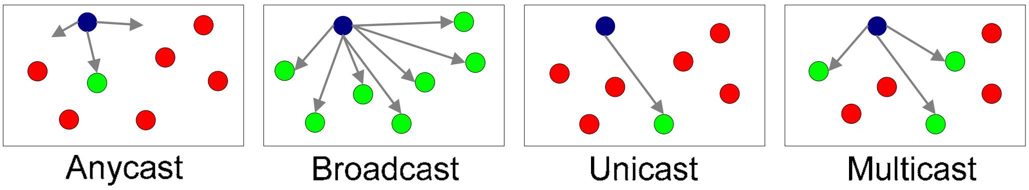


Figure 2-7 *Routing schemes*

In the cases of anycast and broadcast, no complex routing algorithms need to be designed and implemented. More explicitly, in anycast the transmitter sends data without knowing or caring to which node data are sent, and usually the closest node receives them, while in broadcast the transmitter sends data to all nodes.

The other two cases of unicast and multicast are the most demanding ones and require the development of routing algorithms. In these cases, different routing methods can be applied on the networks based on their characteristics such as size, topology etc [Osterloh, 2002]. These methods are:

- *Static routing;* in this case, all possible routes are manually predefined and stored in the routing table usually by the network designer or the administrator. Moreover, nodes don't exchange information regarding network topology. This method is used mainly in small networks where parameters, such as the number of nodes and the topology, aren't modified frequently.
- *Adaptive/Dynamic routing;* in this case, routing algorithms are being implemented taking into account various network parameters, such as distance, hops, delay, bandwidth and speed. Hence, routing tables are built and updated dynamically in set intervals adapted to changes made.

It is important to bear in mind that, although dynamic routing is more flexible and detects changes automatically in network topology, it comes with a higher cost in bandwidth, possibly in resources and in processing time. Nowadays, most networks are dynamic incorporating complex topologies, supporting scalability, thus making the use of static routing almost impossible.

The models selected to be analysed in this thesis are unicast and multicast. In the unicast transmission in a WiMAX mesh network as considered in this research, a node will try to contact a specific user either within its range or within the range of another node. On the other hand, in the multicast transmission, a node will try to serve a group of users at the same time (simultaneously).

Within an autonomous network, like a WiMAX mesh network selected for this research, there are two major types of algorithms used for routing, Global or Link State (LS) and Decentralized or Distance Vector (DV) algorithms [Osterloh, 2002]. The feature that differentiates them is the way nodes collect information regarding the topology and state of the network and the evaluation of data upon which they choose a route. More explicitly:

- *Global or Link State (LS) routing algorithms.* LS protocols use more complex methods taking into account parameters such as the link state, bandwidth and delay. The basic concept is that each routing node has information about the rest of the routing nodes in the network. In that case, every routing node builds a graph, containing the nodes to which it is directly connected and the cost to contact each of them, taking into account parameters such as speed, delay time, average traffic, hops etc. After that, LS algorithms simply iterate over the collection of graphs residing in each node, creating paths. The paths, with the lowest sum of costs from a source node to every other node, form the node's routing table which identifies the best next hop to get from one to any other node. LS routing algorithms update the routing tables only in the case where at least one of the network's parameters change.

  The most known LS routing algorithm is the Dijkstra algorithm. Examples of LS protocols are the Open Shortest Path First (OSPF) and the Intermediate System to Intermediate System (IS-IS) which only supports IP traffic.

- *Decentralized or Distance Vector (DV) routing algorithms.* DV protocols decide upon the best route based on how far the target of the transmission is. In this case, a number specified as cost, is assigned to each of the existing connections among all pairs of interconnected routing nodes in the network, usually based on the distance between them. Additionally, information is exchanged among directly connected nodes in order to calculate all possible paths from a source node to a destination node. Finally, data are transmitted via the path with the lowest sum of costs. DV routing algorithms update the routing tables periodically, regardless whether the network has changed or not.

  The most known DV routing algorithm is the Ford–Fulkerson algorithm and its Edmonds–Karp version, while the best known and most popular DV protocol is Routing Information Protocol (RIP) used in Internet.

2.6 Summary

A summary of the theory used and the ideas supported in this research has been provided in the previous sections. WiMAX and its importance as a technology have been presented, while the concept of using wireless Multi-hop Relay Stations (MRSs) in various scenarios is signified. Additionally, the basic ideas behind mesh networks and their functionality are described, presenting the network topologies used for the design of the algorithms. To support further the ideas introduced in this research, a small comparison between centralised and distributed algorithms is made, resulting to the decision taken for the development of the routing algorithms presented in this thesis. Finally, the various routing schemes and types of algorithms used in communication networks are presented and explanation is provided for selecting unicast and multicast as the transmission models.

3 Related research

3.1 Introduction

As already mentioned, wireless networks have rapidly been developed during the recent years and one of the main research subjects is the way routing of data can be implemented in these networks. Therefore, it is reasonable that there are a great number of studies performed and many routing algorithms developed.

The classical Dijkstra and Ford-Fulkerson algorithms, that have stimulated this research, will be analyzed in sections 5.3 and 6.4 respectively. Thus, in section 3.2, some modern approaches and algorithms created will be detailed, while the main advantages and disadvantages of each one will be identified. In section 3.3, some earlier approaches and algorithms created will be described, while in Table 3-1 a summary and an overview of algorithms presented in this chapter is provided.

3.2 Recent advancements

As will be shown in the following chapters, the notion of extending the WiMAX mesh network architecture to include relay stations is completely new and is not yet supported by the current standards. Therefore, there are no algorithms that directly support this concept and can be compared to the ones introduced by this research. However, due to the wide range of routing protocols in wireless networks, algorithms designed for WMNs can be presented, since it is possible that they will be extended to support new architectures, like WRMN.

3.2.1 Fixed routing algorithm

One of the main concerns in networks today is the level of QoS they offer to users. Shetiya & Sharma [2005] have made an effort to create a centralized fixed routing algorithm for supporting QoS. Their idea was to design a scheme that would always provide the same route under all channel conditions.

System model

This study has used the IEEE 802.16 mesh mode. A region is split into small meshes with a base station in each one of them. The rest of the nodes are subscriber stations (SSs). A transmission occurs a) between two SSs inside one of the meshes without necessarily engaging the base station, or b) between two different meshes through the corresponding base stations.

The algorithm

The authors assume that the routing is fixed. In that case, $r_k(i, j)$ is the allocated transmission rate, $X_k(i, j)$ the data received from other meshes and $Y_k(i; j)$ the data received from other nodes within the mesh to node i for output link (i, j) during the frame k. $Q_k(i, j)$ is the queue length at node i for output link (i, j) in the beginning of the frame k. Also, it holds that $\lambda_{i,j} = E[X_k(i, j)]$. If the assumption that the schedule is fixed and the link (i,j) has always $n_{i,j}$ assigned slots in a frame, then:

$$Q_{k+1}(i, j) = (Q_k(i, j) + Y_k(i, j) - n_{i,j} r_k(i, j))^+ + X_k(i, j),$$

where $(x)^+$ is the max(0, x).

For the queue to be stable and have a unique fixed distribution, it has to hold:

$$n_{i,j}\, E[r(i, j)] > E[X(i, j) + Y(i, j)] = \lambda_{i,j} + E[Y(i, j)],$$

where the expectation E[Y(i, j)] is under the fixed distribution.

The traffic when passing through a node is not split to many routes and that implies that there is a tree structure in the network. Thus, if links are indexed by i, there will be a unique output link corresponding to node i. Let $\lambda_i = \Sigma_{0 \leq j \leq M} \lambda_{i,j}$. Then, $E[Y_i] = \Sigma_{1 \leq j \leq mi} \lambda_{ai,j}$, where $\{a_{i,1}, a_{i,2}, \ldots, a_{i,mi}\}$ are the nodes whose data pass through node i. Hence, if:

$$n_i > (\lambda_i + \Sigma_{1 \leq j \leq mi} \lambda_{ai,j}) / (E[r(i)]), \text{ for all } I = 1, \ldots, M,$$

then the system is considered to be stable. Since $\Sigma_{1 \leq i \leq M}\, n_i = N$, then:

$$\Sigma_{1 \leq i \leq M} [\lambda_i - \Sigma_{1 \leq j \leq hi}(1/(E[r(p_{i,j})]))] < N,$$

where $\{p_{i,1}, \ldots, p_{i,hi}\}$ are the nodes through which the data of node i is routed. It can be observed that for each node i, if the route that minimizes the term $\Sigma_{1 \leq j \leq hi}(1/(E[r(p_{i,j})]))$ is chosen, then the stability region can be maximized.

Therefore, standard shortest path algorithms such as Dijkstra or Bellman-Ford can be applied by assigning cost $1/(E[r(p_{i,j})])$ to link (i, j). The routing is fixed over all the frames for each node along the path that minimizes $\Sigma_{1 \leq j \leq hi}(1/(E[r(p_{i,j})]))$.

Advantages and limitations

This algorithm finds a shortest path, between the subscriber station and the base station, which remains the same under all conditions. Due to this fact, resources can be reserved and the level of QoS can be guaranteed.

On the other hand, the main disadvantage is that in wireless networks the air interface is not optimal as a medium. Therefore, when the link breaks or when the channel conditions become severe, it is most likely that the routing will fail and the same will happen in case of a node failure. It is not mentioned how this situations is dealt with, however it is possible to recalculate routes. Additionally, in order to be able to reserve resources it presupposes that there are resources at your disposal, which is not feasible, especially in cases where the network scales.

3.2.2 Interference-aware routing

Another approach is introduced by Wei *et al*. [2005]. They have presented a centralized scheme that is interference-aware and considers traffic load demand at the same time, enabling concurrent transmission and therefore ensuring high throughput and expandability of the system.

System model

The scheme is based on tree routing. This study has also used the centralized IEEE 802.16 mesh mode. A blocking metric namely B(k) has been used for routing, expressing the total number (sum) of interfered (blocked) nodes along a path, when a transmission from a source node s to a target node k occurs. The interference is caused by all the intermediate nodes along the transmission path, from s to k. Therefore, the B(k) of a path is the sum of all blocking values of transmitting nodes along the selected path.

The algorithm

In the beginning, the algorithm computes the blocking metric of all possible paths, from a source node to a destination one, and determines the path with the less interference. The aim of the scheme is to fully exploit concurrent transmissions to provide maximum throughput. In order to enable parallel transmissions, the algorithm performs an iteration of the following procedure. First, all active links are listed and then sorted descending, according to their unallocated traffic demand. The one with the highest value is the first in the list and is selected for transmitting on the first instance. The iterations continue until all traffic requests have been allocated. In this procedure, the interfering links are not accounted for.

Advantages and limitations

The authors have compared their scheme against the random scheduling mentioned in the WiMAX standards and, for chain topology, much higher throughput has been achieved, very close to the theoretical upper limit. For a random mesh topology, the performance of the algorithm is once more better than the standard's performance, but worst compared to the chain's topology.

On the other hand, the main drawback of the scheme is that the total number of blocking nodes is considered as the only metric for routing, without taking under consideration whether these nodes have any data to transmit. As a result, the real picture of the network interference is not presented.

3.2.3 Routing for throughput maximization

In the previous scheme, the only metric was the blocking metric of a path. Jin *et al*. [2007] tried to extend this notion and took into account the number of packets existing in a blocked node, trying to maximize throughput. Thus, the blocking metric in this case, for a node u, is:

$$B(u) = (\text{number of nodes blocked by } u) \cdot (\text{number of packets at } u).$$

Hence, the path selected is the one with the minimum blocking metric.

System model

The objective of the algorithm is to create a routing tree so that the number of timeslot required is to be minimized. The assumptions made by the authors are:

- A node cannot transmit and receive data at the same time
- There can only be one transmitter near a receiver
- There can only be one receiver near a transmitter

The routing tree is created with two methods. In the first one, when a new node enters the network, the routing tree is updated. Then, the base station estimates again the routing node and reconfigures the network. The second one includes the periodic reconstruction of the routing tree, taking under consideration new throughput requirements.

The algorithm

The focus is on the set of edges between two consecutive tiers. The aim is to find within this set the interfering and the non-interfering pairs. Each pair of edges is weighted with the number of packets the source node wants to transmit. The set of edges selected are the non-interfering ones and those for which the sum of weights in the set is maximized.

If U_i is the set of nodes, e_i is an edge in layer i, and $E = \{e_i, 1 \le j \le m\}$ is the set of edges between two layers, where $m = \Sigma_{1 \le i \le n}\, w_i$, then the algorithm selects a set $S \subseteq E$, so that the:

$$f(S) = \Sigma_{ej,ek} \{w(e_j) + w(e_k)\}, \text{ where } e_j, e_k \text{ are non-interfering,}$$

is maximized.

Advantages and limitations

The main advantage of the algorithm described above is that it includes both the interference and the traffic load in the calculations it makes. Additionally, it is updated even when the traffic conditions change, something that improves the algorithm presented in the previous subsection.

However, the disadvantage of this scheme is that the network has to be reconfigured whenever there is registration of a new node. For that reason, a great overhead is created. Additionally, the periodic reconfiguration of the routing tree is vaguely introduced, since the time period for the reconfiguration hasn’t been specified, while it results in extra overhead. Finally, the reconfiguration is also based on traffic conditions, so in networks where traffic is varying constantly or where mobility of nodes is allowed, the scheme will probably perform poorly and it might lead to infinite loops.

3.2.4 Routing for throughput enhancement using concurrent transmission

One of the main requirements for mesh networks is high capacity. To increase it in a multi-hop network, concurrent transmissions must be enabled. Tao *et al*. [2005] propose a simple algorithm to allow concurrent transmissions of subscribers’ stations in both uplink and downlink.

System model

The scheme introduced uses the tree based routing and the centralized IEEE 802.16 mesh mode. Based on interference knowledge, it tries to achieve concurrent transmissions. Other basic assumptions made by the authors are:

- A node cannot receive and transmit data at the same time

- There can be only one transmitter near a receiver

- There can be only one receiver near a transmitter

The algorithm

Let us consider a path BS-y-x, in a routing tree, with BS the base station, x, y the nodes, and $P_y(x)$ the sum of both uplink and downlink interference along the path from the node x to the BS. If interference is denoted by I(a,b) from node a to node b, then Py(x) is given by:

$$P_y(x) = I(x,y) + I(y,x) + P_{BS}(y)$$

Initially, the network has only the base station BS and all nodes enter the network one by one. When a node first enters the network, all of its neighbours are candidates to be its parent node. In order to minimize the interference, the entering node should select as parent, a node having the minimum interference. In order to ensure the minimum interference along a path, the father of a node x, if n(x) is the set of its neighbouring nodes, can be found by:

$$F_x = \arg(\min_{i\epsilon n(x)} P_i(x))$$

In the case where after a new entry the interference value of a node change, this node has to select a father again, since it has an impact on the construction of the routing tree.

Advantages and limitations

The advantage of the algorithm proposed is that it is easy to implement. On the other hand, every time a node enters the network, it can modify the level of interference of the rest of the nodes and therefore, tree reconstruction may be needed. This can lead to the change of interference values of other nodes, which again leads to reconstruction of the tree that may eventually lead to infinite loops. Thus, this algorithm needs a method of handling such occurrences.

3.3 Prior research

The algorithms analyzed above have recently been developed and focused on concurrent transmission and throughput enhancement that are expected to be offered from all schemes introduced. However, there are other schemes, earlier developed, like the Destination- Sequenced Distance Vector (DSDV) protocol, the Ad-hoc On-Demand Distance Vector (AODV) protocol and the Dynamic Source Routing (DSR) protocol, which are being used in wireless mesh networks and in wireless ad-hoc networks as well.

3.3.1 Destination- Sequenced Distance Vector (DSDV) protocol

DSDV is one of the earliest routing protocols, developed by Perkins & Bhagwat in 1994. It introduced sequence numbers in order to ensure that routing is performed correctly and loops are prevented from occurring.

The algorithm

Sequence numbers are assigned as follows; the target node generates an even number when a link between nodes exists or an odd number if it doesn’t. These values are then entered in the routing table and routing information is distributed among nodes. When the routing node receives new sequence numbers, it uses the last one received and checks if it is equal to the last entry in the table. As a result, the route used is the one with the better metric. Entries that haven’t been updated within a time limit are deleted from the routing table.

Advantages and limitations

Although DSDV was one of the first algorithms developed, actually it has never been used in commercial applications. However, it has been thoroughly investigated, while few improvements and updates have been introduced. In addition, many algorithms have been inspired by it, such as AODV described in the next sub-section.

One of its drawbacks is that the process of updating its routing table consumes network resources even when the network is stable. Additionally, whenever the network topology changes, new sequence numbers have to be assigned. Thus, it is not suitable to be used in highly dynamic networks.

3.3.2 Ad-hoc On-Demand Distance Vector (AODV) protocol

AODV was first introduced by Perkins & Royer in 1999 and enables both unicast and multicast transmission. It is quite commonly used in mobile mesh networks and was inspired by DSDV. Therefore, like the DSDV, it also uses sequence numbers to exchange information on routes. In addition, it implements a mechanism based on requests and replies for route detection, storing only the best next hop of a node and not the entire route.

The algorithm

In AODV, there isn’t any action until a node needs a connection. At that point, it broadcasts a message requesting the connection. The other nodes forward this message to the rest, and at the same time they reply back. The node requesting the connection receives the replies and selects the route with the least number of hops.

Advantages and limitations

The protocol is supposed to be scalable, while routes are established dynamically, when there is demand from a node. On the other hand, its main disadvantage is that, once the sequence number of the source node is not updated in time and the intermediate nodes of a path have a wrong destination sequence number, intermittent paths are formed.

3.3.3 Dynamic Source Routing (DSR) protocol

The DSR protocol was introduced by Johnson [1994] and it is similar to the AODV. It also sets routes on demand, but in contrast to the AODV, it stores routes in a route cache memory. Moreover, it allows source nodes to specify the route of a message.

The algorithm

The protocol allows multiple routes to any destination and enables each source node to specify the routes used for its transmission. In order to determine these routes during route discovery, the addresses of each node connecting the transmitter and the receiver are collected and the paths on which data are sent are structured. This information is then cached and maintained by all nodes.

Advantages and limitations

DSR doesn’t need to periodically update its routing tables, eliminating the overhead produced. In addition, a route is established only upon demand from an entering node. The main disadvantage of DSR is that it doesn’t handle broken links and doesn’t specify how routes containing those links are managed.

Table 3-1 *Summary of literature review*

Routing schemes				
Authors	**Year**	**Approach**	**Advantages**	**Limitations**
Shetiya & Sharma	2005	To provide QoS	Fixed routing Qos guarantees can be provided	Link failure is not handled Availability of resources needed
Wei *et al*.	2005	Performs interference - aware routing	Higher throughput Higher spectral efficiency	The metric used does not give the complete picture of the interference within the network
Jin *et al*.	2007	Extend the idea of Wei *et al*. Maximize throughput by maximizing concurrent transmissions	Traffic characteristics are taken into account The metric provides a better view of interference within the network	Several tree reconfigurations lead to extra overhead
Tao *et al*.	2005	Minimizing link interference	Easy to implement	The process of a node entering the network may lead to infinite looping

Perkins & Bhagwat	1994	Introduces sequence numbers	Routing is performed correctly Loops are prevented	Unnecessary updates of routing table Not suitable for dynamic networks
Perkins & Royer	1999	New mechanism for route detection Stores only the best next hop of a node and not the entire route	Scalable Routes are established on demand	Can have inconsistent routes
Johnson	1994	It allows source nodes to specify the route Stores the complete route	No need periodic update of routing tables Routes are established on demand	Broken links are not handled

3.4 Summary

In this chapter, seven algorithms in total have been presented. Four of them are recent, while the other three are former schemes. The first one, presented by Shetiya & Sharma [2005], is a centralized fixed routing algorithm for supporting QoS. The idea was to design a scheme that would always provide the same route under all channel conditions. The second one introduced by Wei et al. [2005] was a centralized scheme that is interference-aware and considers traffic load demand at the same time, enabling simultaneous communication thus offering high throughput and expandability of the system. The third proposal by Jin et al. [2007] tried to extend the second one by taking into account the number of packets existing in a blocked node, trying to maximize throughput in this way. Finally, the algorithm proposed by Tao *et al.* [2005] has been presented. Its aim was to allow concurrent transmissions of subscribers' stations in both uplink and downlink, offering high capacity.

In the second part of the chapter, other schemes, like the Destination- Sequenced Distance Vector (DSDV) protocol, the Ad-hoc On-Demand Distance Vector (AODV) protocol and the Dynamic Source Routing (DSR) protocol, presented, have been formerly developed and are being used in wireless mesh networks and in wireless ad-hoc networks. DSDV, one of the earliest routing protocols developed by Perkins & Bhagwat in 1994, is a table-driven routing protocol introducing sequence numbers in order to ensure that routing is performed correctly and loops are prevented from occurring. The second scheme, AODV, inspired by DSDV, was first introduced by Perkins & Royer in 1999 and enables both unicast and multicast transmission. It is a quite common protocol for mobile ad-hoc networks that also uses sequence numbers to exchange information on routes. In addition, it implements a mechanism based on requests and replies for route detection, storing only the best next hop of a node and not the entire route. The next protocol named DSR [Johnson, 1994] is similar to the AODV. Specifically, it sets routes on demand, but contrary to the AODV, it stores routes in a route cache memory. Moreover, it allows source nodes to specify the route of a message.

The classical Dijkstra and Ford-Fulkerson algorithms, that have stimulated this research, will be analyzed in sections 5.3 and 6.4 respectively, while in Table 3-1 a summary and an overview of all the algorithms presented in this chapter is provided, outlining the advantages and limitations for each one.

4 Proposed novel routing algorithms and network architecture

4.1 Introduction

The primary goal of this research is the design of novel routing algorithms for WiMAX mesh networks, since we can classify relay-enhanced WiMAX networks in two categories and construct different solution for each of them.

Section 4.2 describes the proposed architecture to support the creation of a WiMAX mesh network incorporating the relay stations introduced by the 802.16j standard.

Section 4.3 provides the reasons behind the choice made between centralised and distributed algorithms.

Section 4.4 briefly outlines the algorithms designed, while section 4.5 presents basic definitions to be used in both cases.

4.2 Proposed WiMAX mesh network architecture

Both the IEEE 802.16-2004 and the 802.16e standards include support for optional mesh topology enabling WiMAX to create mesh networks with nodes distributed in an arbitrary manner. Although the IEEE 802.16j standard does not refer to mesh topology, it introduces the concept of relay stations. The algorithms proposed combine the two concepts of relay stations and mesh networks, in order to create a Wireless Relay Mesh Network (WRMN).

As in WMNs, the WRMNs technology under investigation is based on multi-hop transmissions and aims at offering service to end-users [Chochliouros, 2009b]. In the case of WMNs, routers create the backbone of WMNs and are usually static. On the other hand, in the case of WRMNs, mesh routers correspond to both the wireless Multi-hop Relay Base Stations (MRBSs) and the wireless Multi-hop Relay Stations (MRSs) that are interconnected with each other in order to establish the meshed wireless backhaul. The MRBSs and MRSs are basic components of the WRMN structure.

The architecture of WRMNs is composed of: a) a MRBS that is the source of the transmission and is connected to other networks or the Internet, b) the MRSs that are retransmitting nodes and c) the end-user terminals that are the target of the transmission and do not incorporate any routing features. Furthermore, routing decisions are taken exclusively by the MRBS.

In Figure 4-1 a realistic implementation of a small 802.16j network is depicted, where the fractional frequency reuse 1×3×3 pattern is used. This means that one cell is split into three sectors, using three different frequencies – one per sector. Such a deployment would typically comprise two relay stations per sector, using the two frequencies of the other sectors for minimizing interference with the MRBS.

On the other hand, a similar 802.16j network organized in a mesh topology is depicted in Figure 4-2. In such a topology, the frequency reuse pattern could be, for example, 1×1×1 and the MRBS would then use frequency 1, while the population of relay stations would increase exponentially transmitting in any frequency - except from the one used by the MRBS, for example in frequency 2.

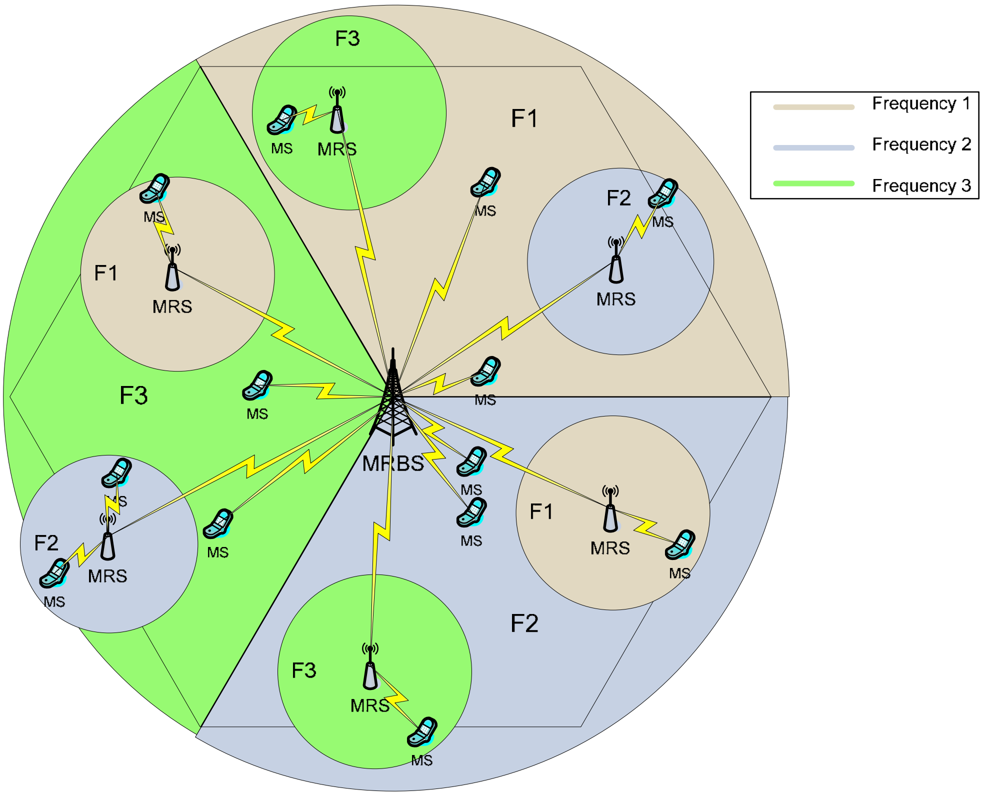


Figure 4-1 *A small WiMAX network*

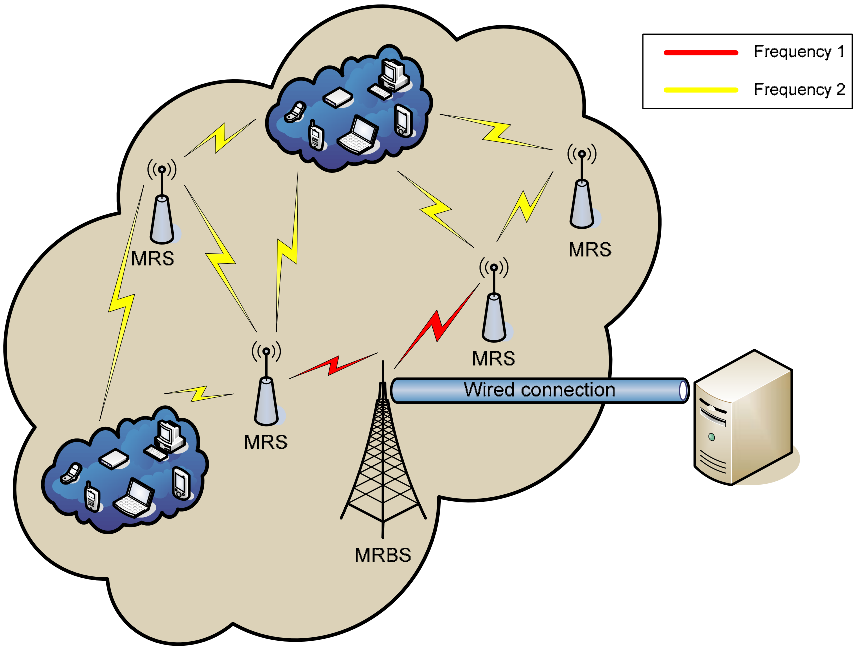


Figure 4-2 *A small WRMN*

The extension to the design above including more than one mesh networks connected to each other is depicted in Figure 4-3. A link can be established between two end-user terminals, each of them belonging either in the same mesh or in a different one. It is clearly portrayed that transmission either within one mesh or even more within two different meshes can be completed with more than one route. Consequently, this operation requires complex but effective routing algorithms.

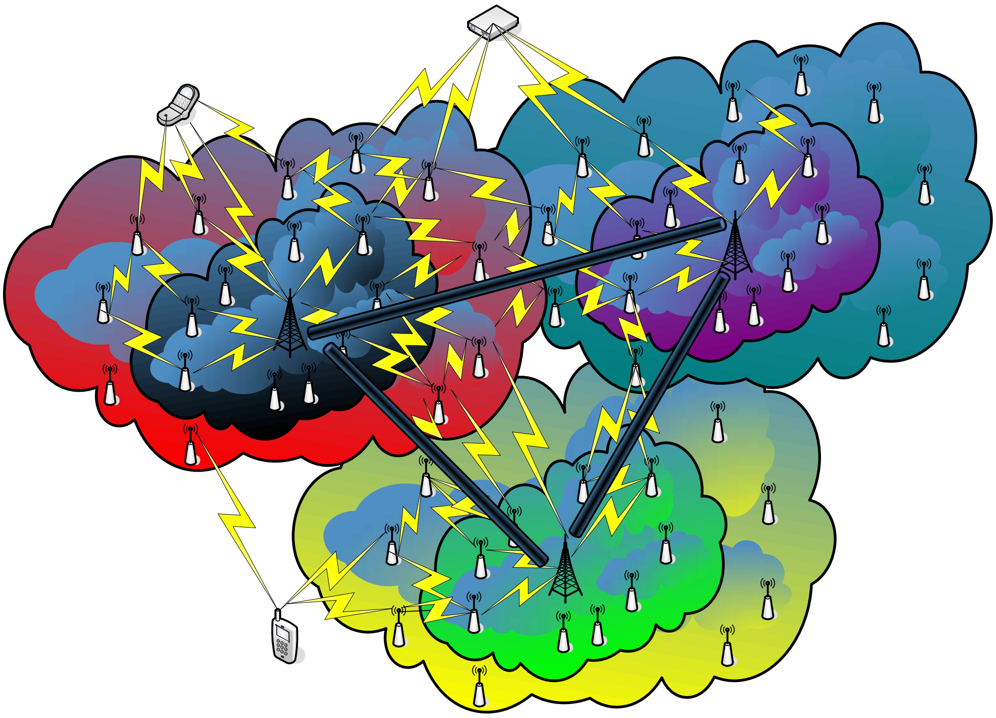


Figure 4-3 *Interconnected WRMNs*

Mesh networks are very sophisticated to manage and pose interference challenges when operating in wireless mode. The routing algorithms have to be carefully designed, because they affect the QoS for different users as well as the load balancing and scheduling functionality of the network.

The throughput capacity of the MRBSs backhaul is the main factor that limits the scalability of such a system and suppresses the number of end-users connected to the network. The most effective solution would be to maximize the throughput capacity, getting it to the theoretical limit.

4.3 Centralised and distributed algorithms

All the decisions regarding routing have to be taken into a time frame of a packet, a session or a time unit. In addition, the demand for high-speed data rates strengthens the point that routing algorithms have to be not only efficient, but also fast and easily implemented with a minimum request for resources and computing power. There are two implementation methods for routing algorithms:

- *Distributed;* every node obtains data regarding the network's topology and its traffic from its connected neighbouring nodes. These data are then used to determine the path where the traffic is forwarded.
- *Centralized;* the main advantage is simplicity. However, the main disadvantage is that, as the network grows, the centralized node gets overloaded and can cause great delays or even crash. The node, which is the source of the transmission, monitors all the available resources and takes all routing decisions.

The IEEE 802.16 Mesh mode MAC supports both centralized and distributed scheduling. Since routing is performed within a scheduler, it can be implemented in distributed and centralized manner as well [Chochliouros *et al.*, 2009a]. The algorithms presented in this thesis implement a centralized mesh design, in which the MRBS is responsible for distributing resources for the network management. The following reasons determine the implementation of centralized algorithms:

- The network topology doesn’t change very quickly in the scenarios under investigation.
- By having centralized and localized Radio Resource Management (RRM) strategy, load sharing among MRSs can be maintained, controlling the level of interference that each MRS or mobile user face in the network.
- The WiMAX standard defines that MRBSs have all the necessary information to take the appropriate decisions (scheduling, power control, load balancing etc).
- Due to the standard, channel knowledge and link states are updated frequently, taking into consideration the needs of the given subscriber and, at the same time, the service loads experienced by each available transmitter.
- In the systems under investigation there is a need to have centralized control, so as to verify that the system is working correctly.
- Although distributed algorithms are better for energy efficient routing, there isn’t a power issue in the scenarios tested.
- The ideas presented and studied are new; therefore, it is easier to begin researching with a centralized scheme to test them.
- With the centralized scheme it is not required to overflow the network with all the necessary information in order for every node to receive them; besides, MRSs don’t have the full functionality of MRBSs.

4.4 Introducing the concepts of a WiMAX mesh network

The algorithms need to incorporate dynamic graph characteristics, so as to model varying link loads in a routing algorithm. The efficiency of the designed algorithms has an impact on the latency, the traffic and the load or congestion in the network. Hence, the design of efficient graph algorithms is of paramount importance.

In the IEEE 802.16j-2009 standard, it is defined that the number of hops for the transparent relay mode is at most two and for the non-transparent relay mode is greater or equal than two. On the other hand, the current frame structure supports only two hops. Therefore, this research has designed the algorithms as if the frame structure supported more than two hops in both modes, which is certain to be implemented in future versions of the standard.

4.4.1 Requirements analysis

At first, let's consider a mobile which is not yet connected to the network, or one that has drifted away from its connecting base. This subsection outlines a scheme of assigning this mobile to a nearby sector. Given that SINR is defined as:

$$\text{SINR} = (\text{desired signal}) / (\text{interference} + \text{noise}),$$

any nearby sector whose closest transmitter (base or relay) provides received SINR beyond a minimal threshold (typically around 0 dB) is a candidate to serve that user.

The potential impact of adding the considered user to each of the neighbouring sectors is being tested. Specifically, based on current channels, scheduling requirements and throughput (or weighted sum rate), optimization takes place at each candidate sector, under the supposition that the considered user has joined this sector. Once some tests are performed, the sectors, which are considered candidates, are those that can admit the new user while still providing sufficient service to the previously existing ones. These tests determine the data rates at each of the candidate sectors. Finally, for each of them, the throughput increase caused by the added user is assessed. The user is then assigned to the base or relay station for which the throughput increase is maximal.

The main graph related algorithms proposed by this research are mostly too computationally intensive in terms of CPU power, memory requirements etc to be carried out online. Thus, in a realistic implementation, most computations will be done offline. Accordingly, the data rates of all a) bases to relays links and b) relays to relays links are computed and updated offline. Likewise, data rates of base and relays transmitted to existing users are always updated offline.

When a new user is about to join a sector, the rates of the sector’s transmitters with respect to that user, are being assessed along with the impact it might have on the previously computed rates. In order for the assessment to take place, an optimization, taking into consideration the previous state of the network, occurs calculating only the difference that the new user will impose. This scheme is not computationally intensive, thus it can work in an online mode supporting the operation of the network.

4.4.2 Network model, assumptions and definitions

The following graph modelling, which originates from graph theory, will provide the required platform for the optimization algorithms introduced.

If one WiMAX sector in a mesh network is considered, as it is depicted in Figure 4-4, the set of the sector's transceivers (base and relay stations) is written by:

$$\mathbf{T} = \{TS_1, \ldots, TS_n\}, \text{ where } TS_1 \text{ is the base station.}$$

The set of currently served mobile stations is denoted by:

$$\mathbf{M} = \{MS_1, \ldots\ldots\ldots, MS_d\}.$$

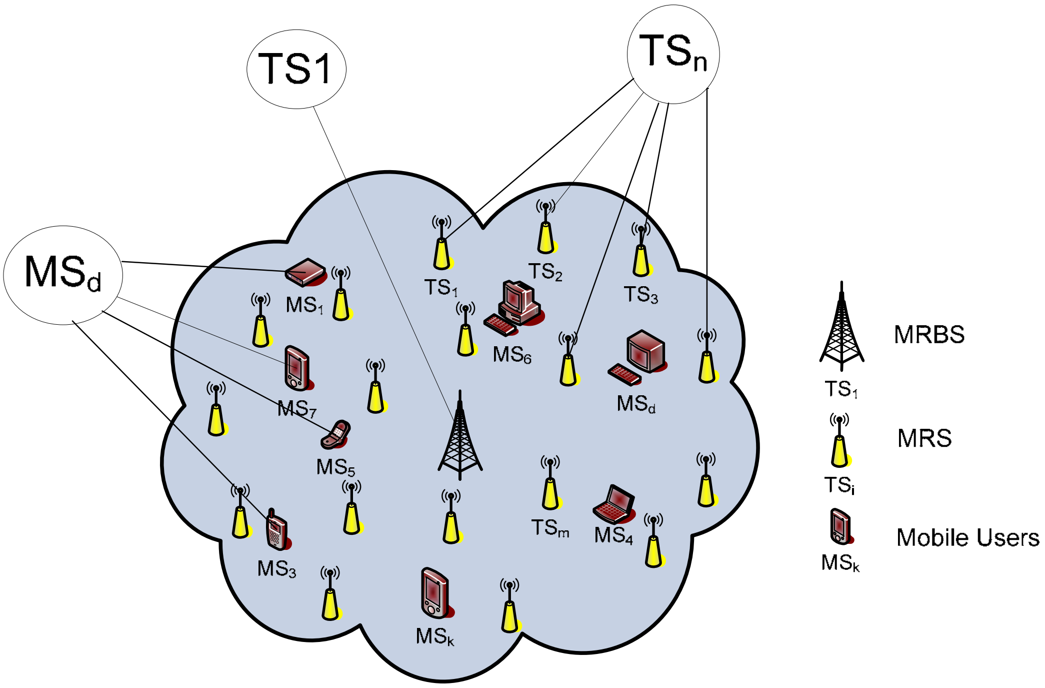


Figure 4-4 *Network model*

The set of all the current cell's wireless units is:

$$\mathbf{V} \equiv \mathbf{T} \cup \mathbf{M}.$$

R_{min} is a system constant signifying the minimal rate per active link.

A directed weighted graph representing the downlink (DL) network is defined as:

$$\mathbf{G = (V, E)},$$

for $v,u \in V$, $(v,u) \in E$ and if v is able to perform DL transmission to u at a rate $R \equiv R(v,u) \geq R_{min}$.

It is assumed that a mobile station can only receive messages and the base or relay station can only transmit to them. This is represented as:

$$(v,u) \in E \rightarrow v \in \mathbf{T} \text{ and } u \neq TS_1.$$

This research deals with routing algorithms in two types of mesh networks. In the first type, called NICI (No Intra-Cell Interference), no mutual intra-cell downlink-interference is allowed. The NICI setting is suitable for deployments in which intra-cell interference is too harmful for the signal quality. Such requirement is typical to open environments like rural areas. In the NICI scheme and in each given cell, only one transceiver (base, relay or mobile) can transmit at a given time and frequency.

The second model is called LICI (Limited Intra-Cell Interference) and is suitable at urban environments where the relays are mainly below roof-top. Its description is more elaborate and it is based on the assumption that each sector is split into several mutually disjoint subsets (tiers) $T_1,...,T_K$ where:

$$T = T_1 \cup ...... \cup T_K,$$

such that the mutual intra-cell interference is low within each tier T_i, typically due to below rooftop relay deployment.

The core assumption is that, at any given DL time and frequency, there is exactly one tier T_i whose transmitters can transmit data simultaneously at an assigned time slot. For each tier T_i, there is a time assignment $0< t_i <1$ that signifies a constant ratio of the transmission time which is allocated to that tier. It thus holds for all tiers that the total duration of transmission equals to one time unit:

$$t_1 + \dots\dots + t_K = 1.$$

Whenever $TS_j \in T_i$, it is defined that $T(TS_j) = i$. This means that $T(TS_j)$ is the index of the tier to which T_j belongs. For our convenience, it is assumed that $T_1 = \{TS_1\}$, meaning that T_1 is the base station. It is also supposed that a transmitter at a given tier T_i transmits either to a relay of the next tier T_{i+1} or to a mobile end user MS_d. At each tier the transmission time is divided between transmission to relays of the next tier and transmission to mobiles. Thus:

$$t_i = t_i' + t_i'',$$

where t_i' is a fixed time ratio assigned to transmission to relays and t_i'' is a fixed time ratio assigned to transmission to mobiles. It still holds that:

$$\Sigma_{1 \le i \le k}\, t_i = \Sigma_{1 \le i \le k}\, (t_i' + t_i'') = 1.$$

This means that when the total duration of transmission equals to one time unit, the communication of the relays to other relays and to end-users must be accomplished within this time frame.

4.5 Summary

In this chapter two types of relay-enhanced WiMAX networks have been identified and the proposed architecture of Wireless Relay Mesh Networks (WRMNs) incorporating the relay stations introduced by the 802.16j standard is described. The WRMNs technology under investigation is based on multi-hop transmissions and aims at offering service to end-users. The MRBSs and the MRSs are basic components of the WRMN structure and are interconnected with each other in order to establish the meshed wireless backhaul. Additionally, a MRBS is the source of a transmission which takes all necessary routing decisions, while the MRSs are the retransmitting nodes. Finally, the subscribers' stations are the target of the transmission.

In section 4.3 it is explained why centralised algorithms have been selected over distributed ones. More explicitly, one of the main reasons is that all the decisions regarding routing have to be taken within a time frame of a packet, a session or a time unit. Therefore, routing algorithms have to be not only efficient, but also fast and easily implemented with a minimum request for resources and computing power. Since the network topology doesn't change very quickly and the WiMAX standard defines that the MRBSs have already got all the necessary information to take the appropriate decisions (scheduling, power control, load balancing etc), it is not required to overflow the network with unnecessary traffic. Finally, although distributed algorithms are better for energy efficient routing, there isn't a power issue in the scenarios tested.

The algorithms designed for WRMNs need to incorporate dynamic graph characteristics, so as to model varying link loads in a routing algorithm. The efficiency of the designed algorithms impacts on the latency, the traffic and the load or congestion in the network. Section 4.4 has briefly outlined the concept of the algorithms designed. At first, a mobile which was not yet connected to the network has been considered. In order to assign this to a nearby sector, the potential impact of adding it to each of the neighbouring sectors is tested. The sectors considered candidates are those that can admit the new user while still providing sufficient service to the previously existing ones. Finally, for each of them, the throughput increase caused by the new user is assessed. The user is then assigned to the base or relay station for which the throughput increase is maximal.

Finally, graph modelling, which originates from graph theory, has provided the required platform for the optimization algorithms designed. The main algorithms proposed in this research are too computationally intensive to be carried out online. Thus, in a realistic implementation, most computations will be done offline. However, when a new user is about to join a sector, the rates of the sector's transmitters with respect to that user, are being assessed along with the impact it might have on the previously computed rates. This process is not computationally intensive; consequently, it can work in an online mode supporting the operation of the network.

5 New routing algorithm proposed based on Dijkstra algorithm

5.1 Introduction

For the algorithm proposed, no mutual intra-cell interference (NICI) is allowed. Moreover, in each given cell, only one transceiver (base, relay or mobile) can transmit at a given time and frequency. The NICI setting is suitable in deployments where intra-cell interference is too harmful for the signal quality, so it is typical to open environments such as rural areas, like that in Figure 5-1, where the deployment is usually on a top of a mountain, a hill or the tallest structure in the area. Therefore, the deployment can costly enough for the equipment to be installed; especially in areas where there is no electricity or road access and a lot of power is required for the transmission.

Figure 5-1 *NICI environments – rural areas*

In the following section, both the network topology and the transmission method are described. Section 5.3 presents the standard Dijkstra algorithm on which the proposed one is based. Section 5.4 analyzes the utilization of Dijkstra's algorithm for unicast transmission in NICI settings, while section 5.5 investigates the implementation of a new algorithm addressing the issue of unicast transmission in a LICI environment. Section 5.6 presents a simulation example for the validation of the designed scheme.

5.2 Conditions and assumptions

The WRMN topology used for studying the algorithm designed is depicted in Figure 5-2. A MRBS is the source of the transmission with several MRSs distributed randomly within its range, while backhaul links are established among the MRSs and the MRBS. For the access level, there is the assumption that the MRBS routes data to the end users through the MRSs.

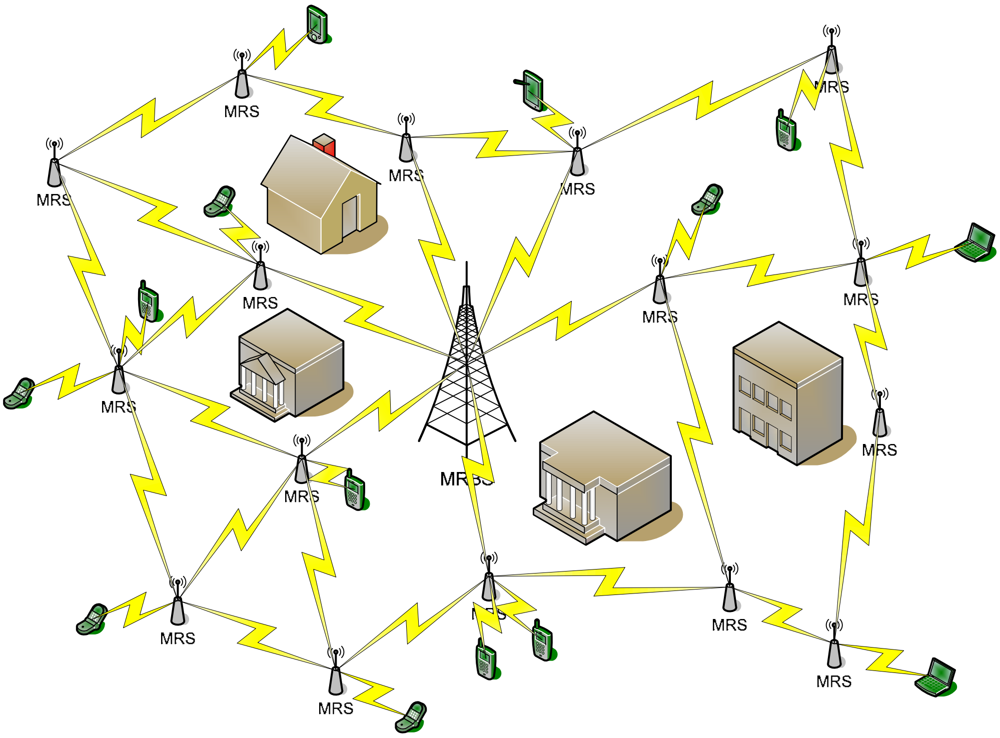


Figure 5-2 *Network topology assumption*

The routing scheme selected for the implementation of the algorithm is unicast, as shown in Figure 5-3. This means that at every moment only one end user can be accommodated while the rest are waiting for their turn.

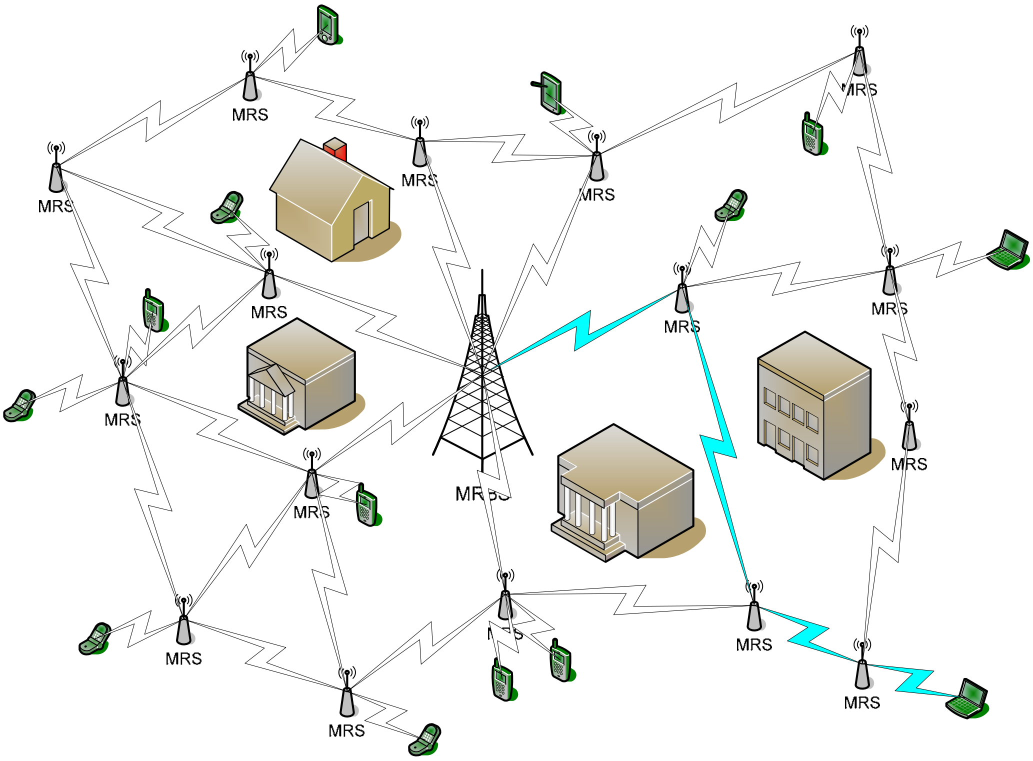


Figure 5-3 *Unicast transmission*

5.3 Description of the standard Dijkstra algorithm

The Dijkstra algorithm [Dijkstra, 1949, Cormen *et al.*, 2001] operates in a very simple manner. In more detail, a node is aware only of its direct neighbours and the weight/cost to contact them. Each node, on a regular basis, sends to every neighbour a list containing the necessary costs to reach all its direct destinations. The neighbouring nodes receive this list and compare it to their own; any record which represents an improvement of their records is inserted in their lists. In the end, all nodes in the network have created a list with the best next hop for all destinations, and the best total cost for every route.

In the case where a node or a link fails, the rest of the nodes discard the entries containing that node and create a new list. Then they pass it on to all their neighbours, who replicate the process. Eventually, all the nodes in the network update their lists. The Dijkstra algorithm goes through these steps shown in Figure 5-4 as a flowchart:

1. The MRBS first constructs a graph of the network and distinguish between transmitting and receiving nodes.
2. Then it creates a matrix where it indicates weights. In the case where a link connecting the two nodes doesn't exist, "infinity" is defined as a weight.
3. The MRBS initializes the list values of all direct links connecting nodes and assigns "infinity" as their weight/cost.
4. The MRBS selects a node as the source, apart from itself which is the first entry in the list.
5. The MRBS checks all the nodes that are connected directly to the source and opts for the one whose link cost is minimal. Afterwards, that node is selected as the target one and the list values are updated.
6. If that node is the last of a path, the algorithm terminates, otherwise the target node becomes the next source node.
7. The MRBS repeats the steps 3-5 until a routing table containing all reachable destinations is created. This way, all paths are created.

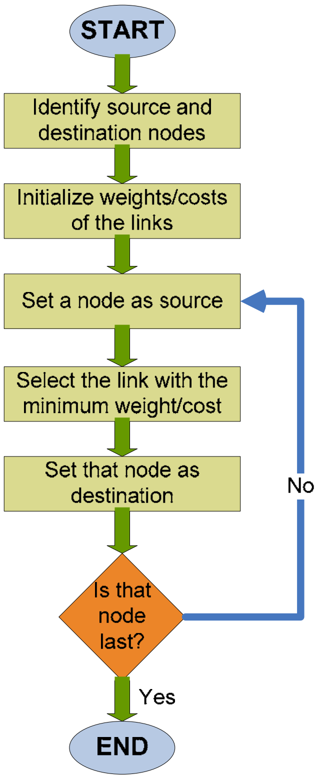


Figure 5-4 *Dijkstra algorithm flowchart*

5.4 Enabling utilization of Dijkstra algorithm for unicast transmission in NICI networks

In the case of a NICI setting, when a user enters the network, for each neighbouring candidate sector, a relay route that maximizes the overall relay-based rate, considering that all candidates sectors, if admitting this additional user, will not disconnect service for any of the existing users, is found. Among these candidates the one that entails the greatest rate to the considered user is chosen.

In order to perform rate maximization for NICI networks, this process must be done independently for each user. It means that, when a new user is considered for a sector, only this user is involved in the optimization process which yields a relatively moderate-complexity algorithm.

The method is based on the following primary relay rule. Let us consider a base station transmitting to a mobile station MS through relay stations MRS_1,.....,MRS_{k-1} according to their order. More explicitly, each message first goes from the MRBS to MRS_1 at rate R_1 and then from MRS_1 to MRS_2 at rate R_2 and so on, until it goes eventually from MRS_{k-2} to MRS_{k-1} at rate R_{k-1} and then finally from MRS_{k-1} to the MS at rate R_k. An example with a relay route containing three MRSs is shown in Figure 5-5, where a message first goes from the MRBS to MRS_1 at rate R_1, then from MRS_1 to MRS_2 at rate R_2, from MRS_2 to MRS_3 at rate R_3 and then finally from MRS_3 to MS at rate R_4.

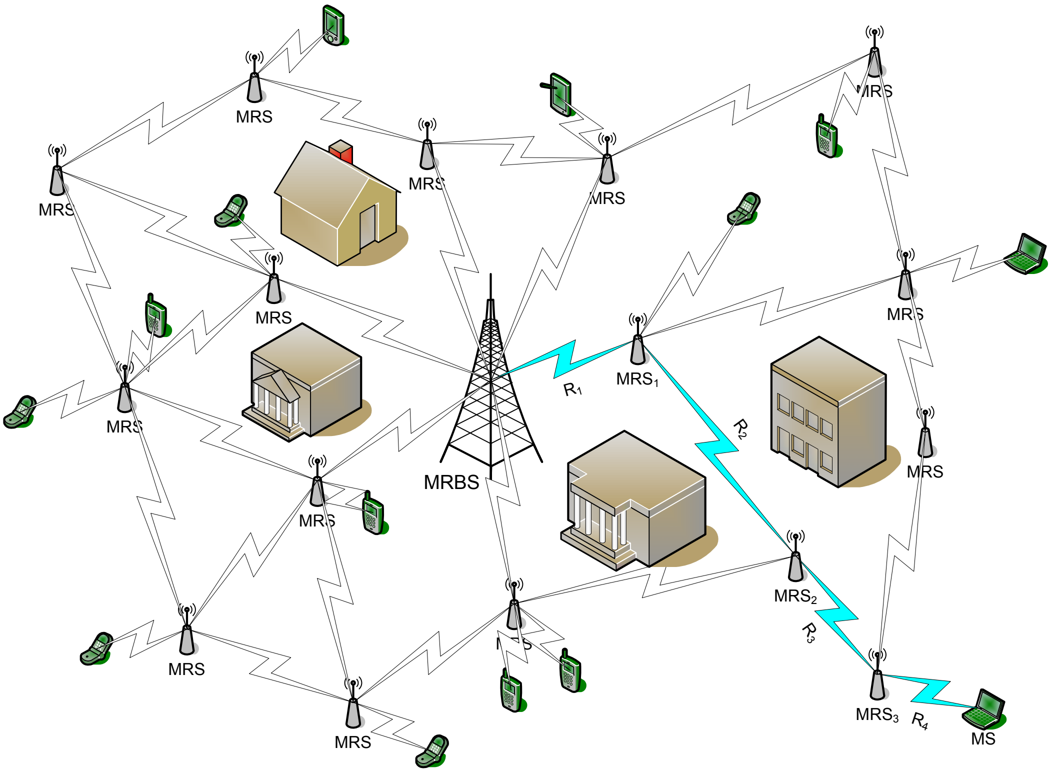


Figure 5-5 *Example of a relay route*

Due to the assumptions of a NICI network there is only one active transmitter at any given time. To clarify the idea, it can be supposed without loss of generality that it uses all the available bandwidth. It will be shown further down that the composite (normalized) rate R, based on dynamic, optimized time sharing, is given by the formula:

$$1/R = 1/R_1 + 1/R_2 + \ldots\ldots\ldots + 1/R_k \quad (1)$$

This optimized transmission configuration can be denoted as:

$$MRBS \rightarrow MRS_1 \rightarrow MRS_2 \ldots\ldots\ldots \rightarrow MRS_{k-1} \rightarrow MS.$$

The additive property expressed by (1), enables utilization of the Dijkstra algorithm in order to maximize the composite rate of each subscriber. The implication is that the bandwidth allocated for a given user and the overall data rate are minimized.

The first step toward the rate's maximization algorithm is the transformation of our directed graph representation G = (V, E) into a weighted graph. This is done by assigning to each edge in the graph $(v,u) \in E$ a weight given by:

$$w(v,u) = 1 \ / \ R(v,u),$$

which is the inverse of the rate of transmission from v to u. The reason behind this is the utilization of Dijkstra's algorithm, which finds the minimum cost. By inversing a variable and by finding the minimal of this inverse, the actual maximum of the variable is calculated.

If the network presented as an example in Figure 5-5 is used and the link values are updated as discussed, then the new link values of the network will be those shown in Figure 5-6. More explicitly, a message first goes from the MRBS to MRS1 at inverse rate 1 / R1, then from MRS1 to MRS2 at inverse rate 1 / R2, from MRS2 to MRS3 at inverse rate 1 / R3 and then finally from MRS3 to MS at inverse rate 1 / R4.

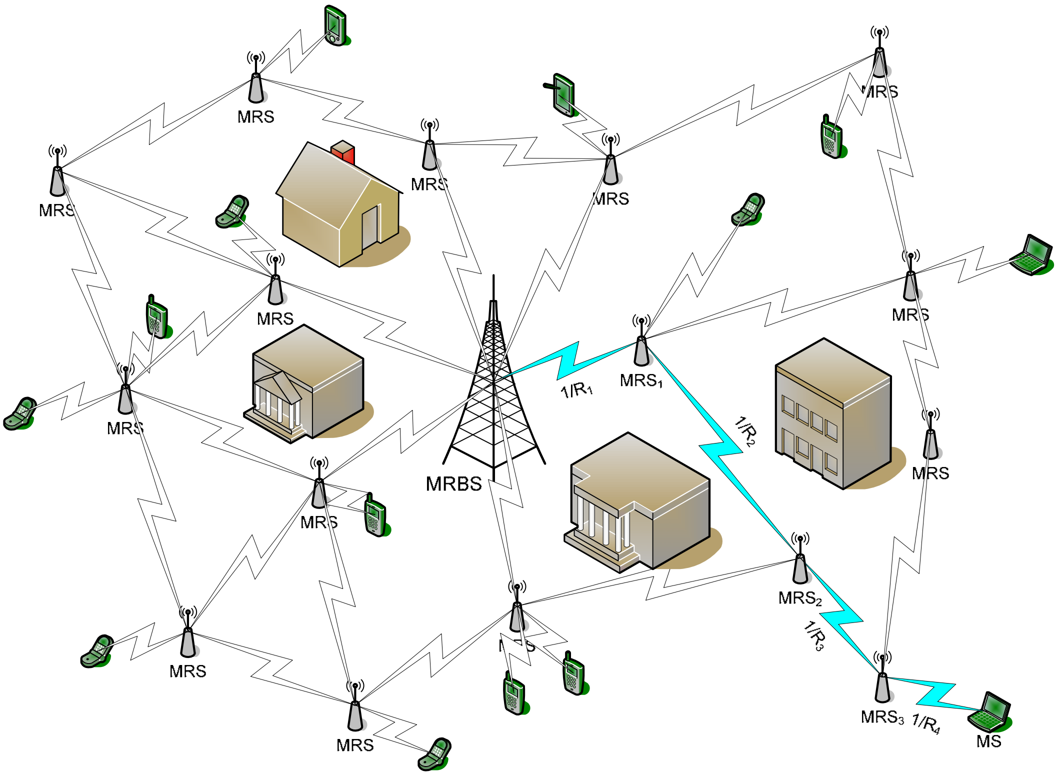


Figure 5-6 *Example of a weighted relay route using the inverse rates*

The Dijkstra minimal-weighted-path-algorithm finds for each subscriber $u \in M$ a route of relays $v_1, \ldots\ldots, v_{k-1}$ for which the composite rate of the transmission

$$MRBS \rightarrow v_1 \rightarrow v_2 \ldots\ldots \rightarrow v_{k-1} \rightarrow u,$$

which is provided by the formula:

$$1/R = 1/R_1 + 1/R_2 + \ldots\ldots\ldots + 1/R_{k,}$$

is minimized. In the case where k=1, it means that the base transmits directly to the mobile.

In other words, it finds the transmission path for which the composite rate R is maximized. In practice, the mean rate of the transmission between the cell's transceivers – base station and relays - (R_i, $1 \leq i < k$) can be computed offline and stored in memory.

The complexity of this algorithm is $O(|V|^2 + |E|)$.

5.4.1 Proof of the fundamental relay-transmission formula: $1/R=1/R_1+...+1/R_k$

Theorem 1: Let us consider a base transmitting to a mobile station MS through relay stations MRS_1, ..., MRS_{k-1} where each message first goes from the MRBS to MRS_1 at rate R_1 and then it goes from MRS_1 to MRS_2 at rate R_2 and so on until it eventually goes from MRS_{k-1} to MS at rate R_k where only one transmitter works at each given time. Then it holds that the composite rate R based on optimized time sharing is given by:

$$1/R= 1/R_1 + 1/R_2 + \dots\dots + 1/R_k \qquad (1)$$

Proof: Our proof is split into 2 parts, the first one proves the case where k=2, which is the induction step, providing the essence of the second part where the general case for k is proven.

1. k=2. In what follows the rate of a 2-hop relay-aided connection with optimized time share for MRBS → MRS & MRS → MS communication is derived. When the base to relay rate is R_1 and the relay to mobile rate is R_2 then the composite rate based on dynamic, optimized time sharing is given by:

$$R = R_1R_2 / (R_1 + R_2)$$

that is:

$$1/R = 1/R_1 + 1/R_2$$

Proof for k=2: The time dedicated to MRBS → MRS is denoted by t_1 and the time dedicated to MRS → MS is denoted by t_2. It can be assumed without loss of generality that:

$$t_1 + t_2 = 1 \qquad (a),$$

i.e. 1 time unit for both transmissions. The optimal composite rate is achieved if:

$$t_1R_1 = t_2R_2\ (\equiv R) \qquad \text{(b)},$$

meaning that there is equal data volume on both links. With (a) it follows:

$$t_1 / t_2 = R_2 / R_1 \Rightarrow$$

$$(1 - t_2) / t_2 = R_2 / R_1 \Rightarrow$$

$$t_2 = R_1 / (R_1 + R_2) \qquad \text{(c)}$$

Thus, from (b) and (c) it derives:

$$R = t_2R_2 = R_2 \cdot (R_1 / (R_1 + R_2))$$

2. The induction step. Suppose now that this assertion holds for k≥2. It will be proven inductively that it also holds for k+1. It is supposed that the base station transmits to a mobile station MS through relay stations MRS_1, …, MRS_{k-1}. The induction assumption says that, when the time distribution of the first k-1 hops is optimized, it holds that the composite rate R' of the base transmitting to MRS_{k-1} is given by:

$$1/R' = 1/R_1 + 1/R_2 + \ldots\ldots + 1/R_{k-1}.$$

Thereby, according to part 1 of the proof, if ρ is the ratio between (a) the time dedicated to the k-1-hops-relay-aided transmission from the base to MRS_{k-1} and (b) the time dedicated to the transmission from MRS_{k-1} to MS, ρ is given by:

$$\rho = R' / R_k.$$

Due to the proof of the case k = 2, the resulting composite rate is given by:

$$1/R = 1/R' + 1/R_k = (1/R_1 + 1/R_2 + \ldots\ldots + 1/R_{k-1}) + 1/R_k.$$

5.5 Optimization of a LICI unicast network via sum-min-max algorithms

The goal of this section is the optimization of unicast transmission in a LICI network. In the unicast network under consideration each user is served at an exclusive time and frequency resource, however the network transceivers can, in part, work simultaneously. The network transceivers are divided into several tiers, where within each tier the interference is low enough to allow simultaneous transmission. Yet, two transceivers of the same tier cannot transmit at the same time and frequency resource. Such a scheme is suitable for an urban environment like that presented in Figure 5-7, where the relays are deployed mainly below roof-top.

Figure 5-7 *LICI environments - Urban areas*

A central element in our approach is that during the initiation of the network an offline-algorithm optimizes the entire network's backhaul. Once this infrastructure algorithm is complete, updates occur only when backhaul links change, which is not very often.

Once backhaul optimization is established offline, dynamic frequent updates occur only with respect to the end users. It is essential to notice that these updates amount to very light-weight executions of the Dijkstra algorithm in which an equivalent of a 2-hop optimization problem is solved. This will be called here as the delta end-user optimization.

5.5.1 The dynamic delta end-user optimization

The delta problem focuses on one mobile user and consists of a MRBS and a set of MRSs, MRS1,....,MRSn. The respective composite rates of the MRBS DL data transmission to the MRSs are denoted respectively by R1, ..., Rn. These rates can be calculated by the ensuing offline backhaul unicast optimization algorithm.

Each end user MS has at a given time DL data rates transmitted from every MRS, R_1', ..., R_n' respectively, and a DL data rate transmitted from the MRBS R_0'. Let:

$$\rho_0 = 1 / R_0',$$

$$\rho_I = 1 / R_i' + 1 / R_i,\ i=1, \ldots, n.$$

The decision on the best link in this case requires only finding the minimum of $\{\rho_0, \rho_1, \ldots, \rho_n\}$.

If ρ_k denotes this minimum, then the maximal rate is given by:

$$R_{composite,k} = 1/\rho_k .$$

If k≥1 it means that the user is served via the k MRS at a composite rate of $R_{composite,k}$. If k = 0, the user is served directly by the base at the composite rate of $R_{composite,0}$.

5.5.2 Unicast network with tiers of simultaneous transmission

Let's consider that the set of transceivers T, of the cell under consideration, is split into several mutually disjoint subsets (tiers) $T_1,\ldots,T_k$, where:

$$T=T_1\cup\ldots\ldots\cup T_K,$$

such that the mutual intra-cell interference is low within each tier T_i, typically, due to the distance and/or the below rooftop MRS deployment. The core assumption is that at any given DL time and frequency there is exactly one tier Ti whose transmitters can transmit data. An example with three tiers is provided in Figure 5-8. Furthermore, all the transmitters of that tier can transmit simultaneously at this assigned time slot.

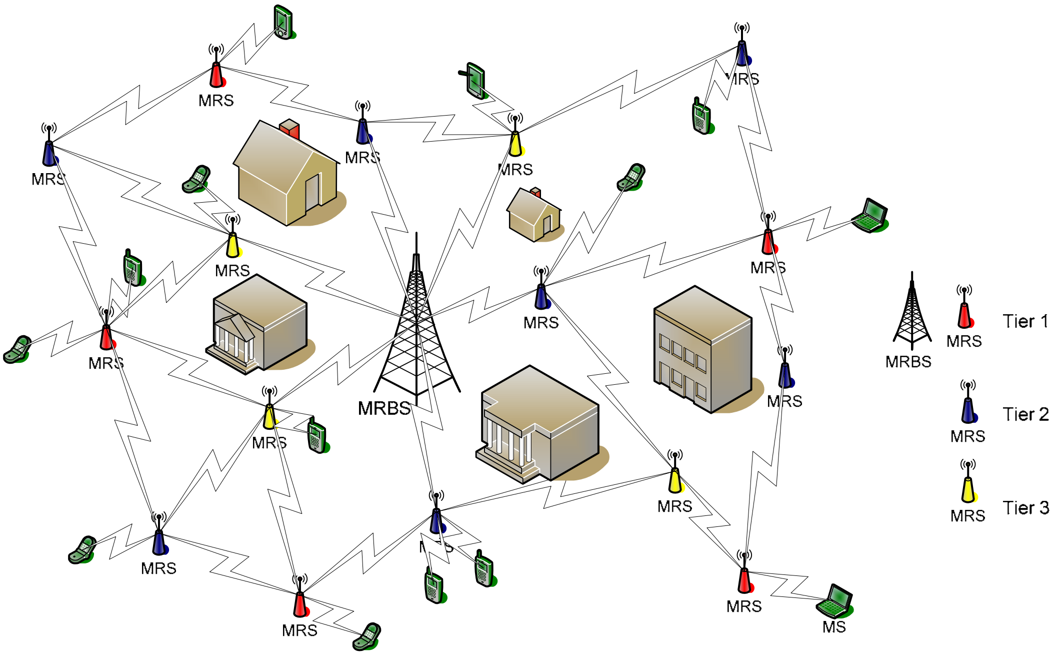


Figure 5-8 *An Example of a LICI network with three tiers of transmitters for DL*

The goal is to identify the maximal composite rate available to each end user and the relay-path that utilizes it. This will be done with a directed weighted graph model, where the definition of path-weight is more general and complex than the classical accumulative path-weight defined in the NICI network and it targets to solve this unicast optimization problem. Once again the path with the minimal weight is to be found.

5.5.3 Unicast network and its related graph model

The graph models of these networks consist of weighted, directed graphs of the form G = (V, E) with ω: E → R^+, a weight function that correspond edges to positive values of weights. The graph vertices correspond to the network transceivers and the weights to the inverse of the respective rate. That is, assign to each edge in the graph $(v,u)\in E$ a weight is assigned, given by:

$$w(v,u) = 1\ /\ R(v,u),$$

which is the inverse of the rate of transmission from v to u. ω is extended to $V\times V$ by setting $\omega(v,w)=\infty$, whenever $(v,w)\in V\times V\backslash E$.

In the above definition, adding the weights of the edges that form a path result to the total of its weight. Here, the set of vertices is split into a few disjoint subsets, V_1, ..., V_K, that correspond to the network tiers. That is,

$$V=\cup_{1\le k\le K}V_k \text{ and } V_i\cap V_j=\emptyset, \text{ for all } 1\le i<j\le K.$$

This model has an underlying assumption that the messages are infinitely long and thus in the interval of time assigned to each of the tiers all the relay-path transceivers that belong to this tier can transmit simultaneously for a long time. Thus, they all have to transmit at the minimal rate of each of their links, which corresponds to the maximal assigned weight.

Consequently, the weight of a path is defined as follows. For a given path in G, p = $<v_0, \ldots, v_n>$, the sub-paths per k-tier are defined, k=1,.....,K, which is the intersection with that tier:

$$p(k) \equiv <v_1, \ldots, v_n> \cap V_k \equiv <v_{i(1,k)}, v_{i(2,k)}, \ldots, v_{i(n_p(k),k)}>$$

$$\text{where } 1 \le i(1,k) < i(2,k) < \ldots < i(n_p(k),k) \le n.$$

In the case where this tier has no representation in this path, then:

$$p(k)=\varnothing, \text{ and } n_p(k)=0.$$

For k=1, ..., K set,

$$\omega_{MAX}(p(k)) = \max\{\omega(v_{i(j,k)}, v_{i(j,k)+1}): 1 \le j \le n(k)\}.$$

In the case where n(k)=0, it is defined: $\omega_{MAX}(p(k)) = 0$. Then, it is defined:

$\omega_{MA}(p) = \Sigma_{1 \le k \le K} \omega_{MAX}(p(k))$ (MA stands for maximum/addition)

A shortest ω_{MA}–path from one source to all vertices or for every pair of vertices has to be found.

5.5.4 Technical assumptions simplifying the graph model

To simplify the presentation detailed, let us denote the vertices of G as V = {1, 2, ..., N}. It can be assumed without loss of generality that any path p under consideration is simple. It is also denoted that:

$$n(k) = \#(V_k),$$

$$N(0) = 0,$$

$$\text{and } N(k) = n(1) + \ldots + n(k-1) + n(k), \text{ for } k=1,\ldots,K.$$

It can be also supposed that:

$$V_k = \{N(k-1) + 1, \ldots, N(k)\}.$$

Define for each n=1,...,N, $\kappa(n)$ to be the unique integer such that $n \in V_{\kappa(n)}$. Finally, it can also be supposed that $n(1) \geq \ldots \geq n(K)$.

5.5.5 Pure min-max solution for K=1

For K=1 an optimal algorithm, which is a straightforward adaptation of the Floyd-Warshall algorithm, that minimizes the path between every two users can be implemented. An equivalent adaptation is possible for Dijkstra. Consequently, the following algorithm is based on the statement that a minimal path can be constructed in such a way that its sub-paths are minimal too.

Let us consider a subset {1, 2, ..., n} of vertices for some n. For any pair of vertices i, j in V ($i, j \in \{1, \ldots, N\}$), consider all routes from i to j whose intermediate vertices belong to {1,....,n}, and let p be a minimum-weight path between them. The algorithm makes use of the relationship between p and minimal weighted paths from i to j with all intermediate vertices in the set {1, 2, ..., n–1}. This relationship depends on whether or not n is an intermediate vertex of path p.

If n is not an intermediate vertex of path p, then all intermediate vertices of path p are in the set {1, 2, ..., n–1}. Thus, a minimal weight path from vertex i to vertex j with all intermediate vertices in the set {1,2,.......,n–1} is also a minimal weighted path from i to j with all intermediate vertices in the set {1, 2, ..., n}.

If n is an intermediate vertex of path p, then p is broken down into a path p_1 from i to n and a path p_2 from n to j. Without loss of generality, it can be supposed that p_1 is a minimal weighted path from i to n with all intermediate vertices in the set {1, 2, ..., n–1}. Similarly, without loss of generality, it can be supposed that p_2 is a minimal weighted path from n to j with all intermediate vertices in the set {1, 2, ..., n–1}.

Consequently, a recursive formulation of the minimal weighted path is defined. For 1≤i, j≤N, let $d^{(n)}_{i,j}$ be the minimal weight of a path from vertex i to vertex j with all intermediate vertices in the set {1,2,...,n} and $p^{(n)}_{i,j}$ be a path from vertex i to vertex j utilizing this minimum. When n = 0, a path from vertex i to vertex j with no intermediate vertex numbered higher than 0, has no intermediate vertices at all. It, thus, has at most one edge, and hence:

$$d^{(0)}_{i,j}=\omega(i,j),\ 1\leq i,j\leq N$$

$$\text{and } p^{(0)}_{i,j}=(i,j),\ 1\leq i,j\leq N.$$

A recursive definition is given by:

$$d^{(n)}_{i,j} = \min(d^{(n-1)}_{i,j}\ ,\ \max(d^{(n-1)}_{i,n}\ ,\ d^{(n)}_{n,j})).$$

The matrix $D=(\ d^{(N)}_{i,j})_{1\leq i,j\leq N}$ gives the final answer for K=1.

Thus, the algorithm for K=1 is as follows:

```
Input: $d^{(0)}_{i,j}=\omega(i,j)$, $p^{(0)}_{i,j}=(i,j)$, $1\leq i,j\leq N$.
    for n=1:N
        for i=1:N
            for j=1:N
```

```
$d^{(n)}_{i,j} = \min(d^{(n-1)}_{i,j} , \max(d^{(n-1)}_{i,n} , d^{(n)}_{n,j}))$
    if $d^{(n)}_{i,j} = d^{(n-1)}_{i,j}$
        then $p^{(n)}_{i,j} = p^{(n-1)}_{i,j}$
    else
        $p^{(n)}_{i,j} = <p^{(n-1)}_{i,n} , p^{(n)}_{n,j}>$
```

The output is: $D=(d^{(N)}_{i,j})_{1\le i,j\le N}$, $P=(p^{(N)}_{i,j})_{1\le i,j\le N}$.

The complexity of the algorithm is N^3 operations to compute D and N^4 to compute P.

5.5.6 Optimal sum-min-max algorithm for minimal path

Let us now consider that the first few tiers have multiple elements and all the rest have one element. It has been mentioned in section 5.5.4 that $n(1)\ge........\ge n(K)$, where $n(k)$ is the number of elements in tier k. Let us assume the case where for some small k_0, $n(k_0+1) = 1$. Due to complexity considerations that will be analyzed below, in practical applications, it is unlikely that k_0 would exceed four.

The goal of this sub-section is an algorithm that finds the minimal weighted path from a fixed source $s\in V$ (this models the MRBS) to each vertex $v\in V$. This minimal sum-min-max path is attained, as the minimum of the few minimal paths are derived from the execution of the standard Dijkstra algorithm.

A key aspect of the algorithm is that the initial backhaul optimization is done offline without the participation of mobile end-users. Each time a mobile user enters or leaves the network or, when the state of a link changes, a very fast standard Dijkstra procedure takes place to update the optimized scheme for the modified parameters.

5.5.6.1 Construction of associated minimal path problem solvable by Dijkstra

Let's define for k = 1, ..., K:

$$E_k = \{(u,v) : u \in V_k \text{ and } v \in V\}$$

$$\text{and } m(k) = \#E_k.$$

E_k is then arranged as a sequence:

$$E_k = \{(u_{i,k}, v_{i,k}),\ i=1, \ldots, m(k)\},$$

with accordance to the edges weights,

$$\omega(u_{1,k}, v_{1,k}) \leq \omega(u_{2,k}, v_{2,k}) \leq \ldots\ldots\ldots \leq \omega(u_{m(k)}, v_{m(k)}).$$

For a positive integer q denote:

$$[q] \equiv \{1, 2, \ldots\ldots., q\}.$$

A key element in the ensuing solution is a weight function parameterized by an integer vector, called t, whose dimension is k_0:

$$t = (t(1), \ldots.., t(k_0)),$$

such that,

$$t(k) \in [m(k)], \text{ for } k=1,\ldots,k_0.$$

In other words, the parameter t is any element in the set

$$M \equiv [m(1)] \times [m(2)] \times \ldots. \times m(k_0)].$$

Next, let

$$E' \equiv \cup_{1\le k\le k_0} E_k.$$

For $t\in M$, ω_t, a weight function on $G=(V,E)$ can be defined by:

$$\omega_t(u,v)=\omega(u,v), \text{ if } (u,v) \in E - E'$$

$$\omega_t(u_{i,k}, v_{i,k}) = 0,\ i\le t(k),\ k\in[k_0]$$

$$\omega_t(u_{i,k}, v_{i,k}) = \infty,\ i> t(k),\ k\in[k_0]$$

5.5.6.2 Sum-min-max algorithm's sketch: minimum of several Dijkstra runs

The sum-min-max algorithm $\forall t\in M$ executes the conventional Dijkstra algorithm on (G, ω_t) and gets $\forall v\in V$ the minimal weighted path:

$$p_t(v) = <v_1, \ldots\ldots, v_q> \text{ (i.e. } v_1 = s,\ v_q = v)$$

and its weight,

$$\omega_t(p_t) = \omega_t(v_1,v_2) + \omega_t(v_2,v_3) + \ldots\ldots\ldots + \omega_t(v_{q-1},v_q).$$

Additionally,

$$\min\{\omega_{MA}(p)\} = \min\{\omega_t(p) : t\in M, \text{ p is a path from s to v}\}.$$

5.5.7 Pseudo-code of the sum-min-max algorithm

1. Input:

```
for k = 1, ...., K: $n(1) \ge \ldots\ldots \ge n(K)$
        $n(k) = \#(V_k)$
        $V_k=\{N(k-1)+1,\ldots,N(k)\}$.
N(0)=0, and $N(k) = n(1) + \ldots + n(k-1) + n(k)$.
```

```
2. Initiation:
    for v = 1, ..., N
        $p(v) = \emptyset$
        $d(v)=\infty$
3. for all $k=1, \dots, k_0$
    sort and index the set $E_k = \{(u,v): u \in V_k \text{ and } v \in V\}$,
    in the form $E_k = \{(u_{i,k},v_{i,k}): l = 1, \dots, m(k)\}$,
    that conforms with $\omega(u_{1,k},v_{1,k}) \leq \omega(u_{2,k}, v_{2,k}) \leq \dots \leq \omega(u_{m(k)},v_{m(k)})$.
4. $\forall t \in M$, and $\forall (u,v) \in E$, insert values to $\omega_t(u,v)$,:
    $\forall (u,v) \in E - E'$
        $\omega_t(u,v) = \omega(u,v)$
        for $k = 1, \dots, k_0$
            for $i=1, \dots, t(k)$
                $\omega_t(u_{i,k}, v_{i,k}) = 0$
                $\omega_t(u_{i,k}, v_{i,k}) = \infty$, $i>t(k)$, $k \in [k_0]$
        for $v=1,\dots N$
            Dijkstra procedure with input $(V, E, \omega_t)$ and v,
            and output the shortest $(V, E, \omega_t)$-additive path p from s to v
                if $\omega_{MA}(p) < d(v)$
                    $p(v) := p$
```

$$d(v) = \omega_{MA}(p).$$

5. Algorithm output: p(v) and d(v), $\forall v=1,....,N$.

The complexity of the algorithm is $O(|M| \cdot N^2)$.

5.6 Simulation results

In order to evaluate the proposed system's architecture and algorithm performance, an example is presented of a simulation performed with a program assembled in C++. A capture of the user interface of the program is shown in Figure 5-9, while the source code of the application software is attached in Appendix III.

```
==================
 M A I N   M E N U
==================
1.Enter Edges
2.Delete Edges
3.Show table
4.Find Shortest path with Dijkstra Algorithm
5.Exit
==================
```

Figure 5-9 *Program's user interface*

At first, seven nodes are considered in a fully connected mesh topology, as shown in Figure 5-10. "Node 1" is the corresponding MRBS, "Node 8" is the MS and the rest of the nodes are the MRSs. Through the user interface of the program, the values of the rates for each link are inserted and the inverse of the rates are calculated and registered as weights for the edges of our graphed network. The respective tables are shown in Figure 5-11, in which the values in row H and column H are zeros, denoting that none of the nodes is connected to "Node 8".

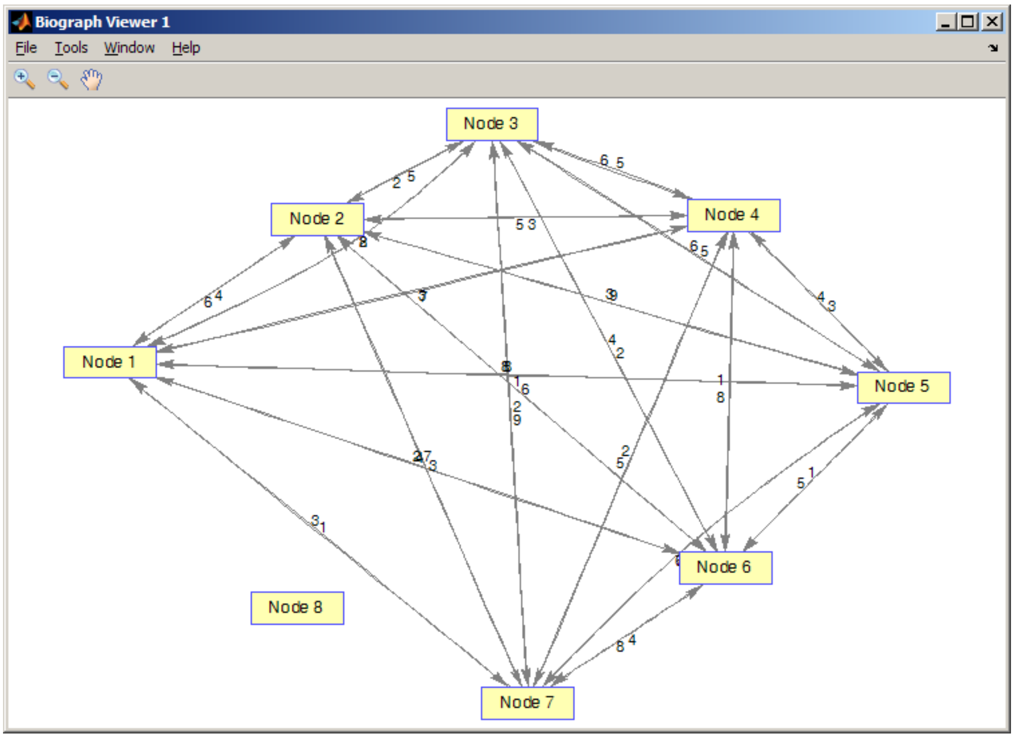
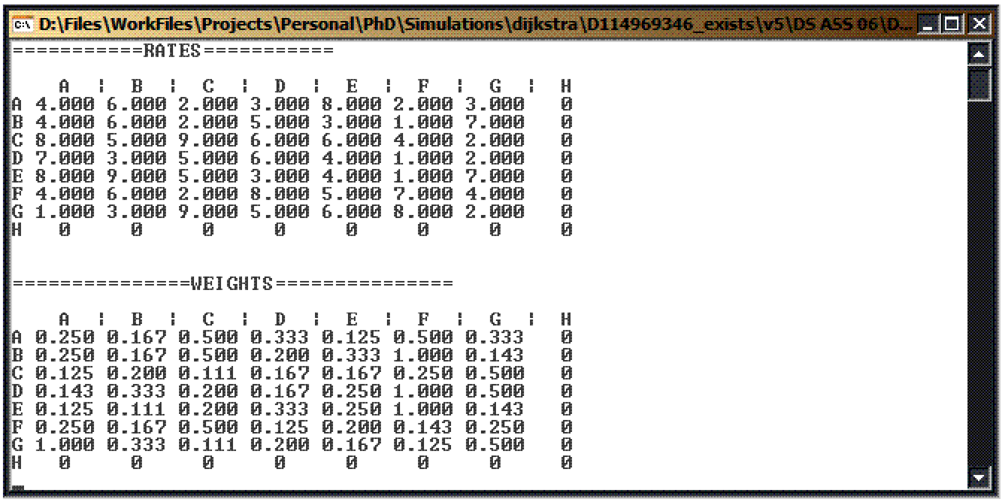


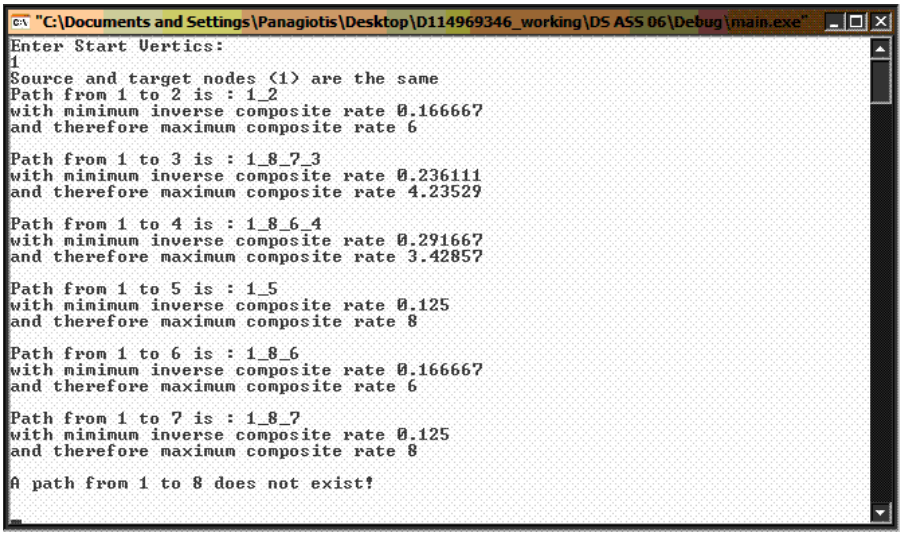
Figure 5-10 *Network topology*



```
===========RATES===========

    A   ¦   B   ¦   C   ¦   D   ¦   E   ¦   F   ¦   G   ¦   H
A 4.000 6.000 2.000 3.000 8.000 2.000 3.000    0
B 4.000 6.000 2.000 5.000 3.000 1.000 7.000    0
C 8.000 5.000 9.000 6.000 6.000 4.000 2.000    0
D 7.000 3.000 5.000 6.000 4.000 1.000 2.000    0
E 8.000 9.000 5.000 3.000 4.000 1.000 7.000    0
F 4.000 6.000 2.000 8.000 5.000 7.000 4.000    0
G 1.000 3.000 9.000 5.000 6.000 8.000 2.000    0
H   0     0     0     0     0     0     0      0


===============WEIGHTS===============

    A   ¦   B   ¦   C   ¦   D   ¦   E   ¦   F   ¦   G   ¦   H
A 0.250 0.167 0.500 0.333 0.125 0.500 0.333    0
B 0.250 0.167 0.500 0.200 0.333 1.000 0.143    0
C 0.125 0.200 0.111 0.167 0.167 0.250 0.500    0
D 0.143 0.333 0.200 0.167 0.250 1.000 0.500    0
E 0.125 0.111 0.200 0.333 0.250 1.000 0.143    0
F 0.250 0.167 0.500 0.125 0.200 0.143 0.250    0
G 1.000 0.333 0.111 0.200 0.167 0.125 0.500    0
H   0     0     0     0     0     0     0      0
```

Figure 5-11 *Rates and weights*

The end-user is “Node 8”, not yet connected to the network, while the program provides the ability to select any of the other nodes as a source during each run. For example, as shown in Figure 5-12, if “Node 1” is selected as the source of the transmission, the best paths from the source node to all other nodes are calculated, whereas a message appears and informs that “Node 8” is unreachable.



```
Enter Start Vertics:
1
Source and target nodes (1) are the same
Path from 1 to 2 is : 1_2
with mimimum inverse composite rate 0.166667
and therefore maximum composite rate 6

Path from 1 to 3 is : 1_8_7_3
with mimimum inverse composite rate 0.236111
and therefore maximum composite rate 4.23529

Path from 1 to 4 is : 1_8_6_4
with mimimum inverse composite rate 0.291667
and therefore maximum composite rate 3.42857

Path from 1 to 5 is : 1_5
with mimimum inverse composite rate 0.125
and therefore maximum composite rate 8

Path from 1 to 6 is : 1_8_6
with mimimum inverse composite rate 0.166667
and therefore maximum composite rate 6

Path from 1 to 7 is : 1_8_7
with mimimum inverse composite rate 0.125
and therefore maximum composite rate 8

A path from 1 to 8 does not exist!
```

Figure 5-12 *Path evaluation*

The next step is to insert the values of the link rates from all nodes to “Node 8” and vice versa. In this case, the topology of our network is the one presented in Figure 5-13 and the corresponding tables of rates and weights of the edges are shown in Figure 5-14, in which row H and column H are filled with non-zero numbers, denoting that everybody can now access the end-user.

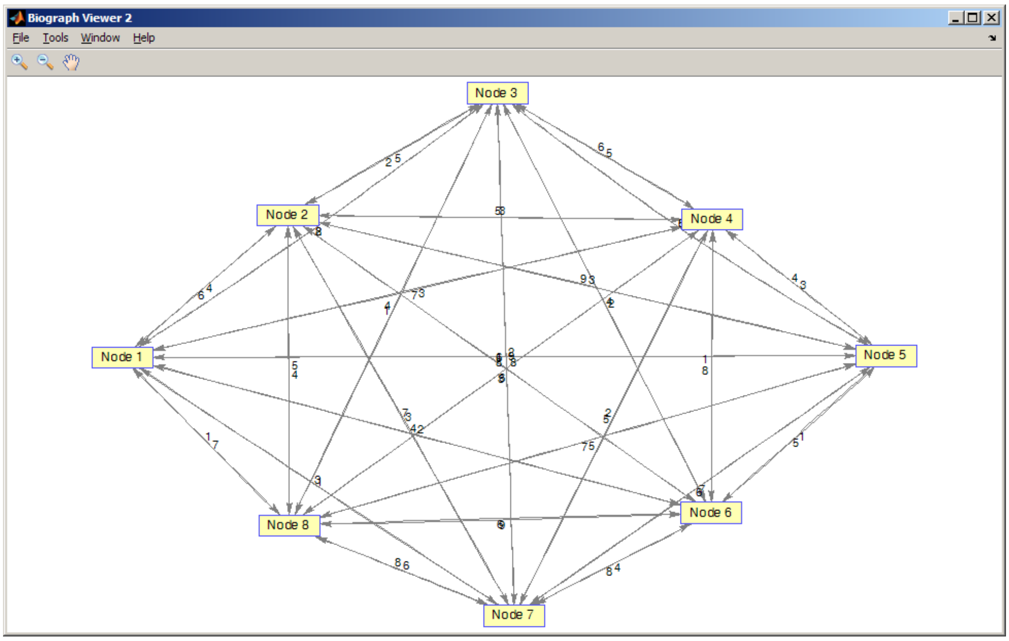
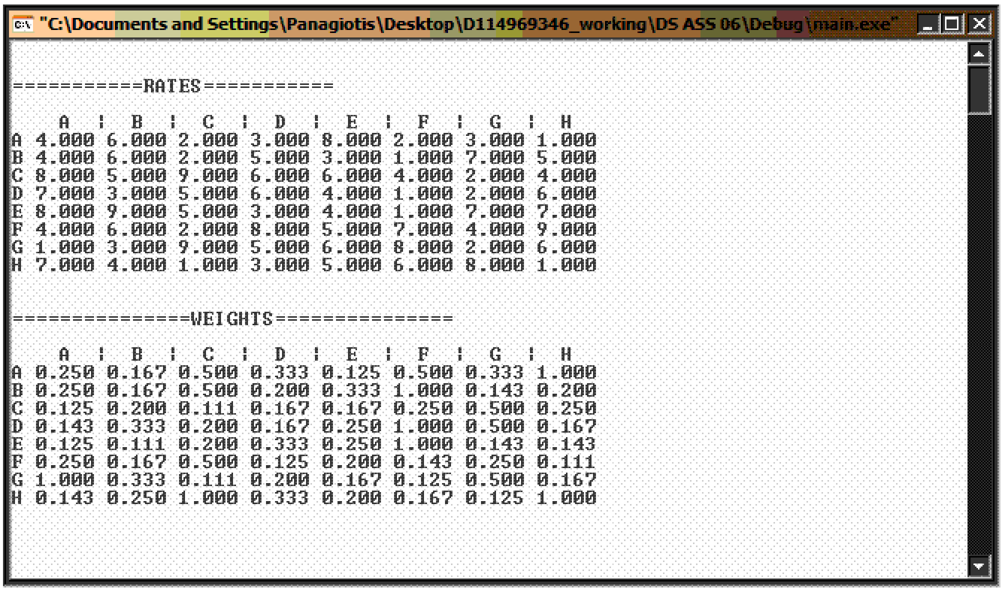


Figure 5-13 *Network topology with “Node 8” connected*

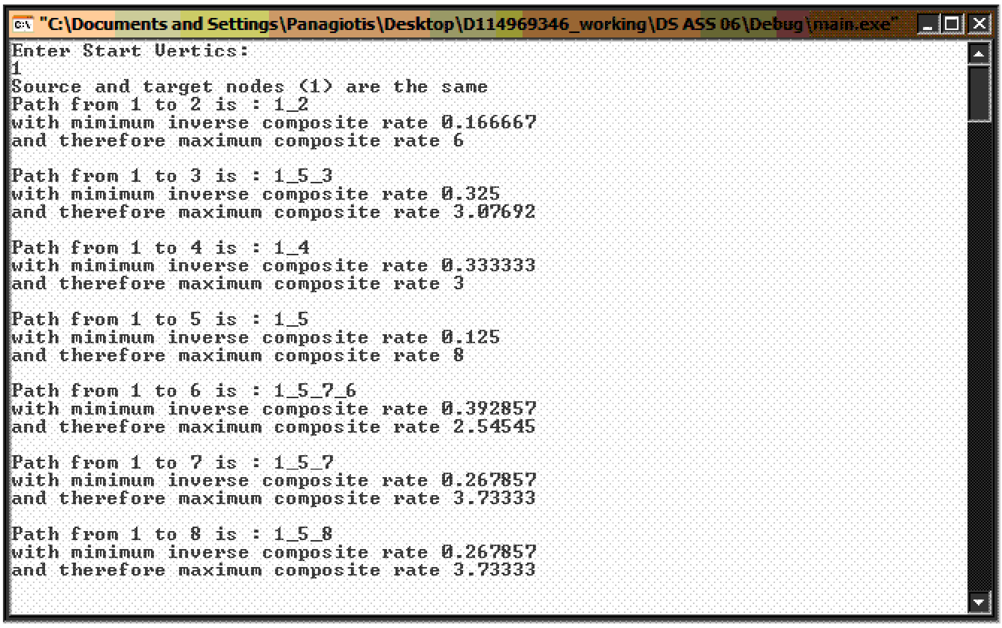


```
===========RATES===========

    A  ¦  B  ¦  C  ¦  D  ¦  E  ¦  F  ¦  G  ¦  H
A 4.000 6.000 2.000 3.000 8.000 2.000 3.000 1.000
B 4.000 6.000 2.000 5.000 3.000 1.000 7.000 5.000
C 8.000 5.000 9.000 6.000 6.000 4.000 2.000 4.000
D 7.000 3.000 5.000 6.000 4.000 1.000 2.000 6.000
E 8.000 9.000 5.000 3.000 4.000 1.000 7.000 7.000
F 4.000 6.000 2.000 8.000 5.000 7.000 4.000 9.000
G 1.000 3.000 9.000 5.000 6.000 8.000 2.000 6.000
H 7.000 4.000 1.000 3.000 5.000 6.000 8.000 1.000


===============WEIGHTS===============

    A  ¦  B  ¦  C  ¦  D  ¦  E  ¦  F  ¦  G  ¦  H
A 0.250 0.167 0.500 0.333 0.125 0.500 0.333 1.000
B 0.250 0.167 0.500 0.200 0.333 1.000 0.143 0.200
C 0.125 0.200 0.111 0.167 0.167 0.250 0.500 0.250
D 0.143 0.333 0.200 0.167 0.250 1.000 0.500 0.167
E 0.125 0.111 0.200 0.333 0.250 1.000 0.143 0.143
F 0.250 0.167 0.500 0.125 0.200 0.143 0.250 0.111
G 1.000 0.333 0.111 0.200 0.167 0.125 0.500 0.167
H 0.143 0.250 1.000 0.333 0.200 0.167 0.125 1.000
```

Figure 5-14 *Rates and weights*

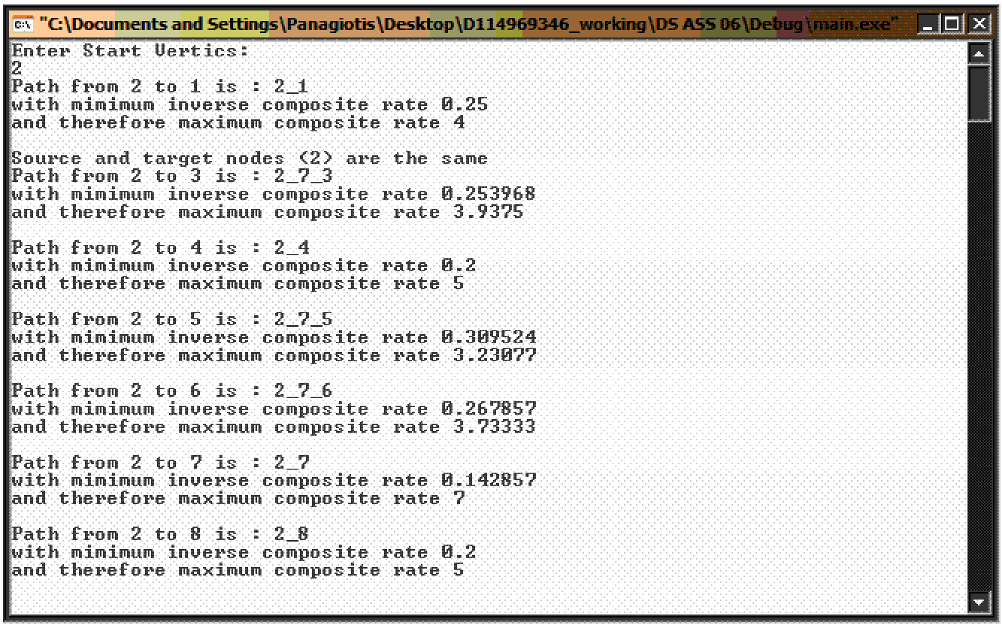
The estimation of paths in this case can be implemented in two ways. The first one is to store the values of rates and weights already calculated in the previous step and then perform a new calculation of only the difference that the new node will induce (delta end-user optimization). The second way is to recalculate all values. This program implements the second approach since it is a small network and the calculations can be performed very quickly.

Finally, the source node has to be selected and the best paths for all nodes are estimated. In the case where “Node 1” is chosen as the source node, the best path to reach “Node 8” is through “Node 5” (Figure 5-15). On the other hand, if “Node 2” is selected as the source of the transmission, the best path to reach “Node 8” is the direct link (Figure 5-16). It has to be mentioned that the paths in this implementation don’t have the limit of 2 hops referred in section 4.4.



```
Enter Start Vertics:
1
Source and target nodes (1) are the same
Path from 1 to 2 is : 1_2
with mimimum inverse composite rate 0.166667
and therefore maximum composite rate 6

Path from 1 to 3 is : 1_5_3
with mimimum inverse composite rate 0.325
and therefore maximum composite rate 3.07692

Path from 1 to 4 is : 1_4
with mimimum inverse composite rate 0.333333
and therefore maximum composite rate 3

Path from 1 to 5 is : 1_5
with mimimum inverse composite rate 0.125
and therefore maximum composite rate 8

Path from 1 to 6 is : 1_5_7_6
with mimimum inverse composite rate 0.392857
and therefore maximum composite rate 2.54545

Path from 1 to 7 is : 1_5_7
with mimimum inverse composite rate 0.267857
and therefore maximum composite rate 3.73333

Path from 1 to 8 is : 1_5_8
with mimimum inverse composite rate 0.267857
and therefore maximum composite rate 3.73333
```

Figure 5-15 *Path calculation with "Node 1" set as source*



```
Enter Start Vertics:
2
Path from 2 to 1 is : 2_1
with mimimum inverse composite rate 0.25
and therefore maximum composite rate 4

Source and target nodes (2) are the same
Path from 2 to 3 is : 2_7_3
with mimimum inverse composite rate 0.253968
and therefore maximum composite rate 3.9375

Path from 2 to 4 is : 2_4
with mimimum inverse composite rate 0.2
and therefore maximum composite rate 5

Path from 2 to 5 is : 2_7_5
with mimimum inverse composite rate 0.309524
and therefore maximum composite rate 3.23077

Path from 2 to 6 is : 2_7_6
with mimimum inverse composite rate 0.267857
and therefore maximum composite rate 3.73333

Path from 2 to 7 is : 2_7
with mimimum inverse composite rate 0.142857
and therefore maximum composite rate 7

Path from 2 to 8 is : 2_8
with mimimum inverse composite rate 0.2
and therefore maximum composite rate 5
```

Figure 5-16 *Path calculation with "Node 2" set as source*

Figures 5-15 and 5-16 present the results of this simulation, where it can be observed that the path from the source node to the destination one doesn't necessarily have to be through another node (MRS), but it can also be the direct link from the MRBS to the end-user, as shown in Figure 5-16. If the numerical results are further analyzed, it can be noticed that, by selecting the inverse of rates as weights in edges, the throughput is maximized. This is better illustrated in Table 5-1, where it can be noted that the best route to reach the end-user from the source node is the one that maximizes the composite rate. Another conclusion derived is that, in some occasions, a path with many "hops" can have increased throughput compared to that of another path with less "hops", as it is clearly shown in the last three rows of Table 5-1.

Table 5-1 *Sorting of paths from source "Node 1" (MRBS) to target "Node 8" (MS)*

Path from "Node 1" to "Node 8"	***Mimimum inverse composite rate***	***Maximum composite rate***
1_5_8	0.267857	3.73333
1_2_8	0.366667	2.72727
1_5_7_8	0.434524	2.30137
1_4_8	0.5	2
1_5_7_6_8	0.503968	1.98425
1_5_3_8	0.575	1.73913
1_8	1	1

Nevertheless, it has to be stated that adding relays in a network, it also adds time delay because of the management messages that have to be exchanged etc. Unfortunately, due to the lack of support from the current frame structure of the 802.16j standard, this cannot be studied with the simulation provided. In the future, when the frame structure of the MRSs is completed and ready to support more than 2 "hops", the use of MRSs will be easier to be studied in terms of delay and scalability of the system.

5.7 Summary

In this chapter two cases of transmission have been studied. Both of them consider unicast transmission in different settings regarding the intra-cell interference level. Additionally, a MRBS is the source of the transmission with several MRSs distributed randomly within its range, while backhaul links are established among the MRSs and the MRBS. For the access level, the assumption that the MRBS routes data to the end users through the MRSs has been made. In both cases, the schemes introduced are based on the standard Dijkstra algorithm.

In the first case, no mutual intra-cell interference (NICI) is allowed, while only one transceiver (base, relay or mobile) can transmit at a given time and frequency. When a user enters the network, for each neighbouring candidate sector a relay route that maximizes the overall relay-based rate is found, considering that all candidate sectors, if admitting this additional user, will not disconnect service for any of the existing users. Among them the one that entails the greatest rate to the considered user is selected. This process is performed independently for each user, meaning that when a new user is considered for a sector, only this user is involved in the optimization process.

The second case is the one of a unicast transmission in a LICI network. In such a setting, each user is served at an exclusive time and frequency; however the network transceivers can partly work simultaneously. The network transceivers are divided into several tiers, where within each tier the interference is low. Yet, two transceivers of the same tier cannot transmit at the same time and frequency. They can operate though at the same time but on different frequencies with sufficient spacing among them to prevent interference. During the initiation of the network, an offline-algorithm optimizes the entire network’s backhaul and updates occur only when backhaul links change. Once backhaul optimization is established offline, dynamic frequent updates take place with respect to the end users. These updates amount to very light-weight executions of the Dijkstra algorithm in which an equivalent of a 2-hop optimization problem is solved. This is called the delta end-user optimization.

Finally, an attempt to provide simulation results for the scheme proposed has been made, which is subject to restrictions imposed by the frame structure of the 802.16j standard that doesn’t support more than one MRS in a path. Preliminary results demonstrate that throughput can be maximized with the algorithm introduced, depending on the rates of the links in the network.

6 New routing algorithm introduced based on maximum graph-flow algorithms

6.1 Introduction

For the algorithm proposed [Tsiakas *et al.*, 2009], multicast transmission with Limited Intra-Cell Interference (LICI) is allowed. In particular, several end-users can be served simultaneously while relays of the same tier can also transmit at the same time. This setting is suitable for urban environments like the one in Figure 6-1. Usually, such networks have much higher capacity than NICI networks due to the increased number of end-users who they have to serve, and the high demand of bandwidth which they have to provide. This can be strengthened when supported by the MIMO-SDMA technology and various optimization schemes.

Figure 6-1 *An example of a LICI environment - Urban area*

Routing decisions regarding which MRBS will serve each mobile user are more complex in the LICI setting integrating MRss, where additional degrees of freedom may be applicable, since dense deployment is typically assumed and users are able to interact with a wider range of serving nodes. A usual case is that a mobile user can be served by more than one transmitter, either a base station or a relay. An example is depicted in Figure 6-2, where every end-user can be served by at least two transmitters, including the MRBS. The dynamic decision about the desired link is based on updated channel knowledge, taking under consideration both the needs of the given subscriber and the service loads experienced by each available transmitter.

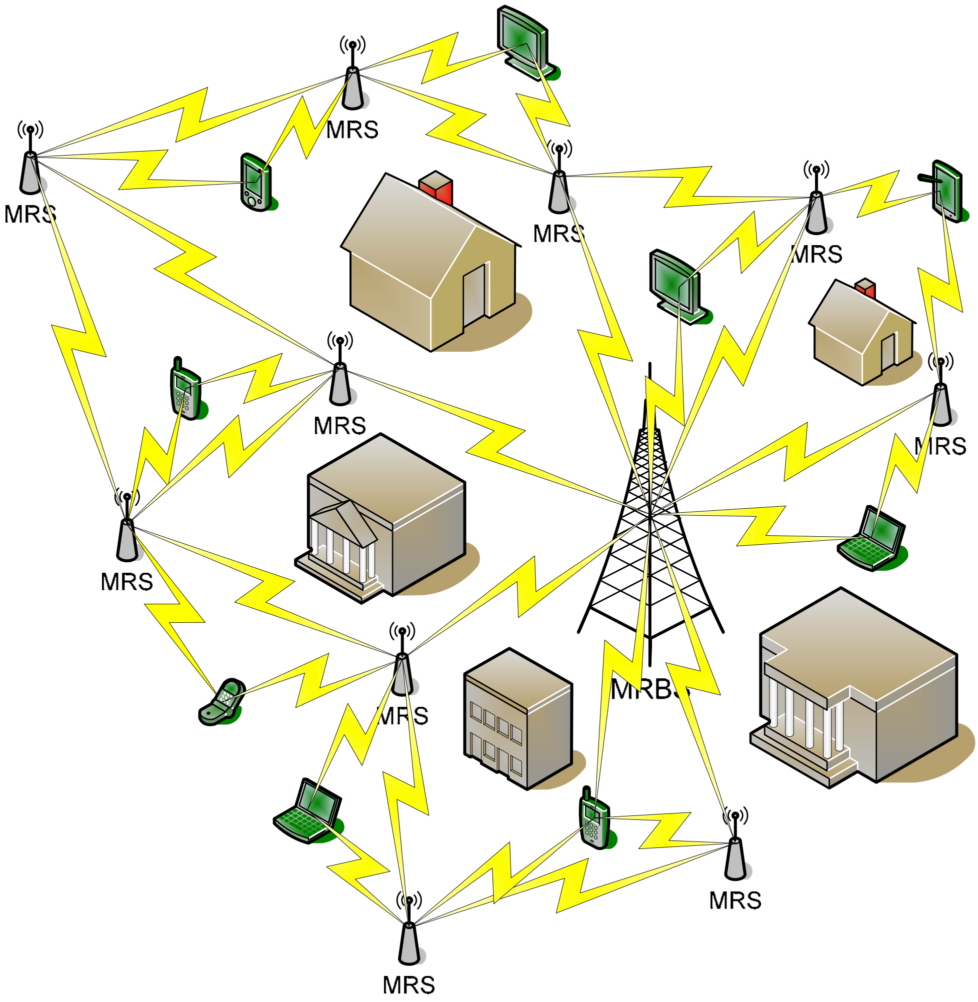


Figure 6-2 *Users able to be served by more than one transmitter*

The core element of the algorithm introduced is the maximization of the throughput of each candidate sector - including the new user. However, unlike the NICI case, mutual interference within each tier exists and plays an important role. The proposed algorithm acknowledges this interference and offers the means to deal with it. Another advantage is that it offers an optional scheme where more demanding and costly applications optimize the backhaul network in an offline mode without considering the end-users, in which each base/relay is assumed to provide a maximal data rate at the access level.

In the following section, the network topology and transmission method under investigation are described. Section 6.3 introduces the classical notion of flow networks and its relation to the selected settings. Section 6.4 describes the functionality of the standard Ford-Fulkerson algorithm on which the proposed one is based. Section 6.5 presents the details of the new algorithm designed. At first, a scheme of wireless network data flow, which directly adapts the maximum flow method of Ford-Fulkerson (FF), is described. The maximum flow concept refers to finding the most suitable and feasible path through a number of nodes in order to achieve transmission from a source to a sink node. For the maximum flow to be found, all the available routes between the source and the destination of a transmission have to be evaluated. Finally, the FF procedure is utilized as a sub-routine in an algorithm which is more tuned towards SDMA beam-forming-optimized wireless communication.

6.2 Conditions and assumptions

The WRMN topology used for studying the algorithm designed is depicted in Figure 6-3. In particular, a MRBS is the source of the transmission with several MRSs distributed randomly within its range, while backhaul links are established among the MRSs and the MRBS.

For the access level, there is the assumption that the MRBS routes data to the end users through the MRSs.

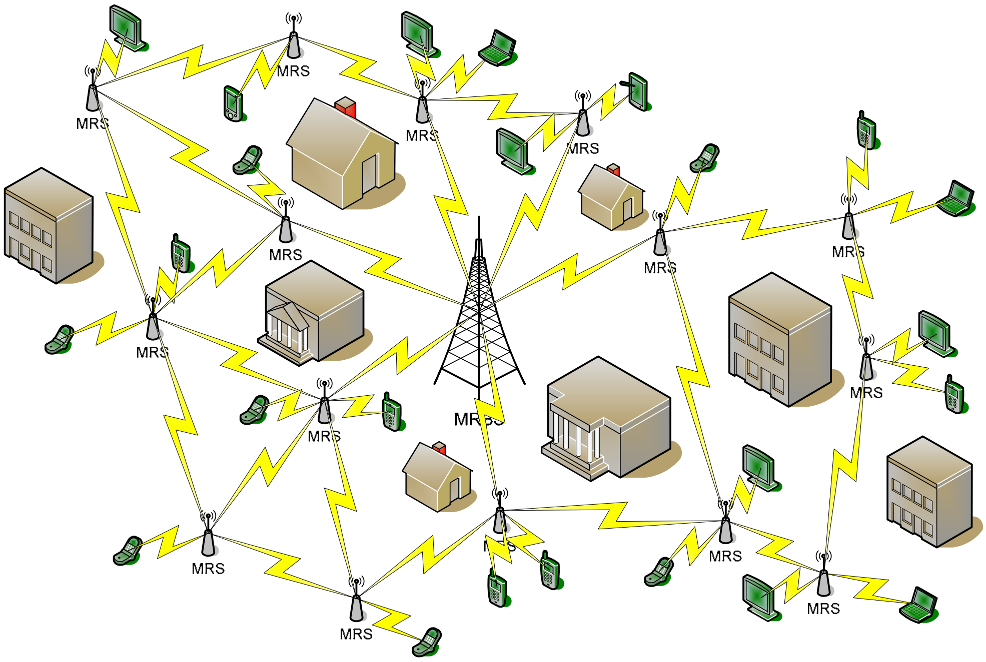


Figure 6-3 *WRMN Network topology*

The routing scheme selected for the implementation of the algorithm is multicast, as shown in Figure 6-4. This means that more than one end user can be accommodated at the same time.

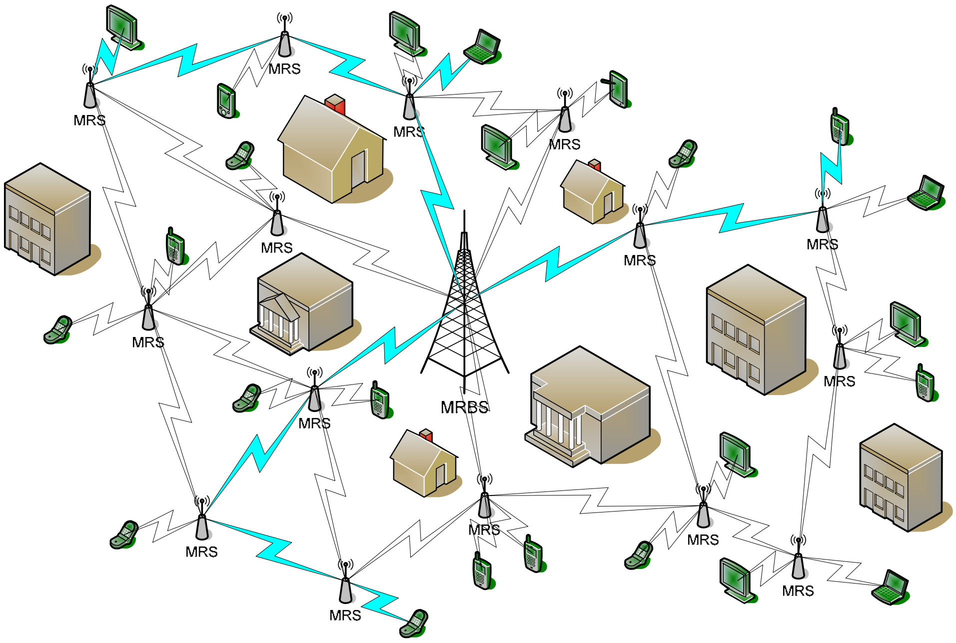


Figure 6-4 *Multicast transmission*

6.3 Introduction to the classic notion of flow networks, and its relation to multicast LICI transmission

6.3.1 Definition of a flow network

LICI’s core optimization algorithm is based on a generalization of the notion of the flow network, which is a directed graph G = (V,E) with a capacity function that is a real function:

$$c: V\times V\rightarrow R^{+}\ (R^{+}\equiv\{x\in R: x\geq 0\}).$$

For every edge (u,v) $\in$ E it has a capacity with a positive value:

$$c(u,v) \text{ and for } (u,v)\notin E,\ c(u,v)=0.$$

Two vertices can be singled out in a flow network: the source (s) and the sink (t). The graph G is connected when every vertex exists in a random route connecting the source with the sink,. In this context, the capacity function has the role of the upper bound of the data flow in the link from u to v.

6.3.2 Definition of a flow f in the network G, and its value

A flow in G is a real function $f:V\times V\rightarrow R$ with the following three properties for all the nodes u and v:

- *(i)* *Capacity constraints:* $f(u,v)\leq c(u,v)$. The flow along an edge cannot exceed its capacity.
- *(ii)* *Skew symmetry:* $f(u,v) = -f(v,u)$. The net flow from u to v must be the negative of the net flow from v to u.
- *(iii)* *Flow conservation:* $\Sigma_{w\in V} f(u,w) = 0$, for all $u\in V\backslash\{s,t\}$. The net flow of a node is zero, except for the source which "produces" flow, and the sink which "consumes" flow.

The value of a flow f is:

$$|f| = \Sigma_{v \in V} f(s,v).$$

This value signifies the amount of flow in the network and the main goal in this chapter is its maximization. This is termed as the maximal flow problem.

6.3.3 Other essential definitions

Some basic definitions for the terms “residual capacity”, “residual network” and “augmenting path” have to be provided for the better understanding of the following sections.

Residual capacity of an edge in a network G = (V,E) with a capacity function c and a flow f is defined as follows:

$$c_f(u,v) \equiv c(u,v) - f(u,v), \text{ for all } u,v \in V.$$

Set

$$E_f = \{(u,v) \in V \times V\text{: } c_f(u,v) > 0\}.$$

This defines a directed graph G_f = (V, E), called a *residual network,* with capacity function c_f. It provides the amount of available capacity for the given capacity c and existing flow f.

An *augmenting path* is a path:

$$p = (v_1, \ldots., v_k),\ v_i \in V,\ (i = 1, \ldots, k)$$

in the residual network, where:

$$v_1 = s,\ v_k = t \text{ and } c_f(v_i,v_{i+1}) > 0,\ i = 1, \ldots, k.$$

A network is at maximum flow if and only if there is no augmenting path in the residual network. The residual capacity of the path is defined by:

$$c_f(p) = \min\{ c_f(v_i,v_{i+1}): i=1,...,k\}.$$

The iterations of the ensuing maximal flow algorithm are justified by the following notion of flow network “cut”, and its related propositions. Let us consider the flow networks G = (V, E) with capacity c and flow f.

(i) If $X, Y \subseteq V$ then

$$f(X, Y)= \Sigma_{x\in X,y\in Y}\, f(x, y)$$

$$c(X, Y)= \Sigma_{x\in X,y\in Y}\, c(x, y)$$

(ii) A *cut* in G is a split of the nodes set V into two disjoint sets S and T whose union is V, such that s is in S and t is in T. Hence, there are $2^{|V|-2}$ different possible cuts in a graph.

(iii) The capacity of a cut (S, T) is c(S, T), that is the sum of the capacity of all the edges crossing the cut, from the region *S* to the region T.

6.3.4 The underlining intuition and some examples

Example 1: Intuitively each vertex can be resembled to a pipe of a defined diameter, so as to supply a certain volume of water flow, while the edges can be viewed as pipes junctions [Cover & Thomas, 2006]. In each junction, the total amount of water going in and out of it must be of the same quantity. Additionally, there is a water inlet, which is the source, and an outlet, which is the sink, in the system. A flow is a possible route for the water to get from the source to the sink so that the quantity of water going into the inlet and coming out of the outlet is the same. Following this illustration, it can be evident that the total flow of a network is the rate at which water comes out of the source, which is equal to the rate it comes into the outlet. The flow networks of this research model information flow in a wireless relay mesh network.

In Figure 6-5, a flow network is depicted, containing a source (s), a sink (t), and four additional nodes (a, b, c and d). The flow and capacity of the network are denoted by f and c respectively. It can be observed that the network upholds the three properties defined earlier. It can also be seen that the total outgoing flow from the source is the same as the total incoming flow to the sink, which is 5. Thus, no flow is generated or consumed in any of the other nodes.

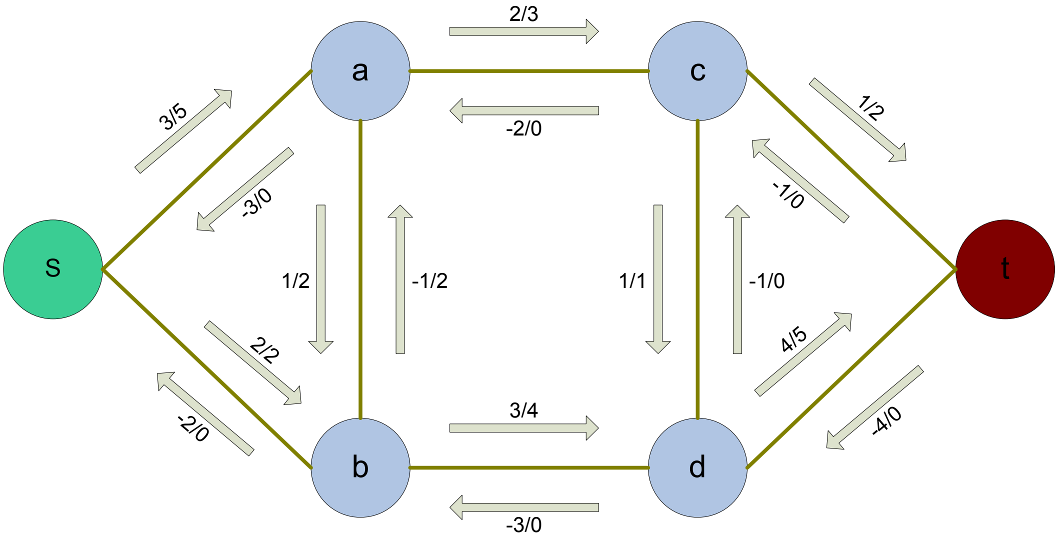


Figure 6-5 *An example of a flow network*

The residual network of the flow shown in Figure 6-5 is analyzed below. Since there is a positive residual capacity on some edges, the current flow is not the maximum. Therefore, there is an available capacity along other routes: (s, a, c, t) and (s, a, b, d, t), which are the augmenting paths. The residual capacity of the initial path is:

$$\min[(c(s,a) - f(s,a)), (c(a,c) - f(a,c)), (c(c,t) - f(c,t))] =$$

$$\min[(5 - 3), (3 - 2), (2 - 1)] =$$

$$\min(2, 1, 1) = 1 .$$

Example 2: Another example of a flow network can be provided by a railway network where people start from one station (source) and go to another (sink). In this case, each vertex can resemble to a coach of a defined capacity, so as to transport a certain number of people. The edges can be viewed as terminals where the entrances and exits are sealed so as no-one can enter or exit before they arrive at the terminal station, but they can change coaches and, therefore, routes. In each terminal, the total number of people going in and out must be the same. A flow is a possible route for people to get from the source to the sink so that this number of people going into the departure station and coming out of the arrival station is the same. Following this illustration, it is evident that the total flow of the network is the rate at which people come out of the source, which is equal to the rate they come into the outlet.

In Figure 6-6, this flow network is depicted. One hundred fifty passengers start from “Piraeus Port” (source) to reach the “Airport” station (sink). It can be seen that the total outgoing flow from the source is the same as the total incoming flow to the sink, which is one hundred fifty. Thus, no flow is generated or consumed in any of the other stations, regardless the route each passenger has followed to reach his/her destination. For example, thirty of them left from “Monastiraki” and went through the blue route to “Syntagma” station where ten of them took the red route and the rest remained in the blue one, while one hundred twenty remained in the initial green route, from which ninety of them split in “Omonia” station and joined the other ten in the red line. Finally, people from both the red and the green routes merged in the green one after “Attiki” station, changed at “Neratziotissa” and met the rest of the passengers (twenty) of the blue line in “Doukissis Plakentias” station to follow the blue route to the “Airport” station.

Figure 6-6 *Metro lines as a flow network*

The residual network of the flow shown in Figure 6-6 is analyzed below. Since there is a positive residual capacity on some edges, the current flow is not the maximum. Therefore, there is an available capacity along other routes: (“Piraeus Port”, “Monastiraki”, “Syntagma”, “Doukissis Plakentias”, “Airport”), (“Piraeus Port”, “Monastiraki”, “Syntagma”, “Omonia”, “Attiki”, “Neratziotissa”, “Doukissis Plakentias”, “Airport”), (“Piraeus Port”, “Monastiraki”, “Omonia”, “Attiki”, “Neratziotissa”, “Doukissis Plakentias”, “Airport”) etc, which are the augmenting paths. The residual capacity of the initial path is:

min[(c(“Piraeus Port”, “Monastiraki”) − f(“Piraeus Port”, “Monastiraki”)), (c(“Monastiraki”, “Syntagma”) − f(“Monastiraki”, “Syntagma”)), (c(“Syntagma”, “Doukissis Plakentias”) − f(“Syntagma”, “Doukissis Plakentias”)), (c(“Doukissis Plakentias”, “Airport”) − f(“Doukissis Plakentias”, “Airport”))] =

min[(200 − 150), (50 − 30), (50 − 20), (300 − 150)] =

min(50, 20, 30, 150) = 20.

6.3.5 Representation of a WRMN as a flow network

Following the previous examples, a WRMN can also be drafted as a flow network. Each channel can be modeled to a vertex, so as to supply a certain data rate, while the edges can be viewed as MRSs. In each junction, the data going in and out must be the same. Additionally, there is a source for the transmission, which is the MRBS, and a target, which is the MS. Then, a flow would be a possible route for data to get from the MRBS to the MS. This flow network is depicted in Figure 6-7.

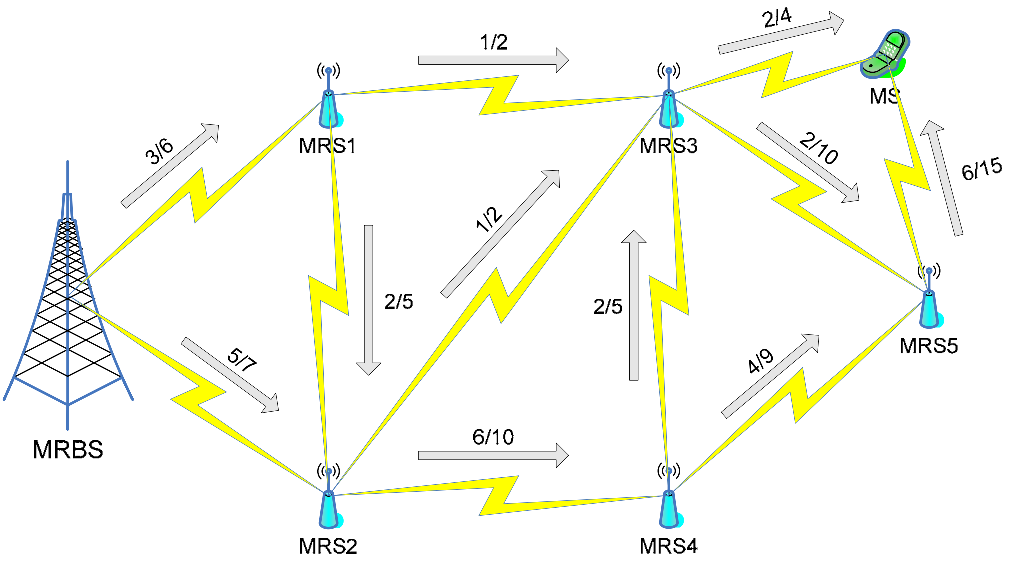


Figure 6-7 *A WRMN as a flow network*

The residual network of the flow shown in Figure 6-7 is analyzed below. Since there is a positive residual capacity on some edges, the current flow is not the maximum. Therefore, there is an available capacity along other routes: (MRBS, MRS1, MRS3, MS), (MRBS, MRS2, MRS4, MRS5, MS), (MRBS, MRS1, MRS3, MRS5, MS), (MRBS, MRS1, MRS2, MRS3, MS) etc, which are the augmenting paths. The residual capacity of the initial path is:

$$\min[(c(\mathrm{MRBS}, \mathrm{MRS1}) - f(\mathrm{MRBS}, \mathrm{MRS1})), (c(\mathrm{MRS1}, \mathrm{MRS3}) - f(\mathrm{MRS1}, \mathrm{MRS3})), (c(\mathrm{MRS3}, \mathrm{MS}) - f(\mathrm{MRS3}, \mathrm{MS}))] =$$

$$\min[(6 - 3), (2 - 1), (4 - 2)] =$$

$$\min(3, 1, 2) = 1.$$

6.4 Description of the Ford-Fulkerson (FF) algorithm

The basic idea behind the FF algorithm [Ford & Fulkerson, 1956, Cormen *et al.*, 2001] is to extract a network graph with a number of network nodes and links from each node, showing how much capacity can flow down that link. Then, a way to get the maximum flow from a source to a destination must be found. This is done by creating paths that contain links with the highest available capacities.

The FF algorithm goes through these steps shown in Figure 6-8 as a flowchart:

1. The MRBS initializes the value of the flow rate of each edge to zero.
2. Then it looks for a path heading from itself to a target user for which the value of an edge can be increased.
3. If one exists

   a. It compares the residual capacity of all the edges in the path.

   b. It adds the smallest residual capacity to all the edges in the path.

4. If there are no more such paths found, the algorithm terminates and the network has been optimized. If there are more paths, the algorithm returns to the 2nd step.

5. The MRBS repeats the steps 2-4 until no more such paths are found and the network has been optimized.

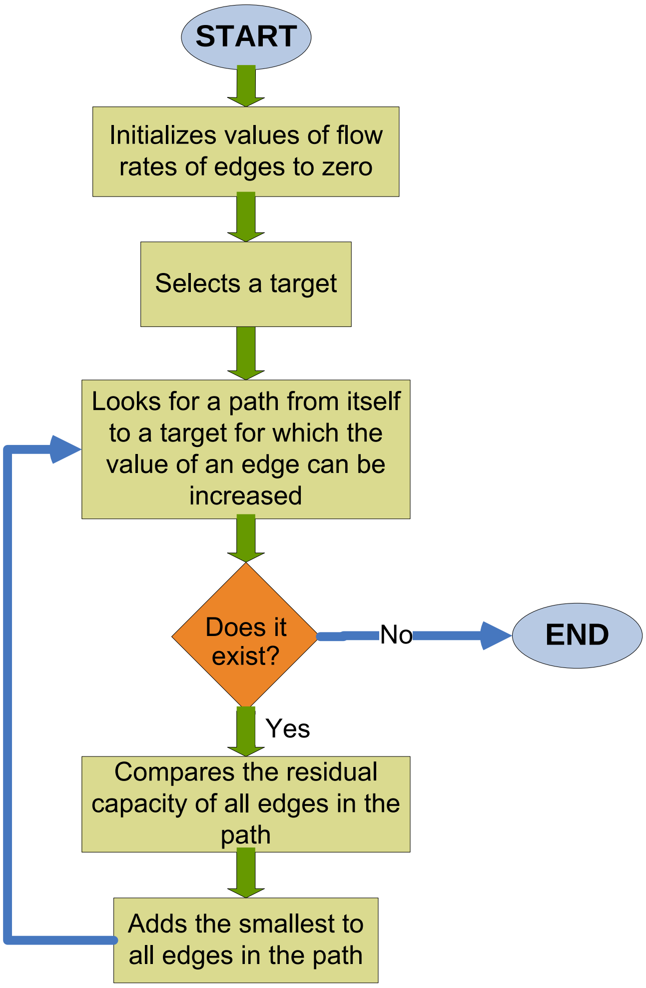


Figure 6-8 *Ford-Fulkerson's algorithm flowchart*

6.5 Proposed algorithm: The maximum flow minimum cut theorem

The maximum flow concept refers to finding the most suitable and feasible path through a number of nodes in order to achieve transmission from a source node s to a sink node s. It can also be seen as the maximum amount of flow that can be transferred from s to t. This view is of paramount importance in data networks, where a maximum throughput and a minimum delay are requested accordingly.

In order to find the maximum flow, one must look through all the available routes between the source and the sink (destination) of a transmission. Links among nodes are represented by pipes with limited capacities and the maximum flow can be found by assigning flow to each of the pipes, so that the total flow from the source to the destination has a maximum value.

A cut is any set of directed links, containing at least one link in every path from the source to the destination. The cut value is the sum of the capacities of all the links of the paths in the direction from the source to the sink. The minimum cut problem is to find the cut across the network that has the minimum value over all possible cuts.

The maximum value of a flow in a network from a source to a sink is equal to the minimum value of a flow from a source to a sink cut in the network, which is the maximum flow minimum cut theorem.

The aim of the introduced algorithm is to implement an efficient multi-hop routing scheme that is interference aware, and hence maximizes parallel transmission, providing at the same time high throughput and scalability.

Let G = (V, E) be a flow network with capacity c and flow f. The FF maximum flow algorithm is based on the following, fundamental theorem:

Max-flow min-cut theorem.

1. f is a maximum flow in G
2. The residual network G_f contains no augmenting paths.
3. | f | = c(S, T) for some cut (S, T).

The proof of this theorem requires the following more elementary lemma.

Lemma:

Let $X,Y,Z \subseteq V$ then:

(1) $f(X,Y) = -f(Y,X)$
(2) $f(X,X) = 0$
(3) $f(X \cup Y,Z) = f(X,Z) + f(Y,Z)$ when $X \cap Y = \varnothing$
(4) $f(V,t) = |f|$
(5) if (S,T) a cut in G then $f(S,T) = |f| \leq c(S,T)$
(6) If f' is a flow in G_f then f + f' is a flow in G and $|f + f'| = |f| + |f'|$.

Proof of the max-flow min-cut FF theorem (based on the lemma).

(1) $\Rightarrow$ (2): If f is a maximum flow in G, then the residual network G_f contains no augmenting paths.

In order to derive a contradiction, let us suppose that f is a maximum flow in G and the residual network G_f contains augmenting paths. An augmenting path f^* can be chosen, and a new flow $f + f^*$ in G can be obtained. Now the fact that:

$$| f + f^* | = | f | + | f^* | > | f |,$$

contradicts to the initial statement that f is a maximum flow.

(2) $\Rightarrow$ (3): If the residual network G_f contains no augmenting paths, then $|f| = c(S,T)$ for some cut (S,T) of G.

Let the set S contain all the vertices v that have a path connected from s on G_f. Since there is no augmenting path in G_f, t is in T=V\S. Thus, (S,T) is a cut in G. It follows for any $u \in S$, $v \in T$ that $f(u,v) = c(u,v)$, because otherwise $(u,v) \in E_f$ and v is also in S, which is a contradiction. Thus, by the above lemma $|f| = f(S,T) = c(S,T)$.

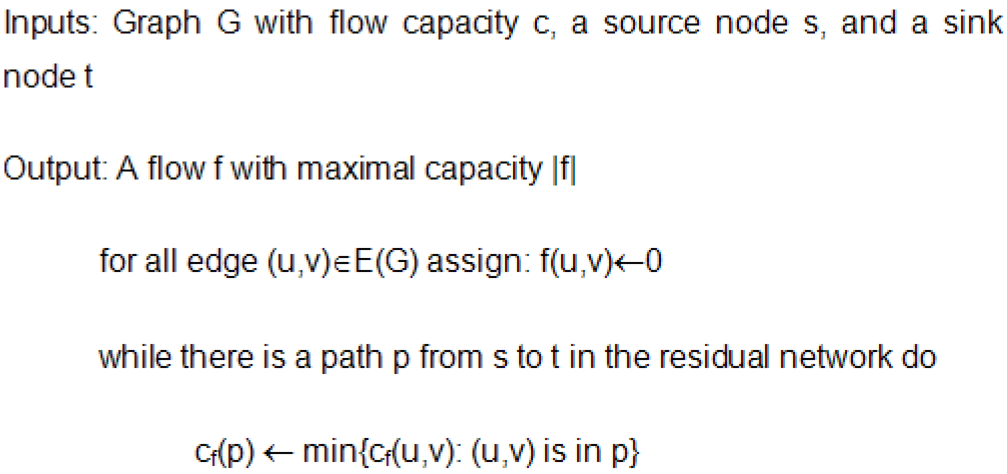
(3)$\Rightarrow$(1): If $| f | = c(S,T)$ for some cut (S,T) of G, then f is a maximum flow of G.

By the above lemma, $| f | \leq c(S,T)$ for cuts (S,T). Thus the condition $| f | = c(S,T)$ for some cut (S,T) of G implies that f is a maximal flow.

6.6 Implementation issues of the algorithm

The idea behind the algorithm is that added flow can be sent along such a path, that it has, from the source s to the sink t, positive residual capacity on all edges.

Algorithm Ford-Fulkerson



Inputs: Graph G with flow capacity c, a source node s, and a sink node t

Output: A flow f with maximal capacity |f|

```
for all edge $(u,v) \in E(G)$ assign: $f(u,v) \leftarrow 0$

while there is a path p from s to t in the residual network do

    $c_f(p) \leftarrow \min\{c_f(u,v): (u,v) \text{ is in } p\}$
```

```
for each edge $(u,v)\in p$
    $f(u,v) \leftarrow f(u,v) + c_f(p)$
    $f(v,u) \leftarrow -f(u,v)$ ♦
```

The path in the “while” step can be found with a breadth-first search in $G_f(V,E_f)$. The combined algorithm is called Edmonds-Karp in which the augmenting path selected is always the shortest one. According to the max-flow min-cut theorem, when no more paths in step 2 can be found, the flow of f is maximal.

There is no guarantee that this algorithm will ever reach the maximal flow. Thus, the algorithm is correct only when it does terminate. To avoid the possibility of non-termination and bound the complexity, capacities (flow functions and capacity function) can be integer products of some small atomic unit. Thereby, the runtime of FF is bounded by $O(E \cdot f^*)$, where E is the number of edges in the graph and f^* is the maximum integer flow in the graph. This occurs because each (e.g. breadth first) augmenting path can be found in O(E) time and increases the flow by an integer amount that is at least 1.

Suboptimal online stage. In a given time interval in which users’ configuration alters too fast for the full FF algorithm to take place, a simple optimization per user can be done with the following procedure. By selecting an end-user and considering only the FF links that exist with its surrounding network transceivers, the maximal transmission to that user alone (i.e. as if he had been the only user served) can be found. This is done for several users at a time, according to the scheduler. Finally, a convex combination of the flow function for each of these users is taken, where this combination is being based on scheduling consideration.

An illustration of the steps of an algorithm execution would be the one depicted in Figure 6-9. In the beginning (Figure 6-9-1) all flows are set to zero and at the end of the algorithm (Figure 6-9-6) there is the maximum exploitation of the network's capacity. There are four completed iterations of the "while" step, while in the fifth one no other path is found, so the algorithm terminates.

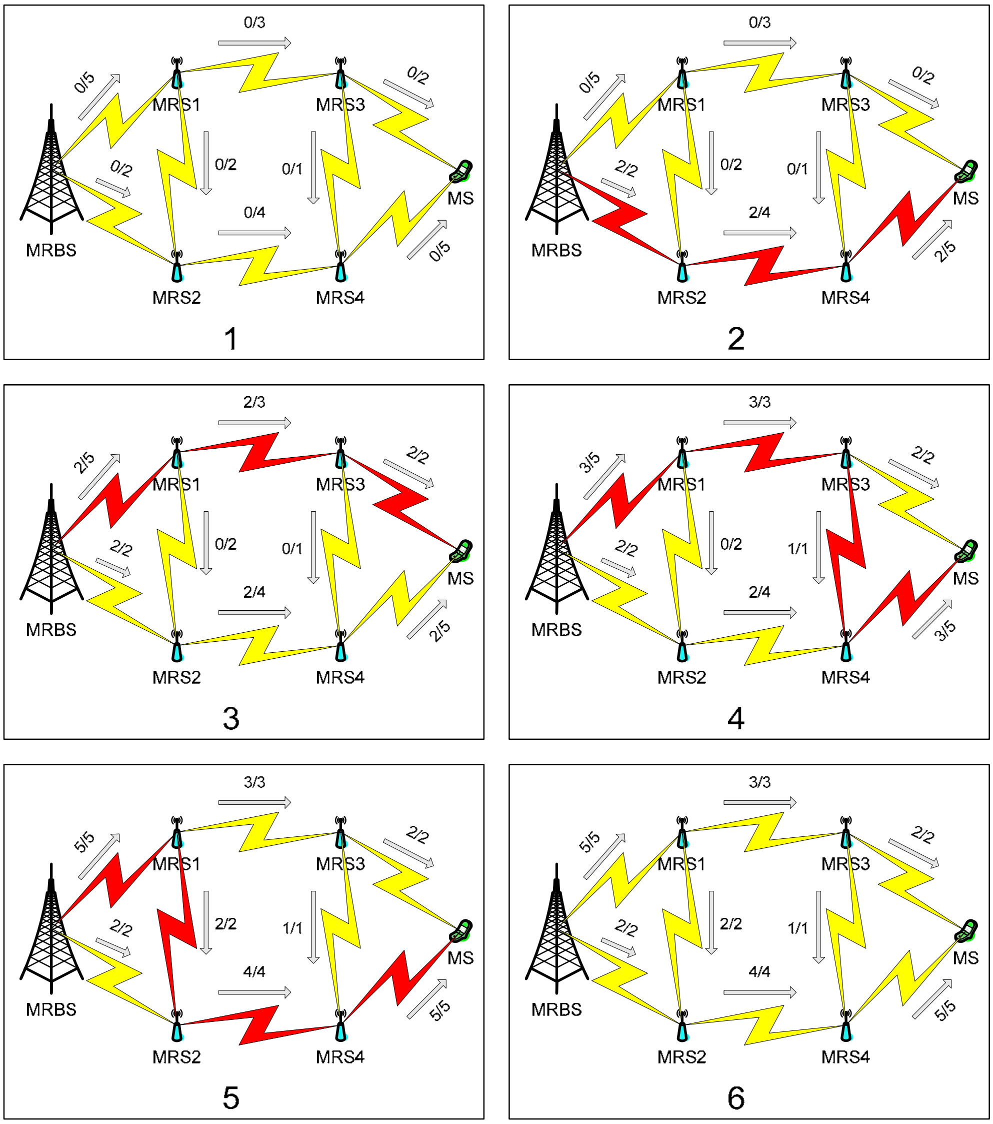


Figure 6-9 *Ford-Fulkerson's algorithm execution*

6.7 Simulation results

In order to evaluate the proposed system’s architecture and the algorithm performance, two scenarios and the results of their simulations performed in MATLAB are presented.

In the first scenario, six nodes are considered in a partially connected mesh topology, as shown in Figure 6-10, with rates as link values. “Node 1” is the source node (MRBS), while “Node 6” is the target of the transmission (MS). The rest of the nodes are the MRSs.

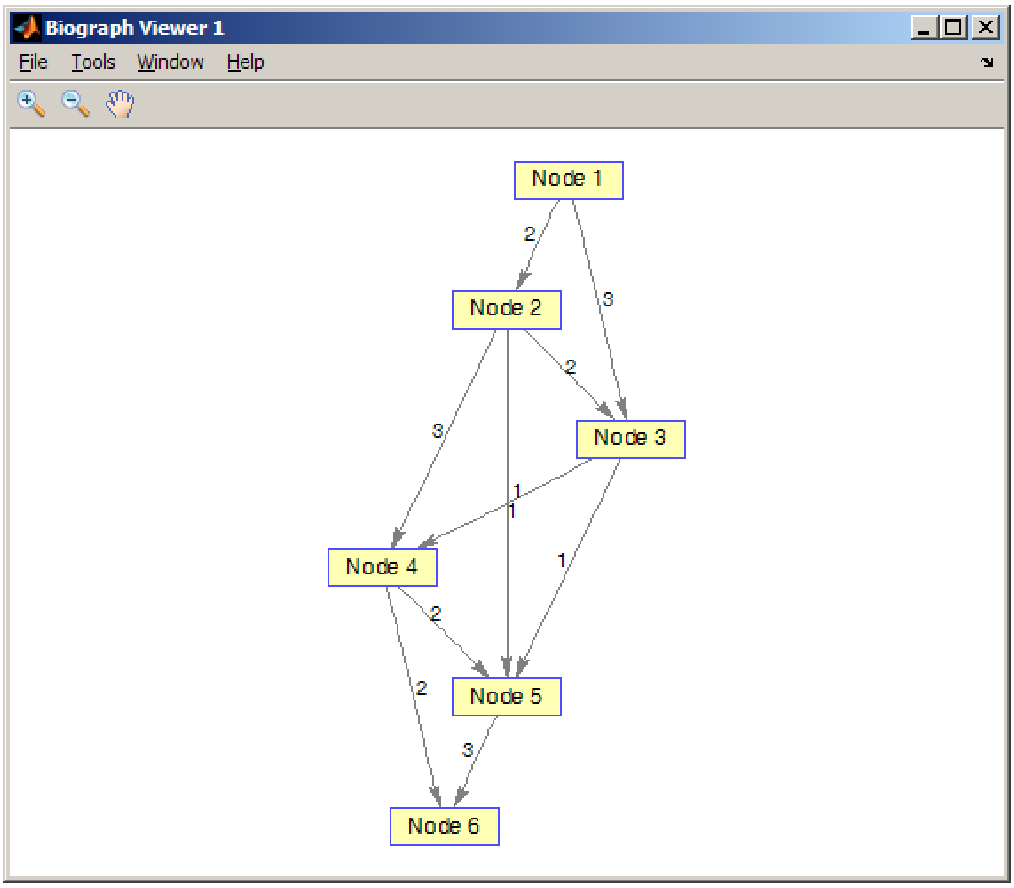


Figure 6-10 *Network topology for 1st scenario*

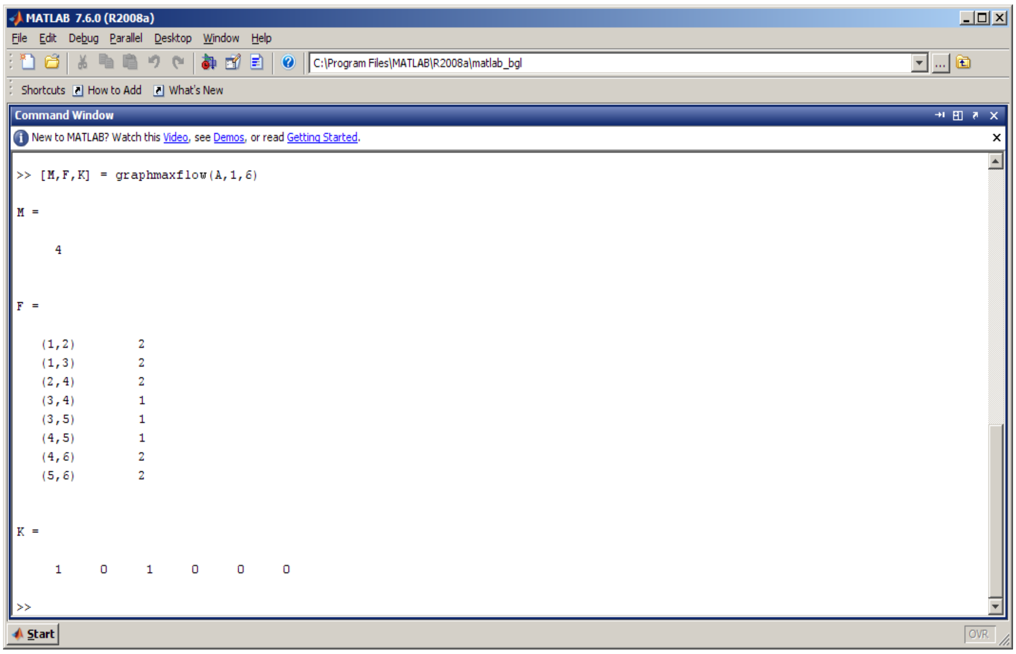
For the execution of the Ford-Fulkerson algorithm, the *graphmaxflow* function of the MATLAB *bioinfo* toolbox has been used with one modification made in the corresponding m-file. In line 61, the value of the parameter "*algorithm*" has to be set to "1", in order to implement the Edmonds and Karp version of the Ford-Fulkerson algorithm. The file includes this parameter set to "2", which is an implementation of a different algorithm (Goldberg). Therefore, line 61 becomes:

algorithm = 1; % defaults to Edmonds

In Table 6-1 the links between nodes and their values are presented, while Figure 6-11 presents a screen-capture, after the execution of the algorithm. It can be seen that the maximum value of the data flow is the value of "M", while the minimum cut is the one displayed in row vector "K". "F" is a sparse matrix containing all the flow values for every link from the source node to the target one in order to achieve the maximum data flow.

Table 6-1 *Link values*

Link	Rate
(1,2)	2
(1,3)	3
(2,3)	2
(2,4)	3
(3,4)	1
(2,5)	1
(3,5)	1
(4,5)	2
(4,6)	2
(5,6)	3



New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> [M,F,K] = graphmaxflow(A,1,6)

M =

     4


F =

   (1,2)        2
   (1,3)        2
   (2,4)        2
   (3,4)        1
   (3,5)        1
   (4,5)        1
   (4,6)        2
   (5,6)        2


K =

     1     0     1     0     0     0

>>
```

Figure 6-11 *Algorithm execution*

Examining the results of the simulation, it can be derived that the maximum value of data flow is “M=4”, while there is only one minimum cut in this network. Additionally, in order to achieve the maximum flow, the utilization of the network is the one depicted in Figure 6-12. In this case, all nodes have to transmit and all links have to be active for maximizing the flow. On the other hand, the minimum cut of the initial network is shown in Figure 6-13, where the value of the maximum flow for the minimum cut can be accomplished by deleting “Node 1” and Node 3. The sum of the values of the link rates from the deleted nodes to the rest equals to four:

$$(1,2) + (3,4) + (3,5) =$$

$$2 + 1 + 1 = 4$$

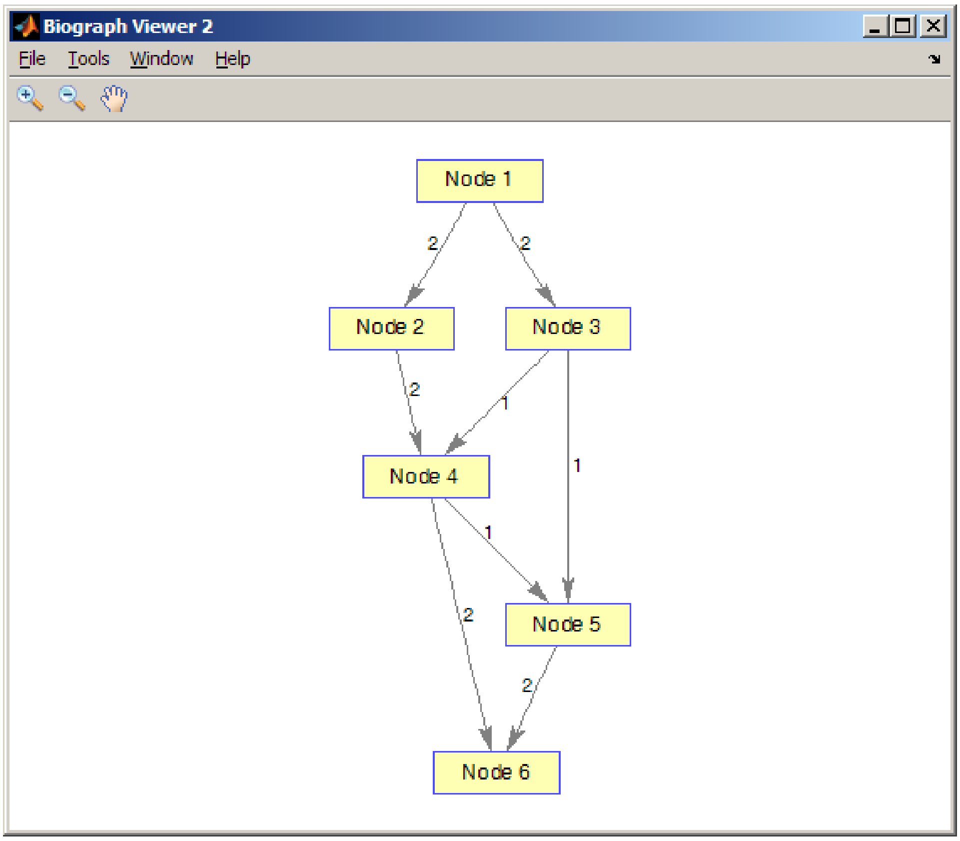


Figure 6-12 *Maximum flow*

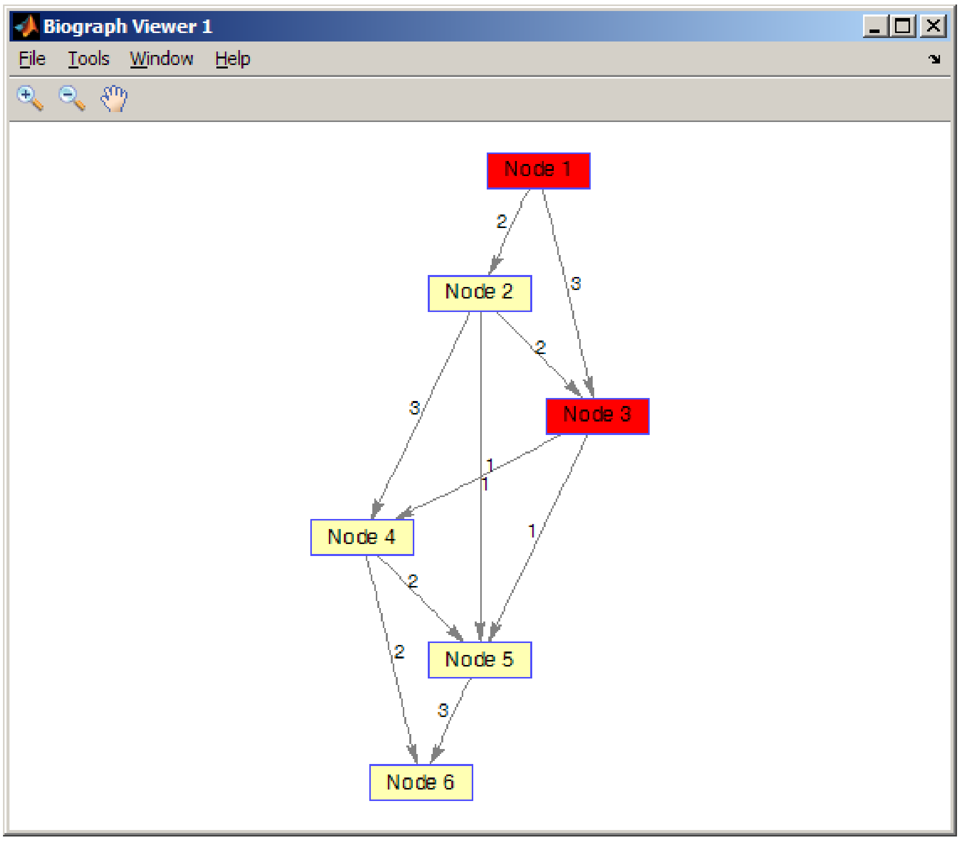


Figure 6-13 *Minimum cut*

The second scenario is slightly more complex, since eight nodes appear in a partially connected mesh topology, as shown in Figure 6-14. “Node 1” is the source node (MRBS) and “Node 8” is the target of the transmission (MS), while a direct link connecting them also exists, meaning that the source can access its target directly.

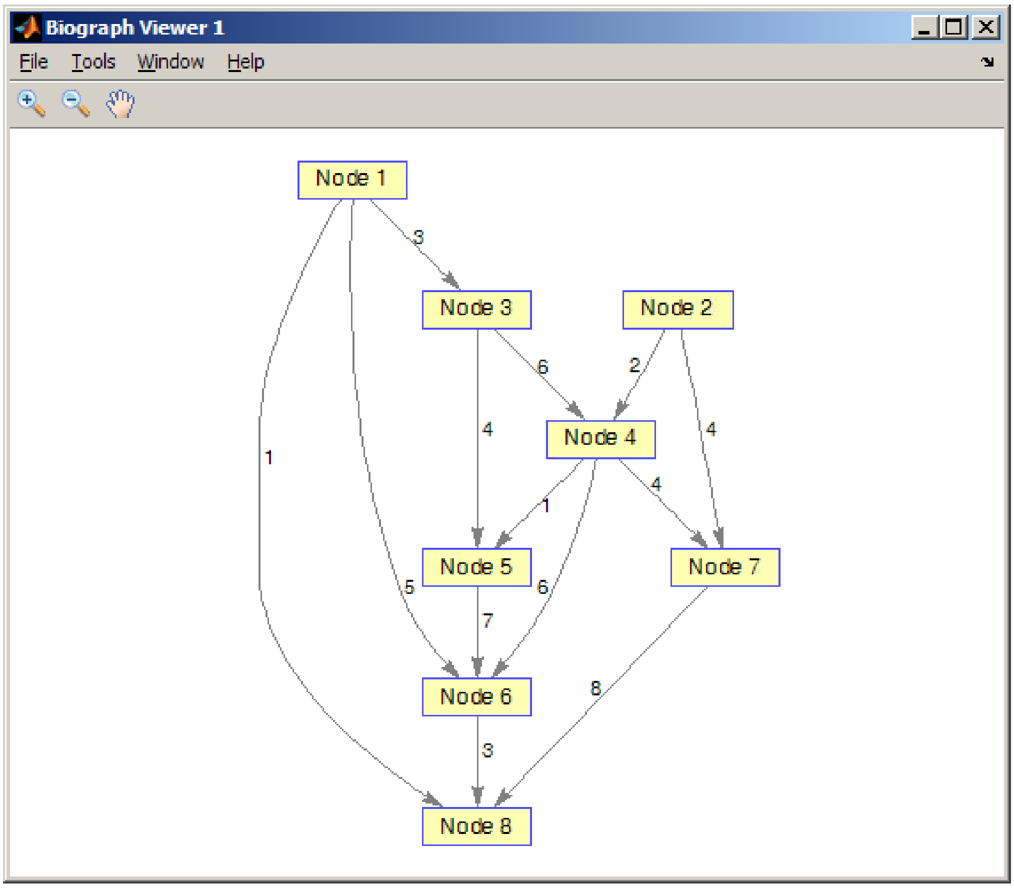
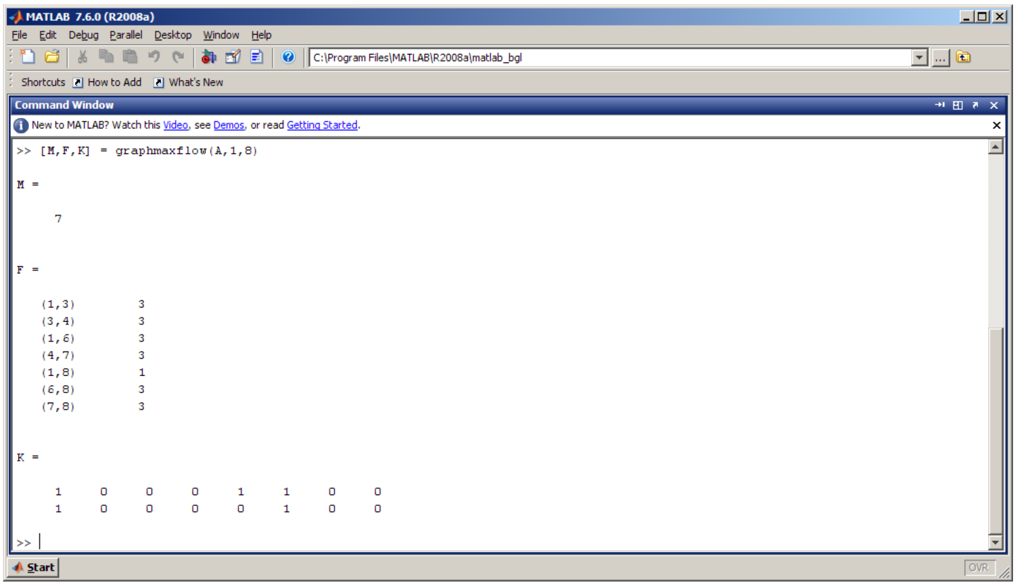


Figure 6-14 *Network topology for 2nd scenario*

For the execution of the algorithm, the same function, as the one in the first scenario of the MATLAB bioinfo toolbox, has been used with the same value modification of the parameter “*algorithm”.* In Table 6-2 the links between nodes and their values are presented, while Figure 6-15 presents a screen-capture, after the execution of the algorithm.

Table 6-2 *Link values*

Link	Rate
(1,3)	3
(2,4)	2
(3,4)	6
(3,5)	4
(4,5)	1
(1,6)	5
(4,6)	6
(5,6)	7
(2,7)	4
(4,7)	4
(1,8)	1
(6,8)	3
(7,8)	8



New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> [M,F,K] = graphmaxflow(A,1,8)

M =

     7


F =

   (1,3)        3
   (3,4)        3
   (1,6)        3
   (4,7)        3
   (1,8)        1
   (6,8)        3
   (7,8)        3


K =

     1     0     0     0     1     1     0     0
     1     0     0     0     0     1     0     0

>> 
```

Figure 6-15 *Algorithm execution*

These results can lead to the conclusion that the maximum value of data flow is “M=7”, while there are two minimum cuts in this network. In addition, in order to achieve the maximum flow, the utilization of the network is the one depicted in Figure 6-16. In this case, not all nodes have to transmit and not all links have to be active in order to maximize the flow. It can be seen that “Node 2” and “Node 5” do not participate in the transmission that achieves the best result.

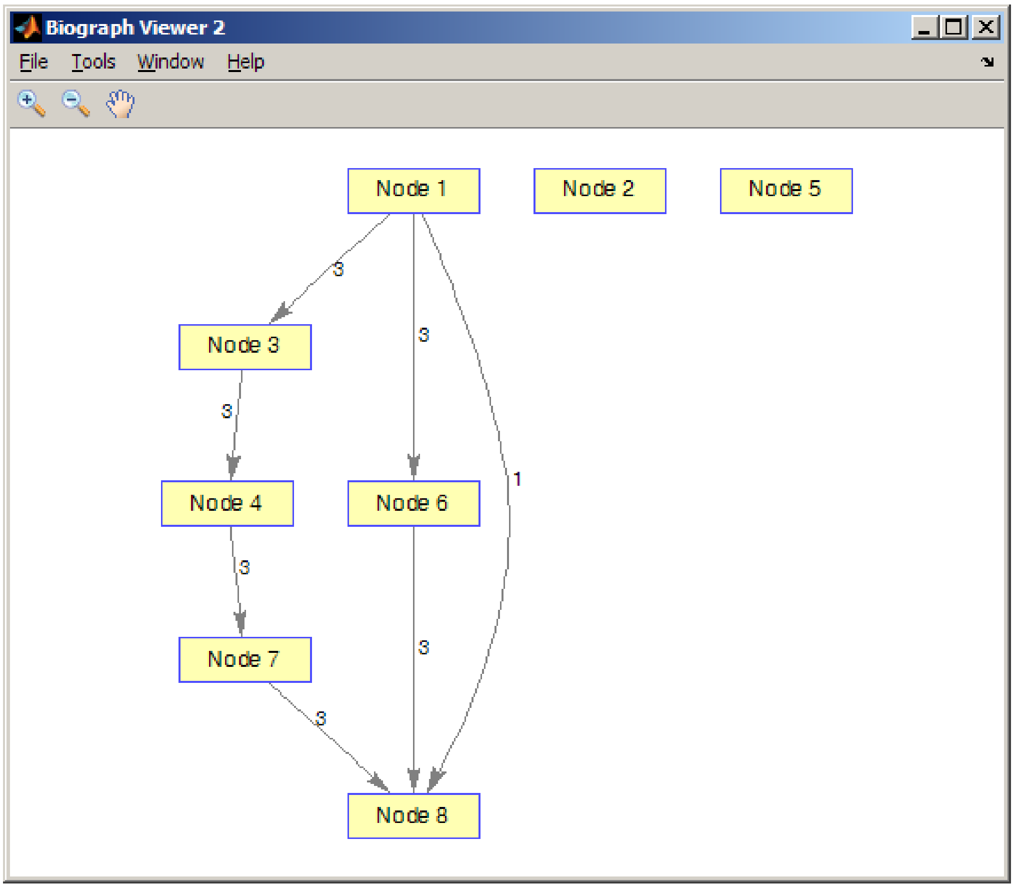


Figure 6-16 *Maximum flow*

The first of the two possible minimum cuts in the initial network is presented in Figure 6-17, where the value of the maximum flow for the minimum cut can be accomplished by deleting “Node 1” and “Node 6”. The sum of the values of link rates from the deleted nodes to the rest equals to seven:

$$(1,8) + (1,3) + (6,8) =$$

$$1 + 3 + 3 = 7$$

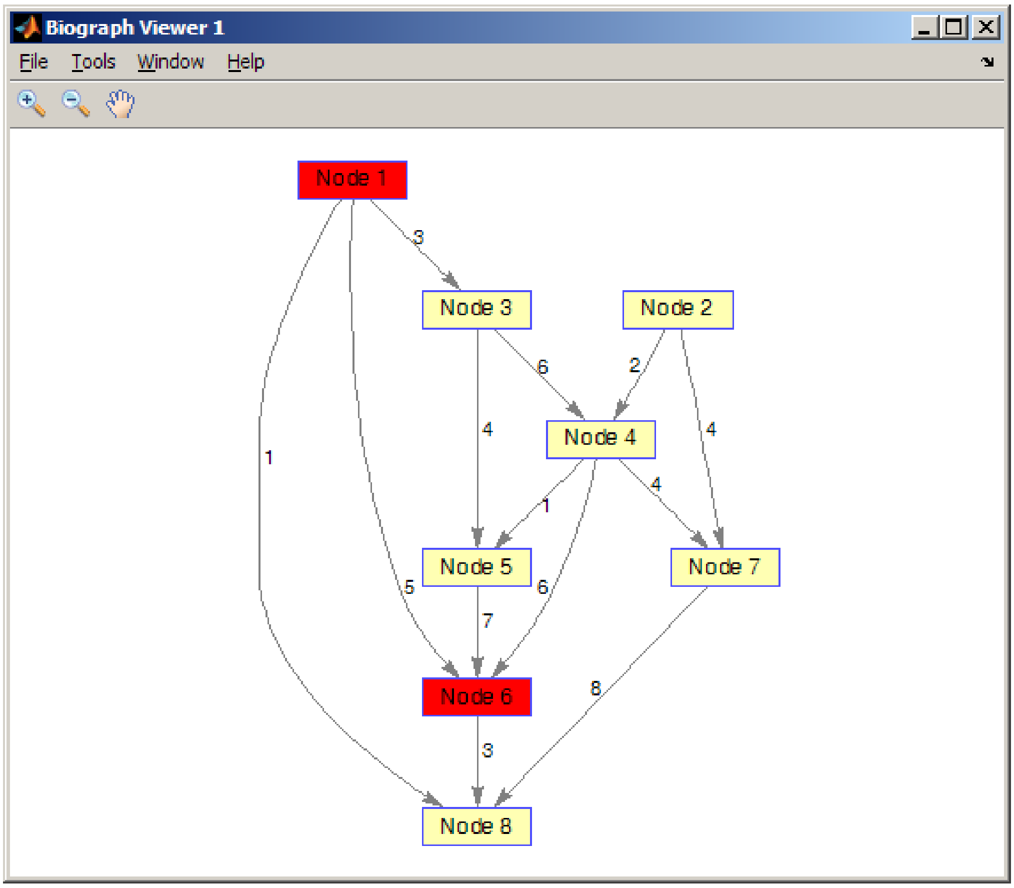


Figure 6-17 *1st minimum cut*

The second of the two possible minimum cuts in the initial network is presented in Figure 6-18, where the value of the maximum flow for the minimum cut can be accomplished by deleting “Node 1”, “Node 5” and “Node 6”. The sum of the values of link rates from the deleted nodes to the rest equals to seven:

$$(1,8) + (1,3) + (6,8) =$$

$$1 + 3 + 3 = 7$$

As it can be observed, the above equality is the same as in the first cut. The reason is that, although “Node 5” is deleted, it has no link to any other node but only to “Node 6” which is also deleted; therefore there is no other value to add to the calculation.

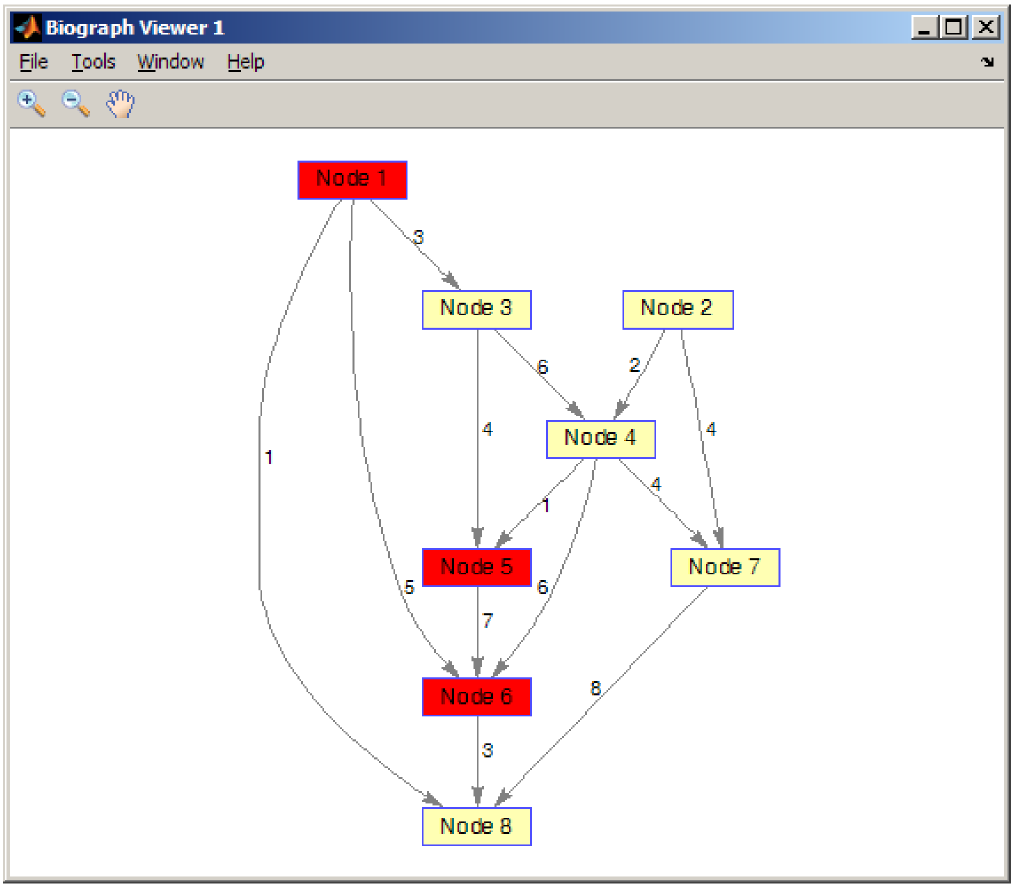


Figure 6-18 *2nd minimum cut*

Due to the results of the scenarios presented, it is concluded that the Ford Fulkerson algorithm can be used in order to maximize the data flow in a WiMAX mesh network. The paths from the source to the target node are computed, while the direct link from the MRBS to the SS is also used, if available.

However, once more, it has to be stated that adding relays in a network, it also adds time delay. Unfortunately, due to the lack of support from the current frame structure of the 802.16j standard, this cannot be studied within this research.

6.8 Summary

For the algorithm introduced in this chapter [Tsiakas *et al.*, 2009], the selected setting has been multicast transmission in a LICI environment. In particular, several end-users can be served simultaneously, while relays of the same tier can also transmit at the same time. Routing decisions regarding which MRBS will serve each mobile user are more complex in this setting, since end users are able to interact with a wider range of serving nodes. The dynamic decision about the desired link is based on updated channel knowledge, taking into consideration both the needs of the given subscriber and the service loads experienced by each available transmitter. The aim of the algorithm is to implement an efficient multi-hop routing scheme that is interference aware, and hence maximizes parallel transmission, providing at the same time high throughput and scalability.

The core element of the algorithm introduced is the maximization of the throughput of each candidate sector including the new user. The proposed algorithm acknowledges the mutual interference within each tier and offers the means to deal with it. Another advantage is that it offers an optional scheme where more demanding and costly applications optimize the backhaul network in an offline mode without considering the end-users, in which each base/relay is assumed to provide a maximal data rate at the access level.

The network topology and transmission method is described, while the classical notion of flow networks and its relation to the selected settings is explained. At first, a scheme of wireless network data flow, directly adapted to the maximum flow method of Ford-Fulkerson, is introduced. Finally, the FF procedure has been utilized as a sub-routine in an algorithm which is more tuned towards SDMA beam-forming-optimized wireless communication.

The last section of the chapter presents the simulations of two scenarios using MATLAB, in order to demonstrate the implementation of the algorithm proposed in a WiMAX mesh network using relays as nodes. The maximization in data flow has already been shown, while some questions regarding time delay have been left unanswered, due to the lack of support of the current frame structure of the 802,16j standard for more than two “hops”.

7 Conclusions – future work

7.1 Introduction

This chapter presents the conclusions of the current research and gives ideas and directions for future work on how to continue and extend it. However, a comparison of the algorithms cannot be made, since they are using different network models and assumptions. Yet, a description on how the algorithms designed can be used for load balancing is given. Finally, examples and ideas of other networks, where the algorithms introduced may be applicable, are also provided.

7.2 Conclusions

As it is clearly seen, the notion of extending the WiMAX mesh network architecture to incorporate relay stations is new and is not yet supported by the current standards. Thus, the aim of this research is to persuade other researchers in the area about the importance and usefulness of the idea. Finally, the ultimate goal is to affect standardization bodies in order to include optional mesh topology in forthcoming releases of the WiMAX standards.

The main outcomes of this research are the two algorithms developed for unicast and multicast transmission in WiMAX mesh networks. The first algorithm based on Dijkstra has been mathematically modelled and proved to be working both in a No Intra-Cell Interference (NICI) and in a Limited Intra-Cell Interference (LICI) setting. The second algorithm based on the Ford-Fulkerson and its Edmonds-Karp version has been designed and mathematically proved to be working only in a Limited Intra-Cell Interference (LICI) setting, since there can be no multicast environment if there isn't any simultaneous transmission. The latter, is an interference-aware algorithm that provides high-throughput and scalability to the system.

Two concepts have been combined to produce the network architecture, the WRMN, used in this research. The first concept was the mesh topology supported by both the IEEE 802.16-2004 and the 802.16e standards and the second was the MRSs, introduced by the IEEE 802.16j standard. The architecture of WRMNs consists of: a) a MRBS that is the source of the transmission and implements all routing algorithms; b) the MRSs that are the retransmitting nodes; c) the end-user terminals that are the target of the transmission.

The two centralized routing algorithms have been presented, in which the MRBS performs the radio resource management tasks and takes all necessary routing decisions. The main reasons for selecting a centralized scheme are a) the main network topology doesn’t change very quickly in WRMNs and b) by having centralized and localized Radio Resource Management (RRM) strategy, load sharing among MRSs can be maintained.

The general approach that has been used for both algorithms is the following. At first, a mobile which is not yet connected to the network is considered. In order to assign this mobile to a nearby sector, any cell whose closest transmitter (base or relay) provides received SINR beyond a minimal threshold (typically around 0 dB) is a candidate to serve that user. Then, the potential impact of adding the user to each of the sectors is being tested. Finally, for each of them, the throughput increase caused by the added user is assessed. The user is then assigned to the base or relay station for which the throughput increase is maximal.

This thesis introduces routing algorithms for the DL communication of two types of mesh networks. In the NICI setting, which is suitable mostly for rural areas, and in each given cell, only one transceiver can transmit at a given time and frequency. On the other hand, in the LICI setting, suitable for urban environments, each sector is split into several mutually disjoint tiers, such that the mutual intra-cell interference is low within each tier. The core assumption is that at any given time and frequency there is exactly one tier in which there can be simultaneous transmission of data.

Enabling utilization of Dijkstra algorithm for unicast transmission in NICI networks

In the case of a NICI setting, when a user enters the network, for each neighbouring candidate sector, a relay route that maximizes the overall relay-based rate is found, considering that all candidates sectors, if admitting this additional user, will not disconnect service for any of the existing users. Among these candidates the one that entails the greatest rate to the considered user is chosen and this process is done independently per user.

Due to the assumptions of a unicast NICI network there is only one active transmitter at any given time. It has been shown that the composite (normalized) rate R, based on dynamic, optimized time sharing, is given by the formula:

$$1/R = 1/R_1 + 1/R_2 + \ldots\ldots\ldots + 1/R_k.$$

Optimization of a LICI unicast network via sum-min-max algorithms

In a unicast network, as described in this thesis, each user is served at an exclusive time and a frequency resource. The network transceivers are divided into tiers with low interference to allow simultaneous transmission. During the initiation of the network, an offline-algorithm optimizes the entire network's backhaul. Once this infrastructure algorithm is complete, updates occur only when backhaul links change.

It has been shown that once backhaul optimization is established offline, dynamic frequent updates occur only with respect to the end-users. These updates amount to very light-weight executions of the Dijkstra algorithm. This procedure has been named the delta end-user optimization.

New routing algorithm introduced based on maximum graph-flow algorithms

For the algorithm proposed, multicast transmission in a LICI setting is taken under consideration. In particular, several end-users can be served simultaneously, while relays of the same tier can also transmit at the same time. The dynamic decision about which node will serve the end-user is based on updated channel knowledge, taking under consideration both the needs of the given subscriber and the service loads experienced by each available node.

The core element of the algorithm introduced is the maximization of the throughput of each candidate sector, taking into account the existing interference. It also offers an optional scheme for the backhaul network to be optimized offline, without considering the end-users.

Table 7-1 *Comparison with existing algorithms*

Routing schemes				
Authors	**Year**	**Approach**	**Advantages**	**Limitations**
Tsiakas	2009	Dijkstra's algorithm for NICI and LICI networks	Throughput maximization Handles broken links Enables parallel transmissions (for LICI networks) Light-weight executions for end-user optimizations Highly dynamic	Suitable for NICI networks Only one transceiver can transmit at a given time and frequency Unicast transmission Each hop adds delay
Tsiakas	2009	Ford-Fulkerson algorithm for LICI networks	Multicast transmission Maximization of throughput Deals with the interference within	Frame structure allows up to 2 hops Each hop adds delay Message exchange among relays

			the network Enables concurrent transmissions Highly dynamic	may add extra overhead
Shetiya & Sharma	2005	To provide QoS	Fixed routing QoS guarantees can be provided	Link failure is not handled Availability of resources needed
Wei *et al*.	2005	Performs interference - aware routing	Higher throughput Higher spectral efficiency	The metric used does not give the complete picture of the interference within the network
Jin *et al*.	2007	Extend the idea of Wei *et al*. Maximize throughput by maximizing concurrent transmissions	Traffic characteristics are taken into account The metric provides a better view of interference within the network	Several tree reconfigurations lead to extra overhead
Tao *et al*.	2005	Minimizing link interference	Easy to implement	The process of a node entering the network may lead to infinite looping

Perkins & Bhagwat	1994	Introduces sequence numbers	Routing is performed correctly Loops are prevented	Unnecessary updates of routing table Not suitable for dynamic networks
Perkins & Royer	1999	New mechanism for route detection Stores only the best next hop of a node and not the entire route	Scalable Routes are established on demand	Can have inconsistent routes
Johnson	1994	It allows source nodes to specify the route Stores the complete route	No need periodic update of routing tables Routes are established on demand	Broken links are not handled

7.2.1 Publications and contributions

In the beginning of the thesis there is a list of publications based on this study. More explicitly, a chapter in a book published by Springer, an article in a journal and eight conference papers, three of which were IEEE conferences, are included. Additionally, there is one more article submitted in the IET Electronic Letters and is currently under review. Another target is to submit at least two more articles in journals and more than four papers in conferences, in order to be published. Currently, the two journal articles are being written and are expected to be completed by the end of this year.

This research was performed within the context of the FP7-REWIND project and has led to significant results adopted by the consortium members. Afterwards, the consortium expanded both the results and the scope of the research and contributed a part of them to the standardisation bodies. The accepted contribution - mostly affected by the research - is the third one of the listed items in the beginning of the thesis, under the title: “*Improving the performance of DSA and DSC transaction in multi-hop relay systems with RSs operating in distributed scheduling mode. The performance is improved by specifying that the MR-BS shall send a DSA-ACK to all the RSs on the path together with only the modified service flow parameter.*”

In the proposed routing algorithms, there is a scheme that can be executed online for admitting a new user, where the rates of the sector’s transmitters with respect to that user, are being assessed along with the impact on the previously computed rates. An optimization, taking into account the previous state of the network, occurs calculating only the difference that the new user will impose. The idea of taking under consideration only the change in the network triggered the above-mentioned contribution. The outcome was that, for a distributed architecture, the MRBS should send the DSA - ACK message to all the MRSs of a path only with the changes occurred to the service flow.

As described in section 2.2.2, service flows can be either dynamic or static. Dynamic service flows may be managed by a series of MAC messages known as Dynamic Service Addition (DSA) for creating a new service flow, Dynamic Service Change (DSC) for changing an existing flow, and Dynamic Service Deletion (DSD) for deleting an existing service flow.

7.2.2 Load balancing aspect

Load balancing is the process where a routing algorithm distributes traffic among nodes. In the literature there are two major categories: multi-path and single-path approaches. Since our network is of mesh topology, our routing algorithms implement a multi-path approach. This means that nodes are linked together by more than one routes and the traffic between nodes is routed through different paths. Load-balancing includes the estimation of data forwarded on each route, by minimizing a certain cost function, with the aim of fair distribution of load to nodes.

In the algorithms presented in this research, each time the cost function is different. In the first case of unicast transmission in NICI WRMNs, the algorithm maximizes the overall relay-based rate by minimizing the respective composite data rate. In the case of unicast transmission in LICI WRMNs, the minimal relay-path that utilizes the maximal composite rate available for each end-user has to be found. In the last case of multicast transmission in LICI WRMNs, the aim is to find the cut across the network that has the minimum value over all possible cuts. In this way, the throughput of each candidate sector is maximized. In all cases, though, it has been demonstrated that the algorithms function properly and the mathematical models are solid.

7.3 Future work

7.3.1 Further algorithmic research

The whole idea of WRMNs introduced is completely new and of great interest. The value of mesh networks is unchallengeable and relays seem to be very important for future networks [Agapiou, 2009]. Relays as a concept are also introduced nowadays in LTE-Advanced, another wireless technology, which is considered the rival of WiMAX. Therefore, no matter which technology will prevail, it is almost certain that it will incorporate relay stations as a network element, thus, the need for further research in this technology is unquestionable.

Although the algorithms presented are proved to be working properly, research can continue to this direction. There are many more existing routing algorithms that can be tested if they work with the settings presented in this thesis, and of course, there is always room for new algorithms to be developed.

Additionally, it would be of great interest to find more cost functions that can work as well as the ones already detailed in this thesis. For example, the SINR or BER of a signal could be compared to a minimum threshold. This could be used even in combination with the methods proposed in this research. Moreover, prioritization of traffic combined with multiple thresholds could provide quite different and valuable results, because this could also affect the process and the way load balancing is triggered. Therefore, the ideas presented can lead to a more complete or even different load balancing schemes that will take into account many parameters and will provide solutions to deal with most of the situations that may arise during the operation of a network.

Finally, it could be tested whether the proposed algorithms can be extended to be used in other networks as well. For example, with a few modifications, they could be working in systems such as wireless sensor networks, LTE-Advanced networks, Wi-Fi networks etc.

7.3.2 Simulations

Unfortunately, for the algorithms designed, network-level simulation results cannot be provided for two reasons. The first one is that the networks under investigation throughout this research have been of mesh topology and, although the current IEEE 802.16j standard which aims at adding relaying functionality in WiMAX networks is based on the IEEE 802.16e-2005 standard and it is fully compatible with it, it does not support mesh topology. The second one is that current versions of simulators support neither the IEEE 802.16j standard nor the concept of relay stations. So, the mesh topology of previous standards cannot be used in combination with the concept of relay stations to perform simulations.

Table 7-2 lists the most popular simulators and the standards they support. It is apparent than none of them supports the IEEE 802.16j. Therefore, it is upon future research to perform simulations and provide results in order to strengthen the points outlined in this research by demonstrating the performance of the developed routing algorithms in WiMAX mesh networks.

Table 7-2 *List of the most popular simulators*

Vendor	Simulator	Standard supported
OPNET Technologies, Inc.	OPNET Modeler® Wireless Suite	IEEE 802.16-2004 and IEEE 802.16e-2005
Freeware	ns-2 Simulator	IEEE 802.16-2004 and IEEE 802.16e-2005
EDX Wireless	EDX® SignalPro®	IEEE 802.16-2004 and IEEE 802.16e-2005
Forsk	Atoll WiMAX	IEEE 802.16-2004 and IEEE 802.16e-2005
CelPlan Technologies	CelPlanner™ Suite	IEEE 802.16-2004 and IEEE 802.16e-2005
ATDI	ICS telecom nG	IEEE 802.16-2004 and IEEE 802.16e-2005
Scalable Network Technologies (SNT)	QualNet Developer	IEEE 802.16-2004 and IEEE 802.16e-2005
AWE Communications	WinProp	IEEE 802.16-2004 and IEEE 802.16e-2005
SIRADEL	VOLCANO	IEEE 802.16-2004 and IEEE 802.16e-2005

References

Agapiou G., Voudouris K., Tsiakas P., Chochliouros I.: “"Networks transformation by using advanced broadband topologies with wireless relay stations”, 48th Fitce Congress, Prague, Czech Republic, September 3-5, 2009.

Akyildiz I. F., Wang X. and Wang W. (2005), “Wireless mesh networks: a survey”, Computer Networks, v.47 pp.445-487.

Chen K. C. and Marca R. (2008), “Mobile WiMAX”, 1st edition, Wiley & Sons ltd, ISBN: 978-0470519417.

Chochliouros I. P., Mor A., Voudouris K. N., Agapiou G., Aloush A., Belesioti M., Sfakianakis E. and Tsiakas P., “A Multi-Hop Relay Station Software Architecture Design, on the Basis of the WiMAX IEEE 802.16j Standard”, IEEE 69th Vehicular Technology Conference, Barcelona, Spain, April 26–29, 2009.

Chochliouros I., Mor A., Voudouris K., Agapiou G., Belesioti M., Sfakianakis E. and Tsiakas P., “Multihop Relay Stations: An MRBS-RS Link-Level Performance”, ICT-MobileSummit 2009 Conference Proceedings, Santander, Spain, June 10 – 12, 2009, Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management Corporation, 2009, ISBN: 978-1-905824-12-0.

Chochliouros I., Mor A., Voudouris K., Amrani O. and Agapiou G. (2009) “A Mobile Multi-hop Relay Base Station (MRBS) – Relay Station (RS) Link Level Performance of Coding/Modulation Schemes, on the Basis of the REWIND Research Program”, in Mobile Lightweight Wireless Systems, Springer, 1st edition, pp. 93-102, ISBN: 978-3642038181.

Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Stein, Clifford (2001). “Introduction to Algorithms”, 2nd edition, MIT Press and McGraw-Hill. ISBN 0-262-53196-8

Cover T. M. and Thomas J. A. (2006), “Elements of Information Theory”, Wiley-Interscience, 2nd edition, ISBN: 978-0471241959.

Dijkstra E. W. (1959), “A note on two problems in connexion with graphs”. In Numerische Mathematik, 1, S., pp. 269–271.

European Commission (2007), “Progress report on the single European electronic communications market 2007”, Available at: http://ec.europa.eu/information_society/policy/ecomm/doc/library/annualreports/13th/com_2008_153_en_final.pdf [Last accessed 12th July 2009]

Ford, L. R and Fulkerson, D. R. (1956), “Maximal flow through a network”. Canadian Journal of Mathematics 8, pp. 399-404.

Held, Gilbert (2005), “Wireless Mesh Networks”, Auerbach publications, 1st edition, ISBN: 978-0849329609

Hossain E. and Leung K.K. (2008), “Wireless Mesh Networks: Architectures and Protocols”, 1st edition, Springer.

IEEE 802.16 Broadband Wireless Metropolitan Area Network Standards, Available at: http://standards.ieee.org/getieee802/802.16.html [Last accessed 10th September 2009]

IEEE 802.16's Relay Task Group, Available at: http://www.ieee802.org/16/relay/, [Last accessed 10th September 2009]

IEEE Standard for Local and metropolitan area networks, “Part 16: Air Interface for Broadband Wireless Access Systems. Amendment 1: Multiple Relay Specification”, June 2009

Jin F., Amrinder A. Hwang J. and Choi H., “Routing and packet scheduling in WiMAX mesh networks,” Proceedings of Fourth International Conference on Broadband Communications, Networks and Systems, BROADNETS 2007, Raleigh, North Carolina, USA, Sept. 10-14, 2007, pp.574-582.

Johnson D., “Routing in Ad Hoc Networks of Mobile Hosts.”, Proceedings of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, Santa Cruz, CA, December 1994, pp. 158-163.

Kyungtae K., Hong S., “VoMESH: voice over wireless MESH networks”, in IEEE Wireless Communications and Networking Conference 2006, April 3-6, 2006, Las Vegas, USA, pp. 193-198.

McConnell, Jeffrey (2007), “Analysis of Algorithms”, Jones & Bartlett Pub, 2nd edition, ISBN: 978-0763707828

Ohrtman F. (2003), “Wi-Fi Handbook: Building 802.11b Wireless Networks”, McGraw-Hill Professional, 1st edition ISBN: 978-0071412513

Osterloh H. (2002), “IP Routing primer plus”, Sams Publishing, 1st edition, ISBN: 978-0672322105

Perkins C. and Bhagwat P., “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, Proceedings of the ACM SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications, August 31 - September 2, 1994, London, UK, pp. 234-244.

Perkins C. and Royer E., “Ad hoc On-Demand Distance Vector Routing.”, Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 25-26, 1999, pp. 90-100.

Samsung (2009), What is Mobile WiMAX? Available at: http://www.samsung.com/global/business/telecommunication/productInfo.do?ctgry_group=11&ctgry_type=16&b2b_prd_id=40 [Last accessed 9th September 2009].

Senza Fili Consulting (2007), “Meeting the Challenges of High-Capacity Indoor and Outdoor Coverage” November 2007, (Internal FP7-REWIND report).

Shetiya H. and Sharma V., “Algorithms for Routing and Centralized Scheduling to Provide QoS in IEEE 802.16 Mesh Networks”, Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modelling, Montreal, Quebec, Canada, October, 2005, pp. 140 – 149

Tao J., Liu F., Zeng Z. and Lin Z., “Throughput enhancement in WiMax mesh networks using concurrent transmission,” Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, Sept. 23-26, 2005., vol.2, pp. 871-874.

Tsiakas P., Dor A., Voudouris K., Hadjinicolaou M., “Load Balancing in Limited Intra-Cell Interference (LICI) Networks Based on Maximum Graph-Flow Algorithms”, Proceedings of IEEE International Conference on Ultra Modern Telecommunications (ICUMT 2009), St-Petersburg, Russia, October 12-14, 2009, ISBN: 978-1-4244-3941-6

Voudouris K. N., Chochliouros I. P., Tsiakas P., Mor A., Agapiou G., Aloush A., Belesioti M. and Sfakianakis E. (2009): “Developing an Innovative Multi-Hop Relay Station Software Architecture in the Scope of the REWIND European Research Programme”, in Mobile Lightweight

Wireless Systems, Springer, 1st edition, pp.160-172, ISBN: 978-3642038181.

Wei H., Gangulu S., Izmailov R. and Haas Z., “Interference-Aware IEEE 802.16 WiMax Mesh Networks”, Proceedings of 61st IEEE Vehicular Technology Conference (VTC 2005 Spring), Stockholm, Sweden, May 29-June 1, 2005, pp. 3102-3106.

WiMAX Forum (2005), Can WiMAX Address Your Applications? Available at: http://www.wimaxforum.org/technology/downloads/Can_WiMAX_Address_Your_Applications_final.pdf. [Last accessed 7th August 2009].

WiMAX Forum (2006), “Mobile WiMAX – Part I: A Technical Overview and Performance Evaluation”, Available at: http://www.wimaxforum.org/sites/wimaxforum.org/files/document_library/Mobile_WiMAX_Part1_Overview_and_Performance.pdf [Last accessed 9th August 2009]

WiMAX Forum (2006), “Mobile WiMAX – Part II: A Comparative Analysis”, Available at: http://www.wimaxforum.org/sites/wimaxforum.org/files/document_library/mobile_wimax_part2_comparative_analysis.pdf [Last accessed 9th August 2009].

Xiao, Yang (2007), “WiMAX/MobileFi: Advanced Research and Technology”, Auerbach publications, 1st edition, ISBN: 978-1420043518

Appendix I – Publications and Contributions

Chapter in a book

- Konstantinos N. Voudouris, Ioannis P. Chochliouros, **Panagiotis Tsiakas**, Avishay Mor, George Agapiou, Avner Aloush, Maria Belesioti and Evangelos Sfakianakis: “Developing an Innovative Multi-Hop Relay Station Software Architecture in the Scope of the REWIND European Research Programme”, Mobile Lightweight Wireless Systems, Springer, 1 edition (October 1, 2009), pp.160-172, ISBN: 978-3642038181

Publications in journals

- G. Agapiou, K. Voudouris, **P. Tsiakas**, A. Rigas: “Advanced Relay Architectures for Network Enhancement”, Institute of Telecommunications Professionals Journal (to be appeared)

Publications in conferences

- I Petropoulos, K. Voudouris, **P. Tsiakas**, I. Georgas, K. Vergos, G. Agapiou: "A Business Model Analysis for WiMAX Relay Station Networks", Future Network & Mobile Summit 2010 Conference, Florence-Italy, 16-18 June 2010 (accepted for publication).

- N. C. Athanasopoulos, **P. Tsiakas**, K. N. Voudouris, I. Georgas: "Multi-hop Relay in Next Generation Wireless Broadband Access Networks: An Overview", MOBILIGHT 2010, Barcelona Spain 10-12 May 2010 (accepted for publication).

- N. C. Athanasopoulos, **P. Tsiakas**, K.N. Voudouris, D. Manor, A Mor and G. Agapiou: “An IEEE 802.16j Prototype Relay Station Architecture”, 15th IEEE Mediterranean Electrotechnical Conference, Valletta, Malta, 26 - 28 April 2010 (accepted for publication).

- **Panagiotis Tsiakas**, Avner Dor, Konstantinos Voudouris, Marios Hadjinicolaou: “Load Balancing in Limited Intra-Cell Interference (LICI) Networks Based on Maximum Graph-Flow Algorithms”, 2009 International Conference on Ultra Modern Telecommunications

(ICUMT 2009), St-Petersburg, Russia, October 12-14, 2009, ISBN: 978-1-4244-3941-6.

- G. Agapiou, K. Voudouris, **P. Tsiakas**, I. Chochliouros: “"Networks transformation by using advanced broadband topologies with wireless relay stations”, 48th Fitce Congress 2009, Prague, Czech Republic, September 3-5, 2009.

- Ioannis Chochliouros, Avishay Mor, Konstantinos Voudouris, George Agapiou, Maria Belesioti, Evangelos Sfakianakis, **Panagiotis Tsiakas**: “Multihop Relay Stations: An MRBS-RS Link-Level Performance”, ICT-MobileSummit 2009 Conference Proceedings, 10 - 12 June 2009, Santander, Spain, Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management Corporation, 2009, ISBN: 978-1-905824-12-0.

- Konstantinos N. Voudouris, Ioannis P. Chochliouros, **Panagiotis Tsiakas**, Avishay Mor, George Agapiou, Avner Aloush, Maria Belesioti and Evangelos Sfakianakis: “Developing an Innovative Multi-Hop Relay Station Software Architecture in the Scope of the REWIND European Research Programme”, 1st International Conference on Mobile Lightweight Wireless Systems, Athens, Greece, May 18-20, 2009.

- I. P. Chochliouros, A. Mor, K. N. Voudouris, G. Agapiou, A. Aloush, M. Belesioti, E. Sfakianakis and **P. Tsiakas**: “A Multi-Hop Relay Station Software Architecture Design, on the Basis of the WiMAX IEEE 802.16j Standard”, IEEE 69th Vehicular Technology Conference, Barcelona, Spain, 26–29 April 2009.

Papers under review

- N.C. Athanasopoulos, **P. Tsiakas**, K.N. Voudouris, D. Manor, A Mor and G. Agapiou: “An IEEE802.16j Single Unit Relay Station Architecture”, IET Electronic Letters (submitted, under review).

Contributions

- Mandating that in multi-hop relay systems with RSs operating in distributed scheduling mode, upon receiving a DSA-REQ from its super-ordinate station to request for admission control decision, an RS should reply with a DSA-RSP to MR-BS.

- Restricting the transmission of MR HARQ Error Report header by RS to MR-BS or super-ordinate RS as an unsolicited feedback in UL relay zone. Correct ARQ mechanism and state machine in hop-by-hop mode: As defined in current draft 16j/D7, for non transparent mode in distributed scheduler in ARQ Hop-by-hop mode the current mechanism constrains a waiting time when MRBS should wait to the MS ACK. Since the link is managed hop by hop there is no point that the MRBS waits for an MS ACK in order to update the TX window state and the transmitted acknowledged by the R-ACK blocks. When the MRBS receives R-ACK it should change the ARQ acknowledged blocks to done state. The MRBS should wait for MS ACK only to release the acknowledged buffers. The MRBS should release transmitted buffers only after their ARQ Blocks where acknowledged by MS-ACK.

- Improving the performance of DSA and DSC transaction in multi-hop relay systems with RSs operating in distributed scheduling mode. The performance is improved by specifying that the MR-BS shall send a DSA-ACK to all the RSs on the path together with only the modified service flow parameter.

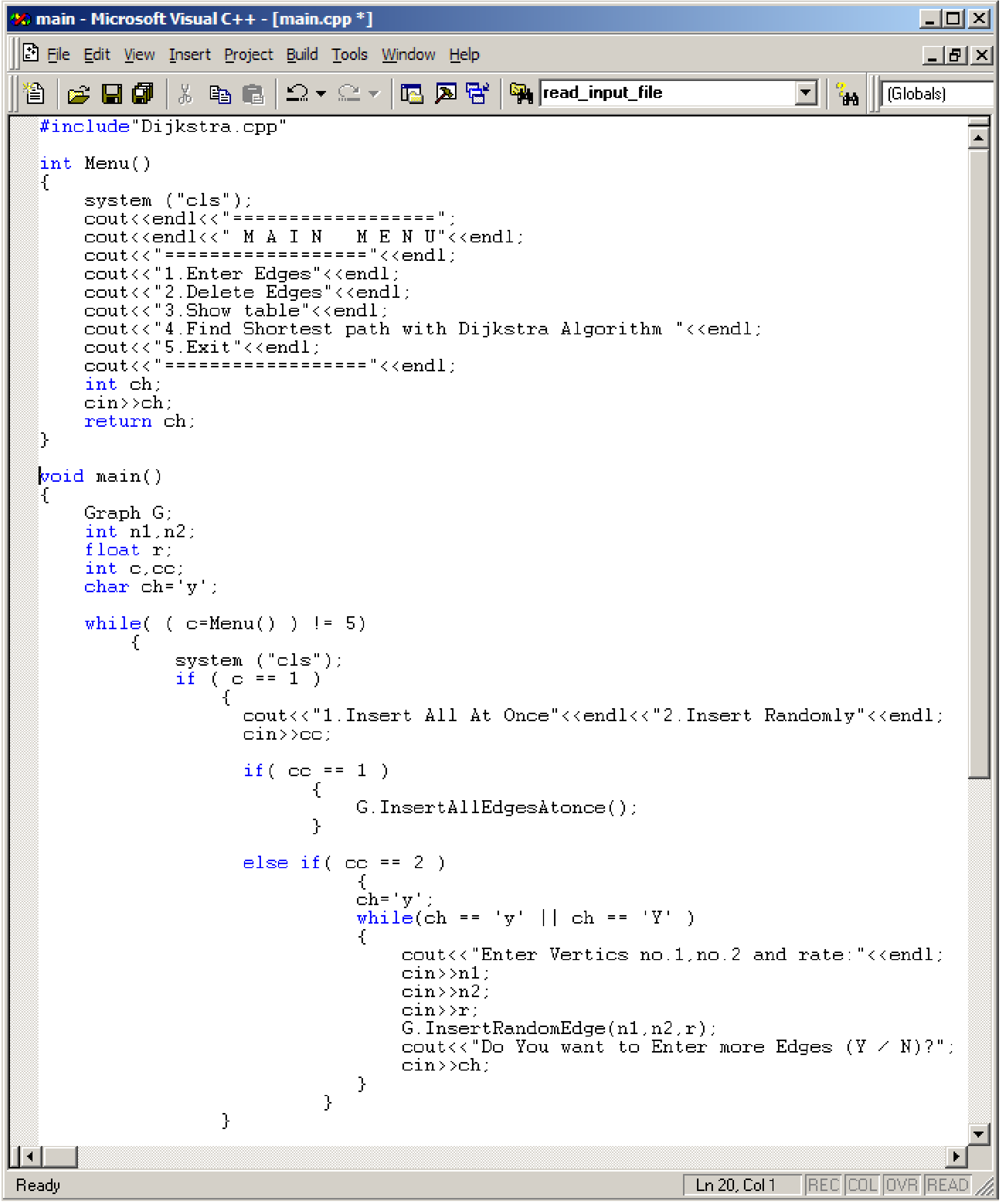
Appendix II – Abbreviations

Abbreviation	***Meaning***
AODV	Ad-hoc On-Demand Distance Vector
BTS	Base Transceiver Station
BWA	Broadband Wireless Access
DL	DownLink
DSA	Dynamic Service Addition
DSC	Dynamic Service Change
DSD	Dynamic Service Deletion
DSDV	Destination- Sequenced Distance Vector
DSR	Dynamic Source Routing
DV	Distance Vector
FF	Ford-Fulkerson
IS-IS	Intermediate System to Intermediate System
LAN	Local Area Network
LICI	Limited Intra-Cell Interference
LLC	Logical Link Control
LOS	Line-Of-Site
LS	Link State
MAC	Medium Access Control
MAN	Metropolitan Area Networks
MIMO	Multiple Input/Multiple Output
MMR	Mobile Multi-hop Relay
MRBS	Multi-hop Relay Base Station
MRS	Multi-hop Relay Station
NICI	No Intra-Cell Interference
NLOS	Non-Line-Of-Site
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OSPF	Open Shortest Path First
PHY	Physical Layer
QoS	Quality of Service

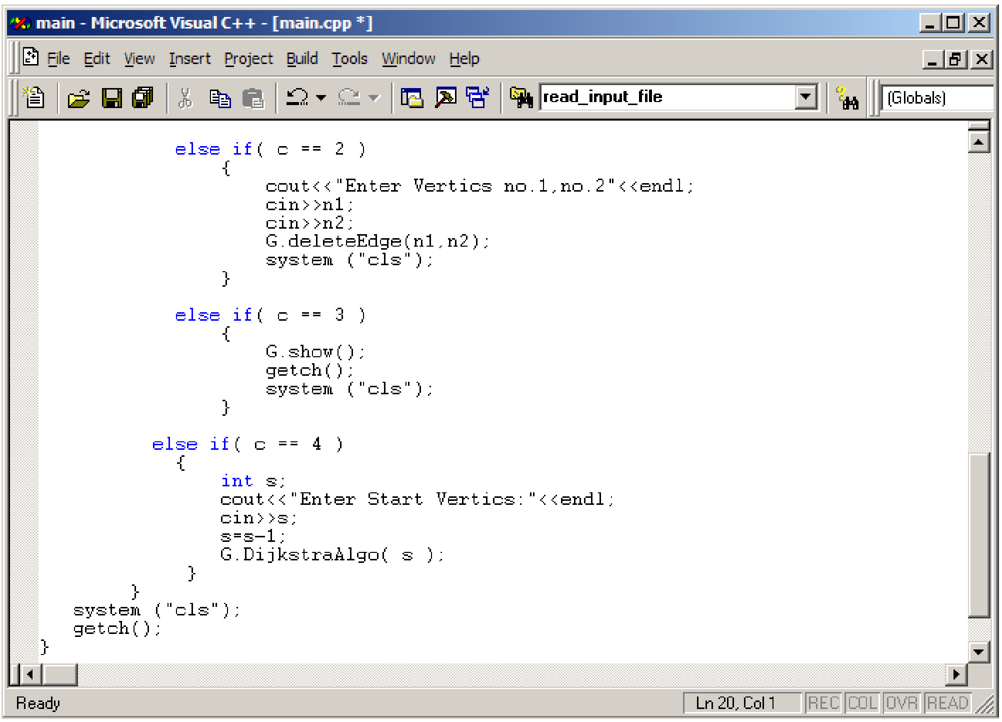
RIP	Routing Information Protocol
RRM	Radio Resource Management
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WMN	Wireless Mesh Network
WRMN	Wireless Relay Mesh Network

Appendix II – Source code

/*********************Main.cpp****************************/

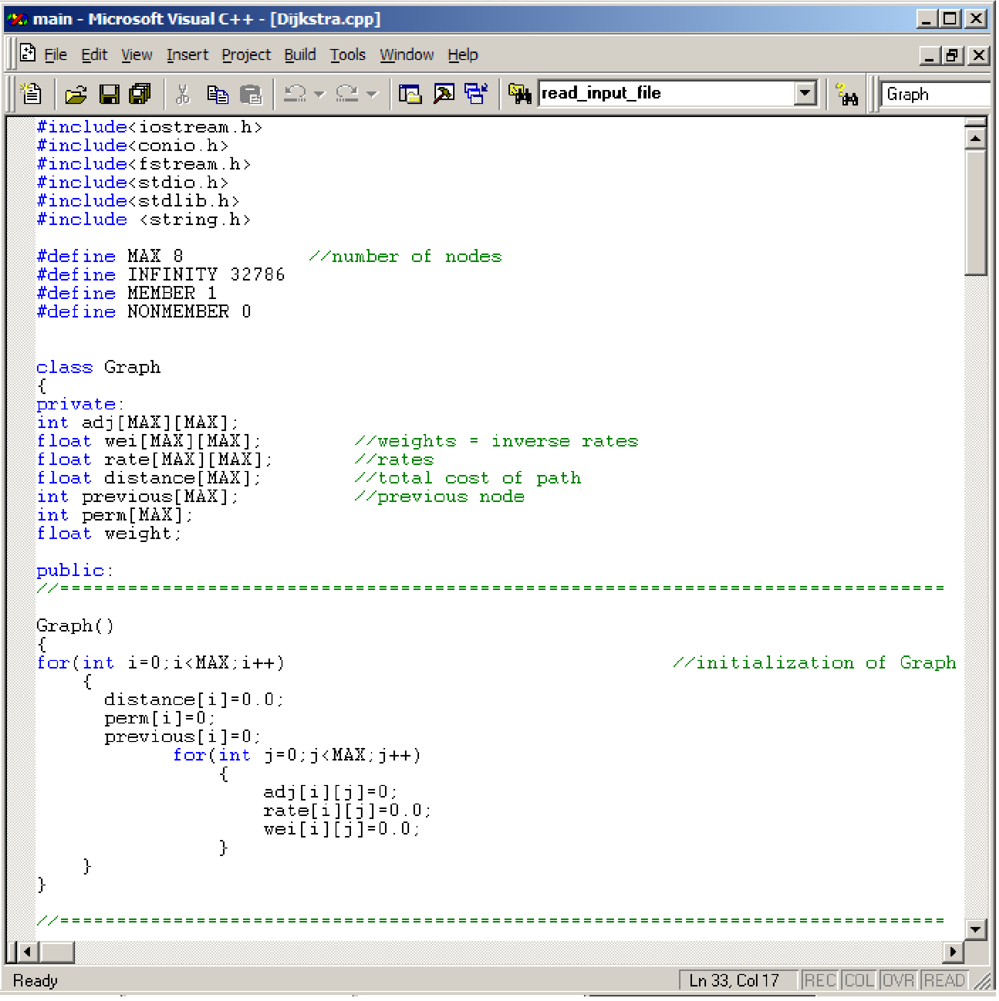


```
#include"Dijkstra.cpp"

int Menu()
{
    system ("cls");
    cout<<endl<<"==================";
    cout<<endl<<" M A I N   M E N U"<<endl;
    cout<<"=================="<<endl;
    cout<<"1.Enter Edges"<<endl;
    cout<<"2.Delete Edges"<<endl;
    cout<<"3.Show table"<<endl;
    cout<<"4.Find Shortest path with Dijkstra Algorithm "<<endl;
    cout<<"5.Exit"<<endl;
    cout<<"=================="<<endl;
    int ch;
    cin>>ch;
    return ch;
}

void main()
{
    Graph G;
    int n1,n2;
    float r;
    int c,cc;
    char ch='y';

    while( ( c=Menu() ) != 5)
        {
            system ("cls");
            if ( c == 1 )
                {
                  cout<<"1.Insert All At Once"<<endl<<"2.Insert Randomly"<<endl;
                  cin>>cc;

                  if( cc == 1 )
                        {
                            G.InsertAllEdgesAtonce();
                        }

                  else if( cc == 2 )
                            {
                            ch='y';
                            while(ch == 'y' || ch == 'Y' )
                            {
                                cout<<"Enter Vertics no.1,no.2 and rate:"<<endl;
                                cin>>n1;
                                cin>>n2;
                                cin>>r;
                                G.InsertRandomEdge(n1,n2,r);
                                cout<<"Do You want to Enter more Edges (Y / N)?";
                                cin>>ch;
                            }
                         }
                }
```

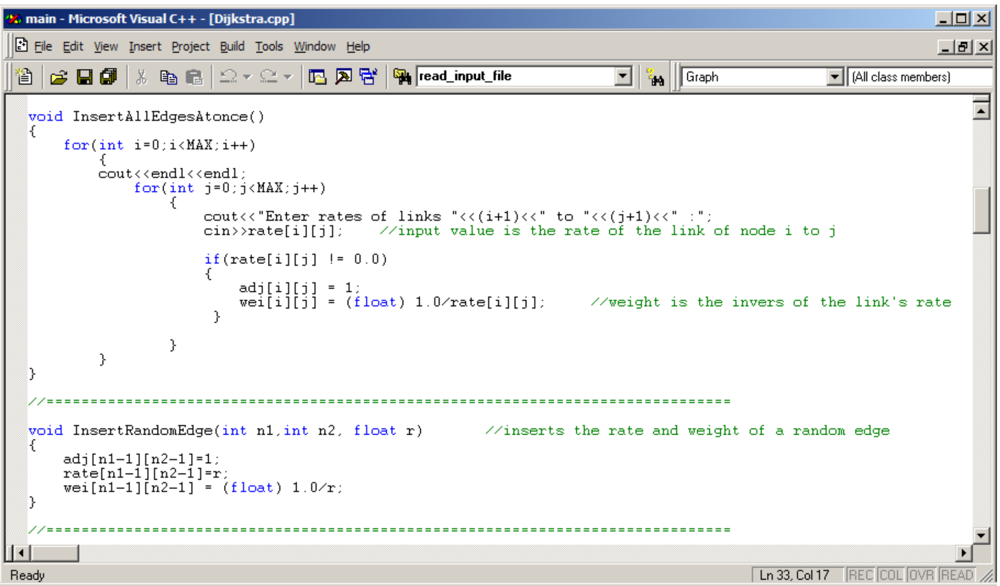


```
            else if( c == 2 )
                {
                    cout<<"Enter Vertics no.1,no.2"<<endl;
                    cin>>n1;
                    cin>>n2;
                    G.deleteEdge(n1,n2);
                    system ("cls");
                }

            else if( c == 3 )
                {
                    G.show();
                    getch();
                    system ("cls");
                }

          else if( c == 4 )
            {
                int s;
                cout<<"Enter Start Vertics:"<<endl;
                cin>>s;
                s=s-1;
                G.DijkstraAlgo( s );
             }
        }
   system ("cls");
   getch();
}
```

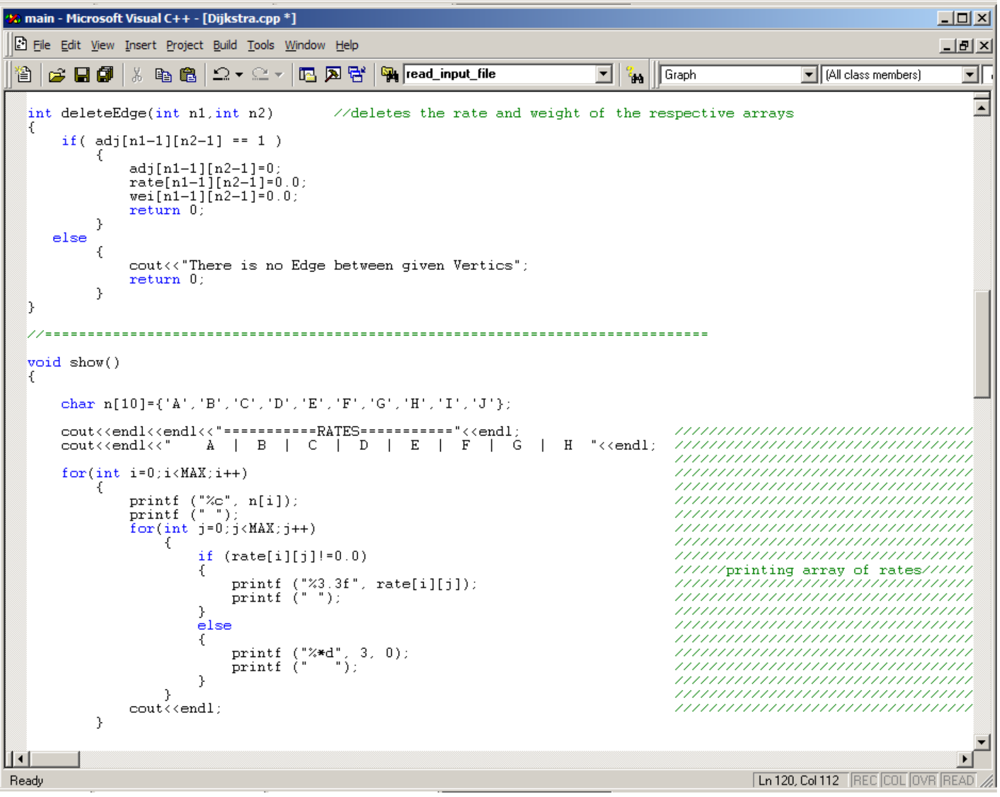
/**********************Dijkstra.cpp*****************************/



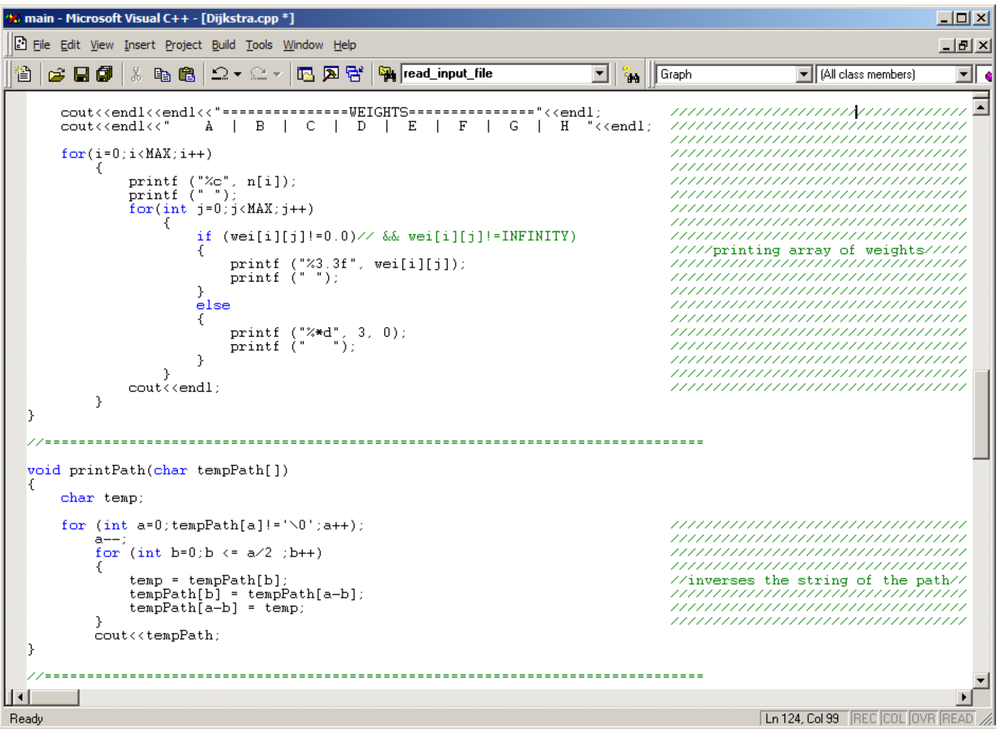
```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
#include<stdio.h>
#include<stdlib.h>
#include <string.h>

#define MAX 8               //number of nodes
#define INFINITY 32786
#define MEMBER 1
#define NONMEMBER 0


class Graph
{
private:
int adj[MAX][MAX];
float wei[MAX][MAX];        //weights = inverse rates
float rate[MAX][MAX];       //rates
float distance[MAX];        //total cost of path
int previous[MAX];          //previous node
int perm[MAX];
float weight;

public:
//=============================================================================

Graph()
{
for(int i=0;i<MAX;i++)                                  //initialization of Graph
    {
      distance[i]=0.0;
      perm[i]=0;
      previous[i]=0;
            for(int j=0;j<MAX;j++)
                {
                    adj[i][j]=0;
                    rate[i][j]=0.0;
                    wei[i][j]=0.0;
                }
    }
}

//=============================================================================
```



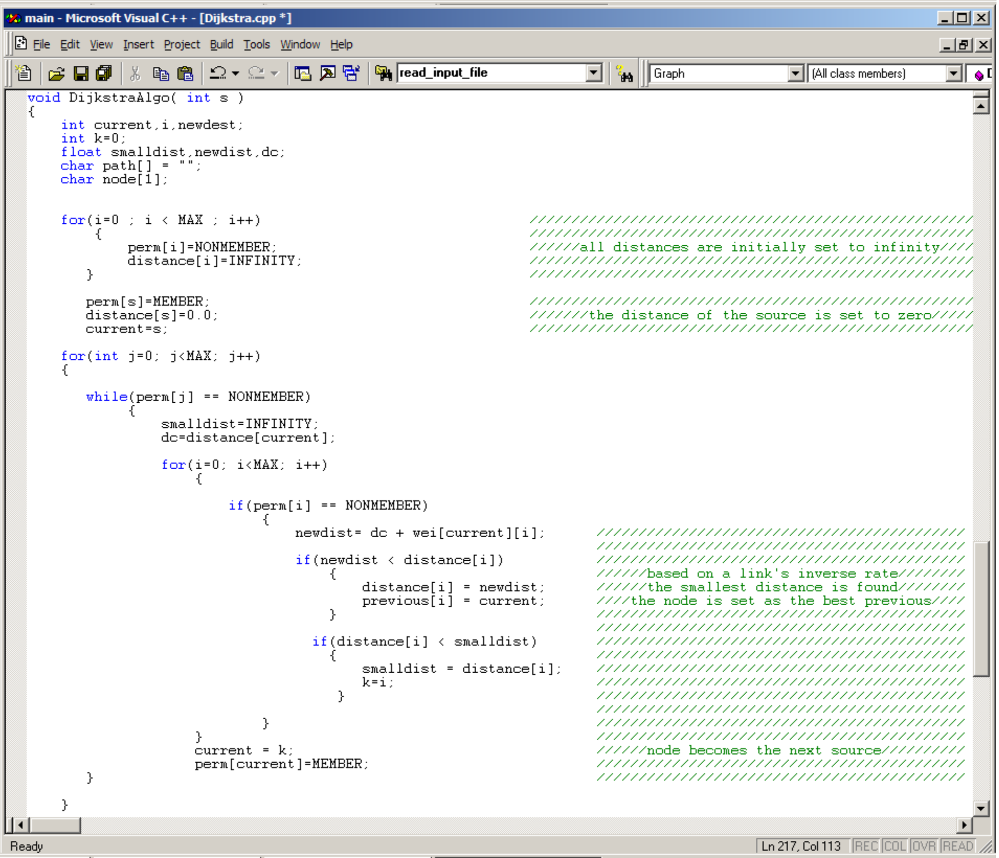
```
void InsertAllEdgesAtonce()
{
    for(int i=0;i<MAX;i++)
        {
        cout<<endl<<endl;
            for(int j=0;j<MAX;j++)
                {
                    cout<<"Enter rates of links "<<(i+1)<<" to "<<(j+1)<<" :";
                    cin>>rate[i][j];    //input value is the rate of the link of node i to j

                    if(rate[i][j] != 0.0)
                    {
                        adj[i][j] = 1;
                        wei[i][j] = (float) 1.0/rate[i][j];     //weight is the invers of the link's rate
                     }

                }
        }
}

//==============================================================================

void InsertRandomEdge(int n1,int n2, float r)       //inserts the rate and weight of a random edge
{
    adj[n1-1][n2-1]=1;
    rate[n1-1][n2-1]=r;
    wei[n1-1][n2-1] = (float) 1.0/r;
}

//==============================================================================
```



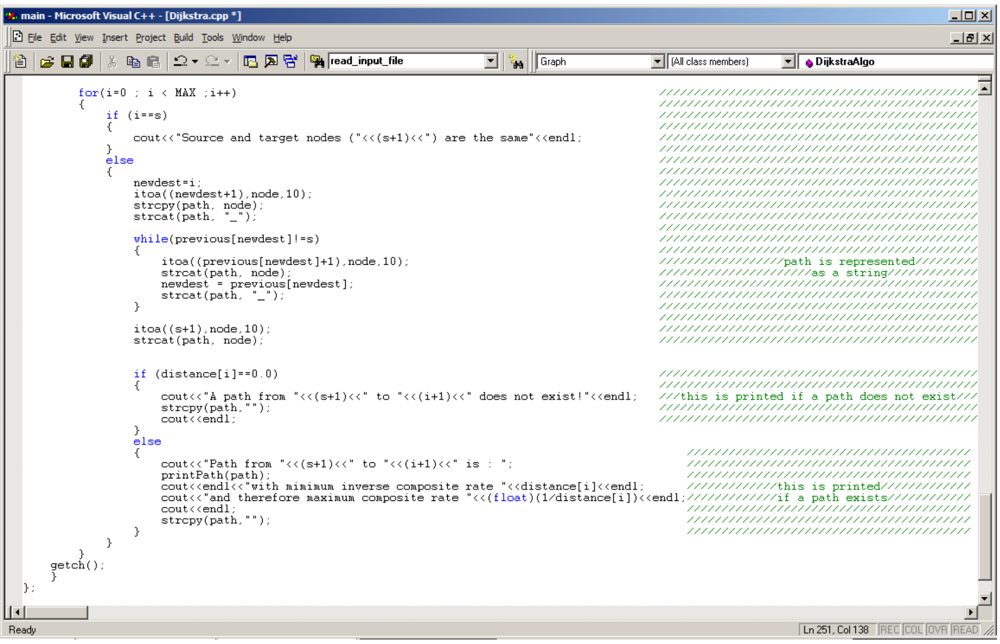
```
int deleteEdge(int n1,int n2)       //deletes the rate and weight of the respective arrays
{
    if( adj[n1-1][n2-1] == 1 )
        {
            adj[n1-1][n2-1]=0;
            rate[n1-1][n2-1]=0.0;
            wei[n1-1][n2-1]=0.0;
            return 0;
        }
   else
        {
            cout<<"There is no Edge between given Vertics";
            return 0;
        }
}

//============================================================================

void show()
{

    char n[10]={'A','B','C','D','E','F','G','H','I','J'};

    cout<<endl<<endl<<"==========RATES=========="<<endl;          ///////////////////////////////////
    cout<<endl<<"    A  |  B  |  C  |  D  |  E  |  F  |  G  |  H  "<<endl; ///////////////////////////////////
                                                                            ///////////////////////////////////
    for(int i=0;i<MAX;i++)                                                  ///////////////////////////////////
        {                                                                   ///////////////////////////////////
            printf ("%c", n[i]);                                            ///////////////////////////////////
            printf (" ");                                                   ///////////////////////////////////
            for(int j=0;j<MAX;j++)                                          ///////////////////////////////////
                {                                                           ///////////////////////////////////
                    if (rate[i][j]!=0.0)                                    ///////////////////////////////////
                    {                                                       //////printing array of rates/////
                        printf ("%3.3f", rate[i][j]);                       ///////////////////////////////////
                        printf (" ");                                       ///////////////////////////////////
                    }                                                       ///////////////////////////////////
                    else                                                    ///////////////////////////////////
                    {                                                       ///////////////////////////////////
                        printf ("%*d", 3, 0);                               ///////////////////////////////////
                        printf ("   ");                                     ///////////////////////////////////
                    }                                                       ///////////////////////////////////
                }                                                           ///////////////////////////////////
            cout<<endl;                                                     ///////////////////////////////////
        }
```



```
    cout<<endl<<endl<<"==============WEIGHTS=============="<<endl;         ///////////////////////////////////
    cout<<endl<<"    A  |  B  |  C  |  D  |  E  |  F  |  G  |  H  "<<endl;  ///////////////////////////////////
                                                                            ///////////////////////////////////
    for(i=0;i<MAX;i++)                                                      ///////////////////////////////////
        {                                                                   ///////////////////////////////////
            printf ("%c", n[i]);                                            ///////////////////////////////////
            printf (" ");                                                   ///////////////////////////////////
            for(int j=0;j<MAX;j++)                                          ///////////////////////////////////
                {                                                           ///////////////////////////////////
                    if (wei[i][j]!=0.0)// && wei[i][j]!=INFINITY)           ///////////////////////////////////
                    {                                                       /////printing array of weights/////
                        printf ("%3.3f", wei[i][j]);                        ///////////////////////////////////
                        printf (" ");                                       ///////////////////////////////////
                    }                                                       ///////////////////////////////////
                    else                                                    ///////////////////////////////////
                    {                                                       ///////////////////////////////////
                        printf ("%*d", 3, 0);                               ///////////////////////////////////
                        printf ("    ");                                    ///////////////////////////////////
                    }                                                       ///////////////////////////////////
                }                                                           ///////////////////////////////////
            cout<<endl;                                                     ///////////////////////////////////
        }
}

//================================================================================

void printPath(char tempPath[])
{
    char temp;

    for (int a=0;tempPath[a]!='\0';a++);                                    ///////////////////////////////////
        a--;                                                                ///////////////////////////////////
        for (int b=0;b <= a/2 ;b++)                                         ///////////////////////////////////
        {                                                                   ///////////////////////////////////
            temp = tempPath[b];                                             //inverses the string of the path//
            tempPath[b] = tempPath[a-b];                                    ///////////////////////////////////
            tempPath[a-b] = temp;                                           ///////////////////////////////////
        }                                                                   ///////////////////////////////////
        cout<<tempPath;
}

//================================================================================
```



```
void DijkstraAlgo( int s )
{
    int current,i,newdest;
    int k=0;
    float smalldist,newdist,dc;
    char path[] = "";
    char node[1];


    for(i=0 ; i < MAX ; i++)                                ////////////////////////////////////////////////////
        {                                                   ////////////////////////////////////////////////////
            perm[i]=NONMEMBER;                              //////all distances are initially set to infinity////
            distance[i]=INFINITY;                           ////////////////////////////////////////////////////
       }                                                    ////////////////////////////////////////////////////

       perm[s]=MEMBER;                                      ////////////////////////////////////////////////////
       distance[s]=0.0;                                     ///////the distance of the source is set to zero////
       current=s;                                           ////////////////////////////////////////////////////

    for(int j=0; j<MAX; j++)
    {

       while(perm[j] == NONMEMBER)
            {
                smalldist=INFINITY;
                dc=distance[current];

                for(i=0; i<MAX; i++)
                    {

                        if(perm[i] == NONMEMBER)
                            {
                                newdist= dc + wei[current][i];      ////////////////////////////////////////////
                                                                    ////////////////////////////////////////////
                                if(newdist < distance[i])           ////////////////////////////////////////////
                                    {                               //////based on a link's inverse rate////////
                                        distance[i] = newdist;      //////the smallest distance is found////////
                                        previous[i] = current;      ////the node is set as the best previous///
                                    }                               ////////////////////////////////////////////
                                                                    ////////////////////////////////////////////
                                  if(distance[i] < smalldist)       ////////////////////////////////////////////
                                    {                               ////////////////////////////////////////////
                                        smalldist = distance[i];    ////////////////////////////////////////////
                                        k=i;                        ////////////////////////////////////////////
                                     }                              ////////////////////////////////////////////
                                                                    ////////////////////////////////////////////
                            }                                       ////////////////////////////////////////////
                    }                                               ////////////////////////////////////////////
                    current = k;                                    //////node becomes the next source//////////
                    perm[current]=MEMBER;                           ////////////////////////////////////////////
       }                                                            ////////////////////////////////////////////

    }
```



```
        for(i=0 ; i < MAX ;i++)
        {
            if (i==s)
            {
                cout<<"Source and target nodes ("<<(s+1)<<") are the same"<<endl;
            }
            else
            {
                newdest=i;
                itoa((newdest+1),node,10);
                strcpy(path, node);
                strcat(path, "_");

                while(previous[newdest]!=s)
                {
                    itoa((previous[newdest]+1),node,10);        ///////////////////path is represented/////////
                    strcat(path, node);                         ///////////////////////as a string/////////////
                    newdest = previous[newdest];
                    strcat(path, "_");
                }

                itoa((s+1),node,10);
                strcat(path, node);


                if (distance[i]==0.0)
                {
                    cout<<"A path from "<<(s+1)<<" to "<<(i+1)<<" does not exist!"<<endl;   ///this is printed if a path does not exist///
                    strcpy(path,"");
                    cout<<endl;
                }
                else
                {
                    cout<<"Path from "<<(s+1)<<" to "<<(i+1)<<" is : ";
                    printPath(path);
                    cout<<endl<<"with mimimum inverse composite rate "<<distance[i]<<endl;    /////////////this is printed/////////////
                    cout<<"and therefore maximum composite rate "<<(float)(1/distance[i])<<endl;/////////////if a path exists/////////////
                    cout<<endl;
                    strcpy(path,"");
                }
            }
        }
    getch();
    }
};
```