

2009

# A Joint Radio and Channel Assignment (JRCA) scheme for 802.11-based wireless mesh networks

Ronnie Koshy  
*Iowa State University*

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Computer Sciences Commons](#)

---

## Recommended Citation

Koshy, Ronnie, "A Joint Radio and Channel Assignment (JRCA) scheme for 802.11-based wireless mesh networks" (2009). *Graduate Theses and Dissertations*. 10665.  
<https://lib.dr.iastate.edu/etd/10665>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

**A Joint Radio and Channel Assignment (JRCA) scheme for  
802.11-based wireless mesh networks**

by

Ronnie Koshy

A thesis submitted to the graduate faculty  
in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:  
Lu Ruan, Major Professor  
Wallapak Tavanapong  
Simanta Mitra

Iowa State University

Ames, Iowa

2009

Copyright © Ronnie Koshy, 2009. All rights reserved.

## DEDICATION

I would like to dedicate this thesis to my dad, mom and my brother, without whose support and encouragement, I would not have been able to complete this work. I would also like to thank my friends for their constant guidance during the writing of this work.

## TABLE OF CONTENTS

|   |      |
|---|------|
| <b>LIST OF TABLES</b> . . . . .                             | vi   |
| <b>LIST OF FIGURES</b> . . . . .                            | vii  |
| <b>ACKNOWLEDGEMENTS</b> . . . . .                           | viii |
| <b>ABSTRACT</b> . . . . .                                   | ix   |
| <b>CHAPTER 1. INTRODUCTION</b> . . . . .                    | 1    |
| 1.1 Overview of the Problem . . . . .                       | 1    |
| 1.2 Contribution of this work . . . . .                     | 2    |
| 1.3 Outline of this work . . . . .                          | 3    |
| <b>CHAPTER 2. IEEE 802.11 WIRELESS PROTOCOL</b> . . . . .   | 4    |
| 2.1 Introduction . . . . .                                  | 4    |
| 2.2 IEEE 802.11 Protocol Architecture . . . . .             | 4    |
| 2.2.1 802.11b (DSSS) . . . . .                              | 5    |
| 2.2.2 802.11g (ERP) . . . . .                               | 6    |
| 2.2.3 802.11a (OFDM) . . . . .                              | 6    |
| 2.2.4 802.11n (MIMO) . . . . .                              | 6    |
| 2.3 802.11 DCF and PCF . . . . .                            | 7    |
| 2.3.1 Challenges for the 802.11 MAC . . . . .               | 7    |
| 2.3.2 The Distributed Coordination Function (DCF) . . . . . | 10   |
| 2.3.3 The Point Coordination Function (PCF) . . . . .       | 11   |
| <b>CHAPTER 3. LITERATURE REVIEW</b> . . . . .               | 12   |

|   |   |           |
|---|---|-----------|
| 3.1   | Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks (HYACINTH)[11] | 12        |
| 3.1.1   | Main Ideas  | 13        |
| 3.1.2   | Strengths   | 15        |
| 3.1.3   | Weaknesses  | 15        |
| 3.2   | Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks (BFS-CA) [12]                     | 16        |
| 3.2.1   | Main Ideas  | 16        |
| 3.2.2   | Strengths   | 17        |
| 3.2.3   | Weaknesses  | 18        |
| 3.3   | A Channel Assignment Algorithm for Multi-Radio Wireless Mesh Networks (MCCA)[13].                             | 19        |
| 3.3.1   | Main Ideas  | 19        |
| 3.3.2   | Strengths   | 21        |
| 3.3.3   | Weaknesses  | 21        |
| 3.4   | Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks [14].                            | 22        |
| 3.4.1   | Main Ideas  | 22        |
| 3.4.2   | Strengths   | 24        |
| 3.4.3   | Weaknesses  | 24        |
| <b>CHAPTER 4. JOINT RADIO AND CHANNEL ASSIGNMENT SCHEME</b> |   |           |
|   | <b>(JRCA)</b>   | <b>25</b> |
| 4.1   | Introduction  | 25        |
| 4.2   | Basic Idea  | 26        |
| 4.3   | Algorithm Description   | 27        |
| 4.4   | Simulation Scenarios and Results  | 32        |
| 4.4.1   | Study 1 : JRCA vs Fixed-Radio Scenarios   | 32        |
| 4.4.2   | Study 2 : Trend of Throughput vs Client Traffic   | 36        |

|   |           |
|---|-----------|
| 4.4.3 Study 3 : Trend of Throughput Efficiency vs Network Density . . . . . | 38        |
| <b>CHAPTER 5. CONCLUSIONS AND FUTURE WORK . . . . .</b>                     | <b>51</b> |
| <b>BIBLIOGRAPHY . . . . .</b>   | <b>53</b> |

## LIST OF TABLES

|            |   |    |
|------------|---|----|
| Table 4.1  | Study 1 - Client Sources . . . . .                            | 34 |
| Table 4.2  | JRCA - Dense WMN - Number of interfaces . . . . .             | 34 |
| Table 4.3  | Study 1 - JRCA vs Fixed-radio scenario results . . . . .      | 34 |
| Table 4.4  | Dense WMN - Client sources (Run 0.5X) . . . . .               | 36 |
| Table 4.5  | Dense WMN - Client sources (Run 0.75X) . . . . .              | 36 |
| Table 4.6  | Dense WMN - Client sources (Run X) . . . . .                  | 37 |
| Table 4.7  | Dense WMN - Client sources (Run 1.25X) . . . . .              | 37 |
| Table 4.8  | Dense WMN - Client sources (Run 1.5X) . . . . .               | 37 |
| Table 4.9  | Study 2 - IGW Statistics (Client Traffic X=30 Mbps) . . . . . | 38 |
| Table 4.10 | Sparse WMN - Client sources (Run 0.5X) . . . . .              | 39 |
| Table 4.11 | Sparse WMN - Client sources (Run 0.75X) . . . . .             | 39 |
| Table 4.12 | Sparse WMN - Client sources (Run X) . . . . .                 | 40 |
| Table 4.13 | Sparse WMN - Client sources (Run 1.25X) . . . . .             | 40 |
| Table 4.14 | Sparse WMN - Client sources (Run 1.5X) . . . . .              | 40 |
| Table 4.15 | Study 3 - Throughput Efficiency vs Network Density . . . . .  | 50 |

## LIST OF FIGURES

|             |   |    |
|-------------|---|----|
| Figure 2.1  | The IEEE 802.11 Protocol Architecture . . . . .   | 5  |
| Figure 2.2  | Positive Acknowledgement . . . . .  | 7  |
| Figure 2.3  | The Hidden Node Problem. Although A and C are hidden from each other, their signals can collide at B. . . . . | 9  |
| Figure 2.4  | RTS-CTS Exchange . . . . .  | 9  |
| Figure 4.1  | Dense Network. . . . .  | 33 |
| Figure 4.2  | Dense WMN - Result of JRCA . . . . .  | 33 |
| Figure 4.3  | Variation of throughput : JRCA vs Fixed-radio scenarios . . . . .   | 35 |
| Figure 4.4  | Variation of delay : JRCA vs Fixed-radio scenarios . . . . .  | 36 |
| Figure 4.5  | A Sparse network . . . . .  | 39 |
| Figure 4.6  | Sparse WMN - JRCA result for traffic 0.5 X . . . . .  | 41 |
| Figure 4.7  | Sparse WMN - JRCA result for traffic 0.75 X . . . . .   | 42 |
| Figure 4.8  | Sparse WMN - JRCA result for traffic X . . . . .  | 43 |
| Figure 4.9  | Sparse WMN - JRCA result for traffic 1.25 X . . . . .   | 44 |
| Figure 4.10 | Sparse WMN - JRCA result for traffic 1.50 X . . . . .   | 45 |
| Figure 4.11 | Dense WMN - JRCA result for traffic 0.5 X . . . . .   | 46 |
| Figure 4.12 | Dense WMN - JRCA result for traffic 0.75 X . . . . .  | 47 |
| Figure 4.13 | Dense WMN - JRCA result for traffic 1.25 X . . . . .  | 48 |
| Figure 4.14 | Dense WMN - JRCA result for traffic 1.50 X . . . . .  | 49 |



## ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis.

First and foremost, I would like to thank my major professor Dr. Lu Ruan for her constant guidance throughout my research and the writing of this thesis. Her insights and words of encouragement have played a quintessential role in the successful completion of this work.

I would also like to thank my committee members, Dr. Wallapak Tavanapong and Dr. Simanta Mitra for all their efforts and contributions to this work. I would additionally like to thank the Department of Computer Science at Iowa State University for providing me with an ideal atmosphere to carry out my research activities.

## ABSTRACT

Wireless mesh network (WMN) is a promising technology for broadband Internet access due to its low cost, ease of deployment, increased coverage, and robustness. Equipping each mesh router with multiple radios can greatly improve the capacity of WMNs since the radios at each node can simultaneously transmit data on orthogonal channels.

Channel assignment is an important problem in multi-radio multi-channel WMNs where the goal is to assign a channel to each radio so that the resulting network topology is connected and the network capacity is maximized. Various channel assignment schemes have been proposed in literature. These schemes assume the number of radios at each node is given, but how this number is determined is not specified.

In this work, we propose a joint radio and channel assignment (JRCA) algorithm that determines the number of radios required at each node based on the traffic demand and produces the channel assignment for each radio to minimize the interference among the links operating on the same channel. Thus our algorithm can produce a network configuration that meets the customer demand and maximize the network capacity.

We conduct simulations to compare JRCA with a scheme where each node is equipped with the same number of radios and channel assignment is done by the MCCA algorithm. Our results show that, compared with the other schemes, JRCA can achieve higher throughput with fewer radios at the network nodes.

## CHAPTER 1. INTRODUCTION

### 1.1 Overview of the Problem

Wireless technology is currently being used to connect homes, neighborhoods, offices and schools to the ever-expanding and ubiquitous entity widely known as the Internet. IEEE 802.11-based wireless networks are usually used to provide *last mile* network service in remote areas. In order to facilitate this service, a wireless mesh network (WMN) is usually employed to enable voice and data communications, both locally and on the Internet. Mesh networking allows for continuous connections and reconfiguration around broken or blocked links by *hopping* from node to node until the destination is reached. A typical wireless mesh network consists of Internet Gateways(IGWs), Mesh routers and Mesh clients. Mesh clients consist of laptops or desktops equipped 802.11-based radios which in turn connect to Mesh routers. Mesh routers are responsible for forwarding client traffic to and from the Internet gateway(s).

An issue facing such community-based networks is the availability of substantial bandwidth required to support consumer's demands. Common wireless mesh implementations typically use commodity hardware equipped with a fixed number of radios that are sometimes inadequate to support client traffic demands. By simple observation, a workaround would be equip each mesh node with certain number of radios depending upon the net demand at the mesh nodes. However, this approach only solves a small part of the bigger problem. In addition to allocating radios to each of the nodes, one must also provide a suitable channel assignment of channels to all the radio interfaces so as to ensure network connectivity between the mesh clients and the Internet. Due to the well-known problem of channel interference, unwise channel assignments can negatively offset the advantage of increased bandwidth attained through the use of multiple radio interfaces. Henceforth, one needs to come up with a clever scheme to determine the right

number of radio interfaces and their respective channel assignments for each mesh node, in order to ensure that the total bandwidth available does indeed meet the client's demands. In light of the aforementioned facts, this work aims at developing a novel scheme to allocate both radios and their respective channels with a goal of satisfying consumer's traffic needs.

## 1.2 Contribution of this work

In this work, we propose a *Joint Radio and Channel Assignment* (JRCA) scheme, which is a novel technique of intelligently allocating radios and their respective channel assignments to wireless nodes in a mesh network with the intention of satisfying customer traffic demands. Previous work in the area of channel assignment in 802.11-based wireless networks work consider the number of interfaces available on wireless nodes as an input to their channel assignment algorithms. Given this assumption, their algorithms produce a suitable channel assignment for the entire network using several channel interference models like Multi-conflict graphs (MCGs)[12], Unit Disc graphs(UDG)[7][8], to name a few. However, their work does not deal with the goal of satiating customer's traffic needs, which we believe is core to success of any community-based wireless mesh network. Our work, to our best knowledge, has no precedence and aims to address the issue of interface allocation and channel assignment jointly.

The proposed scheme (JRCA) has the following features:

1. The issue of interface allocation and channel assignment is addressed jointly. Using a local search heuristic, the scheme creates interfaces at nodes by considering links in the decreasing order of their demands. When a new link is created, the heuristic provides a suitable channel assignment for the newly created interfaces at either ends of the link.
2. High throughput efficiency is achieved through the efficient utilization of network interfaces and the available channel spectrum.
3. The scheme performs very well in contrast to wireless mesh networks consisting of mesh routers equipped with a fixed number of interfaces.

### 1.3 Outline of this work

The rest of the thesis is structured as follows. We start with an overview of the IEEE 802.11[1] protocol, its architecture and its features in chapter 2. In chapter 3, we review relevant literature in the area of channel assignment in wireless networks. In chapter 4, we describe our *Joint Radio and Channel Assignment* (JRCA) scheme and provide comparative simulation results by contrasting our scheme with standard wireless mesh configurations equipped with fixed number of interfaces using metrics like throughput, packet delivery ratio, average end-to-end delay, etc. We also examine how the throughput varies when client traffic is varied and study the trend of throughput efficiency vs network density for sparse and dense WMNs setup using JRCA. Finally, we end the thesis by providing conclusions from our work and outlining future work in chapter 5.

## CHAPTER 2. IEEE 802.11 WIRELESS PROTOCOL

### 2.1 Introduction

A wireless mesh network (WMN) is a network of wireless communicating devices organized in a mesh topology[2]. WMNs are reliable, cost-effective and easy to deploy. These properties make them a promising technology for broadband Internet access. Typical WMN deployments are built using IEEE 802.11 a/b/g technology, with each mesh node equipped with multiple radio interfaces. In this chapter, we briefly review the IEEE 802.11 protocol, its architecture and issues that concern its functioning. The IEEE 802.11 protocol was conceived by the IEEE LAN/MAN Standards Committee (IEEE 802) [3] as a set of standards for wireless networks. The original version of the IEEE 802.11 was released in 1997 and clarified in 1999, but has since then been deprecated. However it was soon replaced by 802.11b followed by 802.11a and 802.11g.

### 2.2 IEEE 802.11 Protocol Architecture

The IEEE 802.11 protocol architecture is shown in Figure 2.1[5]

The lowest layer is the physical layer that contains the operating frequency bands, the modulation techniques used and the data rates available for use. Above the PHY lies the Medium Access Control layer which is the heart of the 802.11 operation. It works on a access control technique called *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA). This layer is used to avoid collisions by putting off transmission if the medium is busy, thereby ensuring that transmitting nodes don't interfere with each other. The CSMA/CA layer is composed to two sublayers, the DCF and PCF and are described in section 2.3. The Logical Link Control layer is right above the MAC layer and acts as an interface to the higher layers

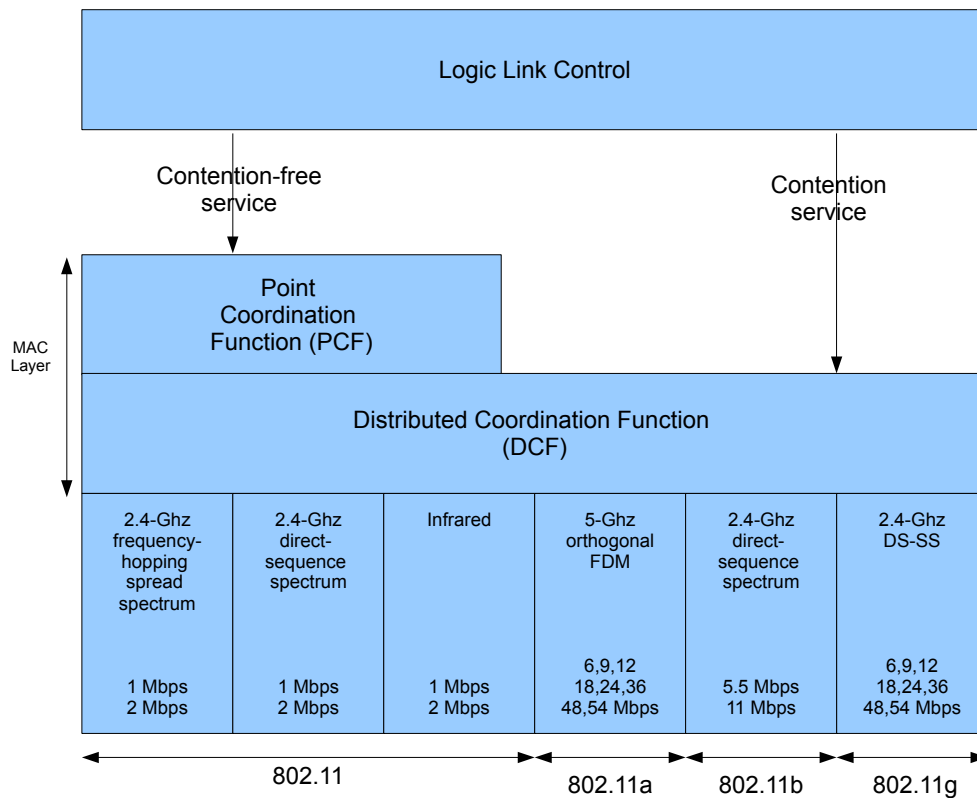


Figure 2.1 The IEEE 802.11 Protocol Architecture

by performing basic link level functions such as error control.

IEEE 802.11 can be classified into several flavors depending upon the modulation techniques used and the physical layer criterion of the operating frequency band. They are:

### 2.2.1 802.11b (DSSS)

802.11b is the most popular and widespread of all the IEEE 802.11 standards. It operates on the unlicensed 2.4 GHz frequency band. The initial revision of the 802.11 specification had a second physical layer based on direct sequence spread spectrum (DSSS). DSSS PHY supported data rates of 1Mbps and 2Mbps initially. On subsequent revision, a PHY with data rates of 5.5 Mbps and 11Mbps was added to 802.11b. However, the observed throughput is much lower since 802.11b faces intense interference from microwave ovens, cordless phone and other such

equipment operating on the same frequency. 802.11b supports only three orthogonal channels.

### **2.2.2 802.11g (ERP)**

802.11b only supported theoretical rates of upto 11 Mbps. However as demand grew, IEEE802.11 saw the need for higher data rates. As a consequence 802.11g was conceived. 802.11g is not a revolutionary specification. It still operates on the unlicensed 2.4 GHz frequency band. Most of 802.11g is occupied with providing backwards compatibility to 802.11b. 802.11g uses an Extended-Rate PHY (ERP). There are several flavors of ERP available. They include ERP-DSSS (1 and 2 Mbps), ERP-CCK (5.5 and 11 Mbps), ERP-OFDM (6,9,12,18,24,36,48,54 Mbps), ERP-PBCC (22 and 33 Mbps), DSSS-OFDM (hybrid scheme combining DSSS and OFDM).

### **2.2.3 802.11a (OFDM)**

The 2.4 GHz ISM bands are crowded and often compete with non-802.11 traffic. In an attempt to attain higher data rates, the working group standardized a PHY using the unlicensed spectrum of around 5GHz. 802.11a is based on orthogonal frequency division multiplexing (OFDM). OFDM is not a new modulation technique. 802.11a theoretically supports data rates of upto 54 Mbps as compared to 802.11b. However 802.11a radios are bit more expensive than 802.11b due to an advanced HAL (Hardware Access Layer) used. 802.11a supports twelve orthogonal channels.

### **2.2.4 802.11n (MIMO)**

802.11n is the latest standard and can potentially deliver upto 600Mbps, which is 50 times 802.11b and 10 times 802.11a or 802.11g. It uses MIMO (Multiple Input Multiple Output) which comprises of multiple antennas at both the transmitter and receiver, to improve communication performance.



## 2.3 802.11 DCF and PCF

### 2.3.1 Challenges for the 802.11 MAC

#### 2.3.1.1 RF Link Quality

On a wired Ethernet, it is expected that a packet sent from the sender to receiver is received correctly without data corruption. However, radio links are an entirely different ball game. They usually operate in hostile territory. They are frequently subject to noise and interference. In addition to noise, multipath fading may also lead to situations in which frames cannot be transmitted because a node moves into a dead spot. In contrast to other link layer protocols, 802.11 MAC uses *positive acknowledgements* (Figure 2.2 [4]) to ensure successful transmission. The receiver explicitly informs the sender when a frame is received correctly by sending an ACK frame back to the sender. If the sender does not receive an ACK within an established period of time, it has to retransmit the frame, upon timeout.

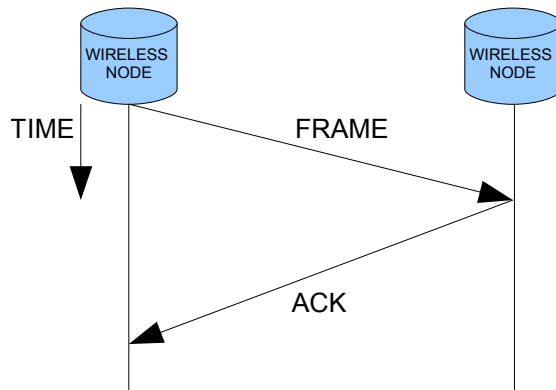


Figure 2.2 Positive Acknowledgement

### 2.3.1.2 The Hidden Node Problem

Consider the situation depicted in figure 2.3, where A and C are both within the range of B but not each other. Suppose A and C want to communicate with B, so both of them first sense the medium. A and C are unaware of each other since their signals do not carry that far. As a result, both of them sense that the medium is idle and both of them send B a frame at the same time. These two frames collide with each other at B, but unlike an Ethernet, neither A nor C is aware of this collision. A and C are said to be *hidden nodes* with respect to each other and this situation is known as the *hidden node* problem [9].

In Ethernet networks, stations depend on the reception of transmission to perform the collision detection functions of CSMA/CD. However, wireless transmissions are an entirely different story. Collision resulting from hidden nodes are hard to detect because wireless transceivers are inherently half-duplex; To prevent collisions, 802.11 uses a mechanism called RTS-CTS[4] (Figure 2.4) to clear the area before a radio starts transmitting. When a wireless node 1 has a frame to send; it initiates the process by sending out a RTS frame. In addition to reserving the radio link for transmission, RTS silences any stations that hear it. If the destination node receives the RTS, its responds with a CTS. Like RTS, CTS silences stations in the immediate surrounding. Once RTS/CTS exchange completes, node 1 can send its frames without worrying from interference from any hidden nodes. When the RTS/CTS mechanism is used, any frame transmitted must be positively acknowledged.

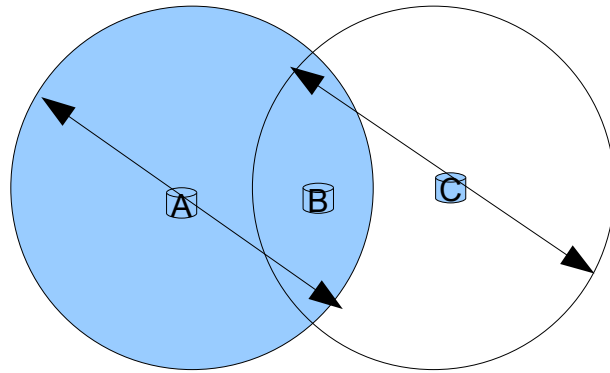


Figure 2.3 The Hidden Node Problem. Although A and C are hidden from each other, their signals can collide at B.

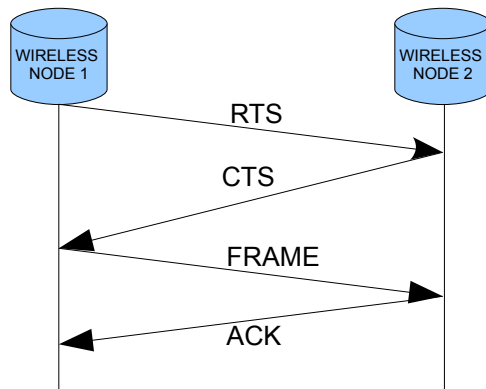


Figure 2.4 RTS-CTS Exchange

### 2.3.2 The Distributed Coordination Function (DCF)

WMNs use DCF, which provides a standard Ethernet-like contention-based service. DCF allows multiple mesh nodes to interact without central control. Before a node attempts to transmit, it checks if the medium is idle. If the medium is idle, the node must wait for a period called DIFS (DCF *interframe space*) before transmission can begin. This is done to avoid collisions. DCF may sometimes employ RTS/CTS[4] (Figure 2.4) frames to reduce the probability of collisions.

If the medium is busy, the node must wait for the channel to become free by deferring access. When access is deferred, the node waits for the medium to be idle for the DIFS and prepares for the random backoff procedure. In the random backoff phase, a period called *contention window* (CW) follows the DIFS. The CW is partitioned into time slots of equal length. When a node enters the random backoff phase, it picks a random time slot between 0 and CW. All time slots are equally likely. When several nodes prepare to transmit, the node that picked the smallest time slot wins and grabs the medium. The node now transmits the frame to the receiver. The other nodes remain silent during the frame transmission. To keep track of frame retransmissions, each frame has a single retry counter associated with it.

Since 802.11 uses positive acknowledgement (Figure 2.2 [4]), the transmitting node has to wait for the frame's acknowledgment. When a positive acknowledgement is received, the node infers that the frame has been successfully received by the receiver. Before transmitting the remaining frames, if any, the node resets the contention window and the retry counter to their minimum value.

However, if the node fails to receive the acknowledgement due to some error after a set time period, after timeout, it increments the retry counter and has to retransmit the frame. Each time the retry counter increases, the CW is doubled. When a frame is retransmitted, the node again picks a random time slot between 0 and CW and counts down to it before it can retransmit the frame. If retransmissions fail repeatedly, such that the retry counter reaches its maximum limit, the node gives up, resets the retry counter and the contention window. The frame is discarded and its loss is reported to the higher-layer protocols.

### 2.3.3 The Point Coordination Function (PCF)

The Point Coordination function provides contention free services. *Point Coordinators* are used to ensure that the medium is without contention. They reside in APs, thus restricting PCF to infrastructure networks. To obtain priority over standard contention-based services, PCF allows stations to send frames after a shorter interval than DIFS.

## CHAPTER 3. LITERATURE REVIEW

In this chapter, we review some of the relevant literature in the area of wireless mesh networks. Wireless Mesh Networks (WMNs) have become a key area of scientific research due to their diverse capabilities and lucrative advantages. Some seminal areas of research (surveyed in [17]), relating to WMNs, include routing, load balancing and channel assignment, to name a few. Routing in WMNs are of several flavors : Multiple-Radio Routing [18], Multi-Path Routing, Hierarchical Routing [19] and Geographical routing [20]. Load balancing is explored in [21], [22], [23], [24] and [25]. Several techniques for channel assignment have been proposed in [11], [12], [13], [14],[26], [27], [28] and [29].

In particular, we review some of broad schemes proposed by the research community to solve the channel assignment problem in Multi-radio Wireless Mesh Networks. We shed light on the basic ideas, advantages and drawbacks of the schemes presented in [11], [12], [13] and [14]. At the time of writing this thesis, we have come across no prior work that proposes a scheme that allocates radio and assigns channels jointly, a premise that we solve.

### 3.1 Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks (HYACINTH)[11]

In this paper, the authors are mainly concerned with the channel assignment problem in Multi-Radio Wireless Mesh Networks. They propose and evaluate one of the first multi-channel multi-hop wireless ad-hoc network architectures that can be built using standard 802.11 hardware by equipping each node with multiple network interface cards. Their main contributions are as follows:

- The first work of its kind that deals with channel assignment and routing in multi-radio

wireless mesh networks.

- An iterative algorithm that switches between channel assignment and the routing problem that terminates when convergence is observed.
- A novel link-wise channel assignment scheme that assign channels to network interfaces on either side of the link based on the expected link load.

### 3.1.1 Main Ideas

Mesh networks are limited in the amount of bandwidth they offer in contrast to Ethernet standards. In order to increase the bandwidth of traditional mesh networks, the authors argue that by equipping mesh router with multiple radios, this current bandwidth limit can be pushed higher. In particular, they mainly deal with two research questions:

1. How to assign channels to interfaces?
2. How to route packets by utilizing the multiple interfaces?

Greater the channel diversity, lesser will be the channel interference. In addition to channel diversity, network connectivity requires that two network interfaces at the two ends of the link should be tuned to the same channel frequency. However the routing methodology is dependent on the capacities of the links tuned to the same channel while the channel assignment is dependent upon the link loads, which in turn, is dependent on the routing methodology. This introduces a cyclic dependency between routing and channel assignment. Thus, in order to solve the problem, channel assignment and routing must be considered together.

A new evaluation metric called *cross-section goodput* is proposed. According to the authors, cross-section goodput is defined as  $X = \sum_{s,d} C(s, d)$  where  $C(s, d)$  is the *useful* network bandwidth assigned between a pair of ingress-egress nodes  $(s, d)$ . The goal of their channel assignment and routing algorithms is to *maximize* the cross-section goodput  $X$ .

The iterative algorithm begins with the traffic profile as input to the *Initial Link Load Estimation* step. This step is responsible for predicting a rough estimate of the expected link loads. In particular, the expected link load is estimated by using the equation  $\phi_l =$

$\sum_{s,d} \frac{P_l(s,d)}{P(s,d)} * B(s,d)$  where  $P(s,d)$  is the number of acceptable paths between  $(s,d)$  and  $P_l(s,d)$  is the number of acceptable paths between  $(s,d)$  that pass a link  $l$ .  $B(s,d)$  is the expected load between  $s$  and  $d$ , found in the traffic profile. The output of this step is fed into the next step in the iterative algorithm, *Channel Assignment*. In the channel assignment phase, links are traversed in decreasing order of load so that links with higher load gets channels with less contention. Several cases are considered when channels have to be assigned to two end network interfaces of a link.

In case 1, when both ends have unused interfaces, the algorithm assigns a channel with least degree of interference. In case 2, if the second node has only an unused interface, a channel from the first node having the least degree of interference is assigned to the second node's interface. In case 3 when neither node has unused interface, use a common channel if there exists one. In the final case, when neither node has unused interface and there exists no common channel, merge channels from one node to the other. The choice of the channel selected for merging is such that the combined degree of interference of the two channels is *minimized*. However, this requires merging network-wide. By *degree of interference*, the authors mean the sum of expected load from the virtual links in the interference region that are assigned to the same radio channel.

Once the channel assignment is done, link capacity estimation is done using the formula  $b_{w_i} = \frac{\phi_i}{\sum_{j \in Intf(i)} \phi_j} * C$ , where  $\phi_i$  is the expected load on link  $i$ ,  $Intf(i)$  is the set of all links in the interference zone of link  $i$  and  $C$  is the sustained radio channel capacity. For the routing component of the algorithm, they use two existing routing algorithms - shortest path routing and randomized multipath routing.

The above steps starting from the *Initial Link load estimation* step to the *routing* step keeps looping till convergence is observed or for all links, capacity  $>$  load. Convergence is attained when no more improvement can be observed with every consecutive run of the iterative algorithm.



### 3.1.2 Strengths

- The first work of its kind that deals with channel assignment and routing in multi-radio wireless mesh networks.
- A novel link-wise channel assignment scheme that assign channels to network interfaces on either side of the link based on the expected link load.

### 3.1.3 Weaknesses

- In the channel assignment algorithm, the merging phase has to be done recursively network-wide. This process is particularly expensive if the network is very large.
- There exists no proof of convergence for the iterative algorithm. It is difficult to bound the running time of their algorithm.
- Their channel assignment approach assigns channels to links. If the node degree is  $x$ , then it can only assign channels to a maximum of  $x$  radios even if more radios are available i.e no parallel links are allowed. Thus, the network may not be able to handle the customers traffic needs.

### 3.2 Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks (BFS-CA) [12]

In this paper, the authors explore the interference to wireless deployments due to co-located wireless networks and propose an algorithm (BFS-CA) by using nodes with multiple radios and assigning them channels using multi-radio conflict graphs. Their contributions are as follows:

- A dynamic, interference-aware channel assignment algorithm that minimizes interference between the mesh network and co-located wireless networks.
- A Multi-radio Conflict Graph (MCG), an extension to the well known conflict graph model, to model the interference relationship between multi-radio routers in a wireless mesh network.
- A novel interference estimation scheme that routers use to estimate the interference level in their neighborhoods.
- A link redirection protocol that prevents the disruption of flows during channel assignment.

#### 3.2.1 Main Ideas

Dynamic channel assignment can result in the changes in topology leading to suboptimal routing and even network partitioning. Another problem is network flows can get disrupted when a channel assignment is changed. Hence authors propose the use of a default radio at each node.

The channel assignment server (CAS) chooses the default channel using the rank of a channel  $c$  for the entire mesh  $R_c$ .  $R_c$  is computed as follows:  $R_c = \frac{\sum_{i=1}^n Rank_c^i}{n}$  where  $n$  is the number of routers and  $Rank_c^i$  is the rank of channel  $c$  at router  $i$ . The default channel is then chosen as the channel with the least  $R_c$  value. The intuition behind this metric is to use the least interfered channel as the default channel in the mesh.

Then the non-default channel selection phase constructs the MCG using the information collected from routers - neighbor identities, delays, etc - via *heartbeat* messages. An extended

model, called the Multi-radio Conflict Graph(MCG) is used to capture the interference relationship between nodes equipped with multiple radios. To create the MCG, each radio in the mesh network  $G$  is represented as a vertex in graph  $G'$  instead of representing routers by vertices as in  $G$ . The edges in  $G'$  are between mesh radios instead of mesh routers as in  $G$ . Given  $G'$ , each edge in  $G'$  is represented as a vertex in the conflict graph. An edge between two vertices in the conflict graph exists if the communication links corresponding to these two vertices in  $G'$  interfere with each other. After constructing the MCG, the CAS uses the BFS-CA algorithm to select channels for the non-default radios. Once the channels are selected for the mesh radios, the CAS instructs the routers to configure their radios to the newly selected channels.

The BFS-CA does a BFS of the MCG and starts assigning channels in increasing order of link hop counts. Radios are then permanently assigned channels that have the highest rank such that it does not conflict (no two adjacent vertices in the MCG are assigned the same color) with the neighbors. If a non-conflicting channel is not available, a randomly chosen channel is assigned to the vertex with the exception of the default channel. Once a vertex is assigned a channel, all vertices that contain either radio from the just-assigned vertex are placed in a list and are removed from the MCG. This step is needed to satisfy the constraint that only one channel is assigned to each radio. Once this completes, vertices at the next level of the breadth first search are visited. This procedure keeps on repeating until all the vertices in the MCG are visited and radios are assigned channels.

Once the channel assignments are decided, the CAS notifies the mesh routers to reassign their radios to the chosen channels via the default radio. To adapt to changing interference, CAS periodically ( 10 minutes) re-assigns channels. When a new channel is being assigned to the non-default radios, existing traffic is routed over the default radio till the switch is done.

### 3.2.2 Strengths

- This work proposes an algorithm that takes care of interference from co-located wireless deployments.

- A novel way of capturing interference relationship for multi-radio wireless mesh nodes using Multi-Radio Conflict Graphs (MCGs).
- Avoids sub-optimal routes and network partitioning by using default channel.
- A link redirection scheme when non-default radios switch channels.

### **3.2.3 Weaknesses**

- Links having the same hop-count are treated the same irrespective of load.
- A dedicated channel required for default channel.
- Assigns a random channel if a non-conflicting channel does not exist.
- Throughput drop during link-redirection.

### 3.3 A Channel Assignment Algorithm for Multi-Radio Wireless Mesh Networks (MCCA)[13].

In this paper, the authors study channel interference in multi-radio wireless mesh networks and propose a centralized algorithm called Maxflow-based Channel Assignment (MCCA) to maximize the throughput. Their contributions are as follows:

- A centralized channel assignment algorithm (MCCA) for Multi-Radio Wireless Mesh Networks.
- The MCCA channel assignment scheme identifies critical links and assigns channels to them such that most critical links experience least possible interference.
- Their scheme solves the recursive channel assignment problem with [11].

#### 3.3.1 Main Ideas

In order to maximize the throughput, the MCCA algorithm first seeks to identify links that are most critical to carry traffic and protect those against interference. The rationale behind this approach is that interference is the major cause for throughput decrease and accordingly the MCCA algorithm assigns channels in such a way that the most critical links experience the least possible interference.

To calculate the critical links, the authors first compute the maximum throughput flowing from the aggregation devices to the gateways on the initial graph. To this end, two extra nodes are connected to the nodes in the graphs and are identified as the supersource and supersink. Using the maxflow algorithm [16], flows on each of the links are calculated, giving an indication of the criticality of the link.

The channel assignment algorithm consists of two stages - the *Link-group binding* stage and the *Group-Channel Assignment* stage. During the *Link-group binding* stage, links are grouped based on the flows they carry. It begins by visiting all the nodes of the network one-by-one. For each node, the set of groups for all the links incident to that node is computed. If the number of groups of links incident on a node is greater than the number of available radios,

such a violation is remedied by repeatedly merging two groups till the number of groups equals the number of radios. At each iteration, the groups chosen to be merged are the two with the least total flow. The merging reassigns all the links belonging to one group to the other. The next step is to assign a group to all the links of a node that are still unassociated with any group. The principle, at the basis of the grouping performed by this step, is to aggregate the links of the node into groups equal to the number of radios available at that node. An attempt is made to give the links with the largest flow higher precedence, in the sense that they are assigned the same group as the links with the smallest flow, ensuring that links carrying large flows are the most critical and should suffer less interference.

The second stage, called *Group-Channel Assignment*, selects a channel for each group and assigns the selected channel to all the links of the group. The *Group-Channel Assignment* algorithm performs this task with the objective of protecting critical links. This is achieved by sorting the groups in decreasing order of the maximum flow associated with any of the links within the group and visiting them one-by-one in such an order. The algorithm starts by considering the set of links *potentially interfering* with the links assigned to a group. Two links *potentially interfere* with each other if one of the end nodes of one link is in the interference range of one of the end nodes of the other link. For each channel, the set  $S(g, c)$  of all the links that are assigned channel  $c$  and *potentially interfere* with the links of  $g$  is computed. If there exists a channel  $c_0$  such that  $S(g, c_0)$  is empty, it means that none of the links potentially interfering with the links in  $g$ , has been assigned channel  $c_0$ . So  $c_0$  can be assigned to the group  $g$ . In case multiple such channels exist, the channel that has been assigned to the highest number of links is chosen.

After the second stage of the MCCA algorithm, the number of channels assigned to a node does not exceed the number of radios since the first stage returns a number of groups per node not greater than the number of radio interfaces. Thus the constraint on the number of radios per node is obeyed and no replacement of previous channel assignments is required.

### 3.3.2 Strengths

- The MCCA channel assignment scheme identifies critical links and assigns channels to them such that most critical links experience least possible interference, so that network throughput can be maximized.
- Their scheme solves the recursive channel assignment problem with [11];
- Simulations show that MCCA achieves higher throughput than IATC[15] scheme.

### 3.3.3 Weaknesses

- The MCCA channel assignment scheme is independent of any traffic profile.
- Sometimes, links adjacent to a node, are assigned the same channels.
- Their channel assignment approach assigns channels to links. If the node degree is  $x$ , then it can only assign channels to a maximum of  $x$  radios even if more radios are available. One way to fix this is to allow parallel links.

### 3.4 Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks [14].

In this paper, the authors consider multihop wireless mesh networks, where each router is equipped with multiple radio interfaces and multiple channels are available for communication. In particular, the paper addresses the problem of assigning channels to communication links in the network with the objective of minimizing the overall network interference. The paper proposes centralized and distributed algorithms for the channel assignment problem. Their contributions are as follows:

- The paper proposes a 'topology preserving' approach i.e all links that can exist in a single channel network also exist in the multichannel network after channel assignment.
- The channel assignment schemes considers traffic load.
- Their work generalizes to non-orthogonal channels, including channels that are supposedly orthogonal but interfere because of crosstalk or leakage[30].
- The work is the first work that establishes good lower bounds on the optimal network interference and demonstrates a good performance of the heuristics by comparing them with the lower bounds.

#### 3.4.1 Main Ideas

Tabu Search [31] is a technique used for coloring vertices in graphs. The authors present a centralized version of Tabu Search for solving the channel assignment problem. According to the authors, centralized algorithms are quite practical in "managed" mesh networks, where there is already a central entity. The authors use the Max-K-Cut problem with interface constraint to model the assignment problem.

The channel assignment problem is to color the vertices  $V_c$  of the conflict graph  $G_c$  using  $K$  colors while maintaining the interface constraint and minimizing the number of monochromatic edges in the conflict graph. *Interface constraint* means that the number of channels assigned to



the links incident on any node is at most the number of radios on that node. A conflict graph is a graph that captures the interference between two or more communicating links and consists of the vertices corresponding to the communication links within the mesh. Two vertices in the conflict graph are connected by a conflict edge iff the two communication links corresponding to the two vertices interfere with each other.

The centralized Tabu-based channel assignment algorithm consists of two phases. In the first phase, the tabu-technique is used to find a good solution  $f$  (a channel assignment) without worrying about the interface constraint. It starts with a random initial solution  $f_0$ , wherein each vertex is assigned to a random color in  $K$ . From this random initial solution  $f_0$ , a sequence of solutions  $f_0, f_1, f_2, \dots$  is created in an attempt to reach a solution with minimum network interference. A *Tabu list* of limited size is maintained to avoid reassigning the same color to a vertex more than once i.e when generating neighboring solutions, neighboring solutions that contain color assignment in the Tabu list are ignored.

In the second phase, their technique removes the interface constraint violations by repeated application of the "merge" procedure to get a feasible channel assignment function  $f$ . Given a channel/color assignment solution  $f$ , a node for the merge operation is picked as follows: Among all the network nodes where the interface constraint is violated, pick the node wherein the violation is maximum.

In addition to the centralized tabu search channel assignment technique, a distributed greedy heuristic is proposed that takes into consideration the interface constraint right from the start. The idea is to initially, color all the nodes in the conflict graph with the same color. Then in each iteration of the algorithm, the color of some vertex is changed in a greedy manner without violating the interface constraint. The iterative algorithm is guaranteed to terminate because each iteration decreases the network interference monotonically. Their approach can be distributed or localized since the interference is local.

Their work also derives lower bounds on the minimum network interference using semidefinite programming(SDP) [32] approaches, in order to evaluate the quality of their proposed solutions.

### 3.4.2 Strengths

- A 'topology preserving' approach allowing all links that can exist in a single channel network also to exist in the multichannel network after channel assignment.
- Both centralized Tabu-based search algorithm and Distributed Greedy algorithm are very simple to implement.
- Their channel assignment schemes consider traffic loads.
- It is the first work that establishes good lower bounds on the optimal network interference using SDP.

### 3.4.3 Weaknesses

- Tabu Search is a local search algorithm. The effectiveness of the Tabu search is heavily dependent upon the starting solution. If the starting solution is not good, then the end result will not be close to the optimal value.
- Channels are assigned on a link basis. So no parallel links are allowed between two nodes.

## CHAPTER 4. JOINT RADIO AND CHANNEL ASSIGNMENT SCHEME (JRCA)

### 4.1 Introduction

Our work deals with the issue of ensuring that customer's demands are satisfied in a 802.11-based wireless mesh network. In order to send a packet to the Internet, a mesh client (say a 802.11-enabled laptop) must forward the packet to a neighboring mesh router within the transmission range of the mesh client. If the mesh router is in direct transmission range of the IGW, it sends the received packet directly to the IGW; else it forwards it to another adjacent mesh router. This happens till the packet is successfully delivered to the IGW. Traffic from the Internet to the mesh client follows the same process as before with exception being that packets start from the IGW to the mesh client.

Depending on the size of the wireless mesh network, the number of mesh clients could be small or large. As a result, a network deployment engineer, undertaking the task of setting up such a network must carefully design a wireless mesh network taking into consideration the customer's needs. Given the placement of mesh routers and customer demands, a network designer is typically faced with the problem of deciding how many radios must each mesh router possess and what channels should they be tuned to so as to reduce channel interference while at the same time extract the maximum possible bandwidth from such a wireless installation. Our work aims to do just that by providing the network designer with the right number of radio interfaces that should be setup at each of the intermediate mesh routers and the IGWs and their respective channel assignments so that the network is connected while at the same time offer the maximum possible bandwidth to the end users.

We propose the *Joint Radio and Channel Assignment* (JRCA) scheme with the following

features:

1. The issue of interface allocation and channel assignment is addressed jointly. Using a local search heuristic, the scheme creates interfaces at nodes by considering links in the decreasing order of their demands. When a new link is created, the heuristic provides a suitable channel assignment for the newly created interfaces at either ends of the link.
2. High throughput efficiency is achieved through the efficient utilization of network interfaces and the available channel spectrum.
3. Our scheme performs very well in contrast to wireless mesh networks consisting of mesh routers equipped with a fixed number of interfaces.

## 4.2 Basic Idea

Given traffic demands between the mesh clients and the IGWs, the JRCA scheme first uses an existing channel assignment algorithm called Maxflow-based Centralized Channel Assignment (MCCA) [13] to obtain the initial channel assignment for the network. This is done so as to give us a beginning estimate of the load that will be carried by the assigned channels in the network. The MCCA algorithm was used over other available channel assignment techniques because of an important feature: The channel assignment scheme should be efficient and should not suffer from weaknesses as in the case of [11] (recursive channel assignment problem).

The next step is to update the links within the network so as to reflect this channel assignment i.e for every link in the network assigned a channel, we create two links - one with the channel assignment of the MCCA algorithm with a bandwidth allocation equal to the bandwidth supported by one radio interface and a second *residual* link with no channel assignment having a demand equal to the actual link demand minus the demand allocated already. Once this operation finishes, we then create a sorted list (in decreasing order of demands) of residual links having non-zero demands. We also create a list of *domains* i.e for every channel available for assignment, a *domain*, representative of the channel, is created. Technically speaking, a domain is basically a data structure that is identified by the channel it represents and contains

the list of links and their demands, assigned to that channel. The list of domains is then updated to reflect the initial channel assignment.

We then visit the list of *residual* links and read their demands one by one. When a *residual* demand is read, we create a new link between the two endpoints of the demand. When such a link is created, we use a local heuristic that employs the list of *domains*, to determine what channel we must assign to this link. Once a channel is assigned to this link, we update the corresponding domain with the allocated demand and update the *residual* link capacity. This process repeats until the residual demand of the original link under consideration becomes zero or until no channels are available for assignment. If no channels are available for assignment, the *remaining* residual demand is allocated in the final step of the JRCA scheme (It involves re-sorting the remaining residual links in decreasing order of demands and allocating them using the channel distribution in the vicinity of that residual link). Once all the links in the list are visited and allocated (no residual demands remain), we will have a comprehensive radio and channel assignment for the entire network, which is basically the output of the JRCA scheme.

### 4.3 Algorithm Description

The following is the pseudocode for the proposed *Joint Radio and Channel Assignment* (JRCA) scheme for 802.11-based wireless mesh networks:

- 1: Input :  $G = (V, E)$ ,  $K$ ,  $d(e)$ ,  $\beta$
- 2:  $\forall u \in V$ ,  $radio(u) = \frac{\sum_{e=(u,v) \in E} d(e)}{\beta}$
- 3:  $MCCA(G, K, d(e), radio(u))$ .
- 4: Create a graph  $G' = (V', E')$  consisting of two kind of links for every single link  $e = (u, v) \in E$  - one link  $l = (u', v')$  with the  $chan(l) = chan(e)$  for every corresponding link  $e \in E$  assigned a channel and allocation  $alloc(l) = \beta$  ( $d(l) = 0$ ); and a *residual* link  $l_{res} = (u', v')$  with demand  $d(l_{res}) = d(e) - \beta$  and  $chan(l_{res}) = -1$ .
- 5: Create a set  $D$  of *domains* representing the total number of channels available such that  $|D| = K$ .
- 6: For each  $e' \in E'$ ,  $load(chan(e')) = load(chan(e')) + d(e')$  such that  $chan(e') \neq -1$

7: Let  $L =$  set of *residual* links in  $G'$  sorted in decreasing order of demands such that  $d(l) > 0$   
and  $chan(l) = -1, \forall l \in E'$

8: **while** all links in  $L$  have not been visited **do**

9:   Let  $l = (u', v')$  be an unvisited link

10: **while**  $d(l) > 0$  **do**

11:    $C = \cup chan(l)$

12:    $C_{u'} = \cup chan(l') | \forall y, l' = (u', y) \in E'$

13:    $C_{v'} = \cup chan(l') | \forall y, l' = (v', y) \in E'$

14:    $K' = K - C - (C_u \cup C_v)$

15:   **if**  $|K'| > 0$  **then**

16:     Pick  $k \in K'$  with least  $load(k)$

17:     Create link  $l' = (u', v')$  with  $chan(l') = k$  and allocation  $alloc(l') = \beta$

18:      $d(l) = d(l) - \beta$

19:      $load(k) = load(k) + \beta$

20:   **else**

21:      $C' = (C_{u'} \cup C_{v'}) - C$

22:     **if**  $C' = \text{NULL}$  **then**

23:       Exit While

24:     **end if**

25:     Compute the number of conflicts for domains in  $C'$  when new link added to a  
domain.

26:     Pick  $k \in C'$  with least number of conflicts.

27:     Create link  $l' = (u', v')$  with  $chan(l') = k$  and allocation  $alloc(l') = \beta$

28:      $d(l) = d(l) - \beta$

29:      $load(k) = load(k) + \beta$

30:   **end if**

31: **end while**

32: **end while**

33: Assign remaining *residual* links (sorted in decreasing order of demands) in  $G'$  to domains with lowest conflicts.

34: Output  $G'$

Consider a 802.11-based wireless mesh network represented as graph  $G = (V, E)$  where  $V$  represents the nodes in the mesh network and  $E$  is the set of links between nodes. A link  $e = (u, v) \in E$  if nodes  $u$  and  $v$  are in the transmission range (TR) of each other i.e  $dist(u, v) \leq TR$ . For clarity of presentation, we assume that the Interference Range (IR) is twice the transmission range i.e  $IR = 2TR$ . Each edge is labeled by a demand  $d(e)$  which is basically the amount of traffic or bandwidth that should be supported by this link  $e$ . Let  $K$  be the total number of channels available for use.  $K$  should be equal to the number of orthogonal (non-overlapping) channels available.  $K$  is not limited to channels of one 802.11 technology, but can consist of the total channels available for the entire 802.11 architecture including 802.11a, b/g. The configuration of the WMN ( $G$ ), the corresponding demands  $d(e)$  in addition to the parameter  $K$ , serve as the input for JRCA scheme. Let  $\beta$  be the capacity (maximum bandwidth) of a radio interface.

The JRCA scheme starts off by initially calculating the possible number of radios at every mesh node (line 2). This step is required, since the MCCA [13] algorithm, in addition to the network configuration  $G$  and channels  $K$ , requires the number of radio interfaces at each of the mesh nodes as input. The graph  $G$  along with the parameters  $K$  and  $radio(u)$ , the number of radio interfaces at each mesh node  $u$ , is fed as input for the MCCA [13] algorithm which does the initial channel assignment (line 3). The motivation for this step was discussed earlier, in the *Basic Idea* section. Once the MCCA algorithm finishes executing, we read the graph  $G$  for the initial channel assignment. Let  $chan(e)$  represent the channel assigned to a link  $e \in E$ .

Given this channel assignment we do two things:

1. We create a new graph  $G'$  (line 4) which will serve as one of the inputs for the JRCA scheme.  $G'$  consists of two sorts of links - a *allocated* link and a *residual* link. A *allocated* link has the following characteristics : It is assigned a channel (via step 3) and has an allocation of  $\beta$  ( $d(l) = 0$ ) while a *residual* link has no channel assignment i.e

$chan(l_{res}) = -1$  and has a demand  $d(l_{res}) = d(e) - \beta$ .

2. We create the *domains*. The idea behind the *domains* concept was discussed in the *Basic Idea* section. In addition to its creation, domains are populated with links and their loads by associating links ( having a channel assignment) to domains. (lines 5-6).

The next step is to create a list  $L$  which basically consists of links in  $E'$  sorted in decreasing order of residual demands such that  $d(l) > 0$  and  $chan(l) = -1, \forall l = (u', v') \in E'$  (line 7). The JRCA scheme then visits all the links in  $L$  (lines 8-32) from the largest  $d_l$  to the smallest  $d_l$  such that  $l = (u', v') \in L$  so that links with higher demands can be assigned to channels with less contention. A while loop (lines 10-31) is run as long as the condition  $d(l) > 0$  is satisfied. During each run of the while-loop, we first create a set  $C$  (line 11) which basically consists of the channels assigned to links that exist between the nodes  $u'$  and  $v'$  and within the transmission range of either  $u'$  or  $v'$ . Sets  $C_{u'}$  and  $C_{v'}$  consists of the channels assigned to links emanating from  $u'$  and  $v'$  respectively (lines 12 and 13). A set  $K'$  is calculated by performing the operation :  $K' = K - C - (C_{u'} \cup C_{v'})$  (line 14).

If  $|K'| > 0$ , this means that there exists atleast one channel that has not be assigned to a link in the neighborhood around  $u'$  or  $v'$ . We pick a channel  $k \in K'$  such that  $load(k)$  is the least load among those in  $K'$ . Thereafter, we create a link  $l' = (u', v')$  with  $chan(l') = k$  and allocation  $alloc(l') = \beta$ .  $d(l)$  is updated to  $d(l) = d(l) - \beta$  and  $load(k) = load(k) + \beta$  (lines 16-19). If  $|K'| = 0$  (lines 21-30), then we create a set  $C' = (C_{u'} \cup C_{v'}) - C$ . If  $|C'| = 0$ , we give up and exit the while-loop; otherwise we calculate the number of conflicts for the channels  $c' \in C'$  when a new link is added to that corresponding channel  $c'$ . By the number of *conflicts*, we mean the number of potential links that will interfere with a new link  $l'$ , when that link is assigned to the same channel as the potentially interfering links (A newly created link  $l' = (u', v')$  *conflicts* with link  $l = (u, v)$  iff  $chan(l') = chan(l)$  and  $(u'$  or  $v')$  lies in the interference range of  $(u$  or  $v)$ ) (A node  $u$  lies in the *interference range* of node  $v$  iff the euclidean distance  $dist(u, v) \leq IR$ ) .

We then pick a channel  $c'$  from  $C'$  with the least number of conflicts and create a link  $l' = (u', v')$  with  $chan(l') = c'$  and an allocation  $alloc(l') = \beta$ .  $d(l)$  is updated to  $d(l) = d(l) - \beta$



and  $load(c') = load(c) + \beta$  (lines 26-29).

Once the while-loop exits, we then traverse the list  $L$  once again to check if there exists any links that have non-zero residual demands. If so, we sort  $L$  once again in the same way as before, but by discarding links with zero residual demands (line 33). We again proceed in the same way by assigning channels in such a fashion to each of these residual links with the least number of conflicts. The JRCA scheme is now complete and outputs the network  $G'$  containing the newly created links along with their corresponding channels. The number of radio interfaces at a mesh router can be inferred from the number of distinct channel links emanating from that mesh router. Formally, the number of radio interfaces at a node  $u$  is given by the following:

$$\{Radios(u) = \#\mathbf{channels\ assigned\ to\ links\ at\ node\ }u\}$$

If two or more links are on the same channel, we use only one interface at the two endpoints of the links.

*Complexity analysis:* We denote  $n$  as the number of nodes and  $m$  as the number of links within the WMN. Let  $K$  be the total number of non-overlapping channels available for channel assignment. The first phase of the JRCA which basically involves initial channel assignment using MCCA, runs in  $O(mn^2)$  time[13].

The most time-consuming operation of JRCA, in addition to MCCA, is the two while loops between lines 8-32 and 10-31. In the outer while loop, we basically visit the links in the decreasing order of their residual loads. This takes  $O(m)$  time as this list can have at most  $m$  residual links. The inner while loop runs as long the demand is greater than zero. This takes  $O(D_{max})$  time where  $D_{max}$  is the maximum residual demand in the list of residual demands. Within the inner while loop, the *if – else* statement takes the most time of  $O(mK)$ . The *if* component takes  $O(K)$  time since we are basically finding the domain with the smallest load, while the *else* component takes  $O(mK)$  time. This is because we have to compute the number of conflicts for the domains in  $C'$ . In the worst-case, if  $|C'| = K$  such that each domain has at most  $m$  links, then then running time is  $O(mK)$ . Hence the total running time of the two

while loops is  $O(m * D_{max} * mK) = O(m^2 D_{max} K)$ . Combining these results, we get an overall running time for JRCA as  $O(mn^2) + O(m^2 D_{max} K)$ .

#### 4.4 Simulation Scenarios and Results

We conduct simulations to evaluate the performance of our JRCA algorithm. Firstly, we study the trend of throughput for WMNs (setup using JRCA) vs fixed-radio WMNs. By fixed-radio WMNs, we refer to those WMNs where each mesh node within the WMN is equipped with the same number of radio interfaces. Secondly, the variation of *throughput vs client traffic* is studied by observing the throughput achieved at the IGW when client traffic is varied, in a WMN, setup using JRCA. Lastly we wanted to study the variation of *throughput efficiency vs network density* (sparse WMN vs dense WMN). By *network density*, we refer to the number of active links within the WMN. A WMN with a high number of active links is classified as a *dense* WMN while a WMN with a low number of active links is classified as a *sparse* WMN.

All the scenarios were setup and simulated using the Qualnet Simulator [10]. 802.11a was used, allowing us to work with 12 non-overlapping channels ( $K = 12$ ). The data rate used was  $6Mbps$  ( $\beta = 6Mbps$ ). Constant Bit Rate (CBR) traffic was used for simulating client traffic. The transmission range was set at  $250m$ . Routing was done manually. All simulations were run for  $300s$ .

##### 4.4.1 Study 1 : JRCA vs Fixed-Radio Scenarios

For the purpose of the first study, a dense network (figure 4.1) was used consisting of 10 nodes - one IGW (node 2) and 9 mesh routers. Four client sources (A, 9, 5, 8) were used to generate the traffic demands (shown in table 4.1). The labels on the edges in figure 4.1 represents the traffic demand (in Mbps) flowing across an edge. The result of the JRCA scheme for the dense WMN is shown in figure 4.2. The value inside the brackets represent the channel assignment for that corresponding link.

To design fixed-radio scenarios, we went through the number of interfaces used at each mesh node outputted for the dense WMN (table 4.2). Starting from the smallest number of

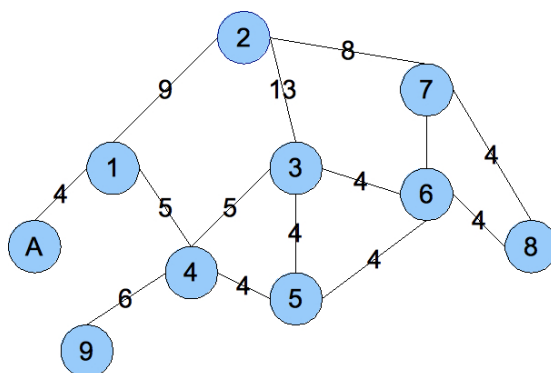


Figure 4.1 Dense Network.

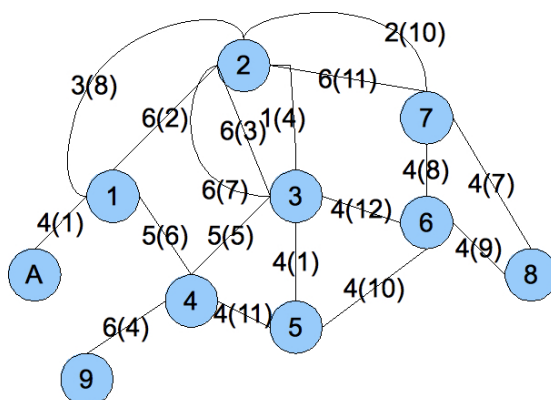


Figure 4.2 Dense WMN - Result of JRCA

radio interfaces (1 interface) to the largest number of radio interfaces (7 interfaces) used for the dense WMN, we equip each mesh node with the same number of radio interfaces. The channel assignment for each of the fixed-radio scenarios was done using the MCCA [13] algorithm. Traffic sources listed in table 4.1 was used in all the simulations for the fixed-radio scenarios.

Table 4.3 shows the simulation results that compares the performance of JRCA with fixed radio scenarios. The results are plotted in figures 4.3 and 4.4. Figure 4.3 shows the throughput observed at the IGW for JRCA (experiment 1) vs throughput for scenarios (experiments 2,3,4,5,6,7 and 8) containing nodes equipped with fixed number of radio interfaces. Figure 4.4 represents the average end-end delays experienced at the IGW of each of the eight experiments. End-end delay can occur due to a number of reasons like delay occurred at the network layer queue, MAC layer delay, transmission delay, propagation delay, etc.

Table 4.1 Study 1 - Client Sources

| Clients | Throughput | Total Bytes Sent | Total Packets Sent |
|---------|------------|------------------|--------------------|
| 5       | 12Mbps     | 450000000        | 450000             |
| 8       | 8Mbps      | 300000000        | 300000             |
| 9       | 6Mbps      | 225000000        | 225000             |
| A       | 4Mbps      | 150000000        | 150000             |

Table 4.2 JRCA - Dense WMN - Number of interfaces

| Node         | Coordinates | Interfaces |
|--------------|-------------|------------|
| 1            | (180,315)   | 4          |
| 2            | (260, 371)  | 7          |
| 3            | (278,291)   | 6          |
| 4            | (212,246)   | 4          |
| 5            | (289,218)   | 3          |
| 6            | (336,262)   | 4          |
| 7            | (340,326)   | 4          |
| 8            | (395,270)   | 2          |
| 9            | (214,163)   | 1          |
| A            | (55,276)    | 1          |
| <b>Total</b> |             | <b>36</b>  |

From the data presented, it can be observed that JRCA is much more efficient than fixed radio scenarios by guaranteeing higher throughput and lower average end-end delays. In addition to this, JRCA uses only 36 radios to achieve better performance results compared to fixed-radio scenarios of 40, 50, 60, 70 radios, assigned channels using MCCA[13].

The variation of throughput for experiments 2 through 8 can be explained as follows: as the number of radios increases, the throughput increases but then flattens out. The initial increase, from experiment 2 to 5, can be attributed to the increase in the number of interfaces which

Table 4.3 Study 1 - JRCA vs Fixed-radio scenario results

| Experiment      | Total Radios | Throughput | Packet Recvd | Bytes Recvd | Average Delay |
|-----------------|--------------|------------|--------------|-------------|---------------|
| 1 (JRCA)        | 36           | 21.01 Mbps | 787782       | 787782000   | 423.78ms      |
| 2 (1 int. each) | 10           | 3.90 Mbps  | 146138       | 146138000   | 1209.43ms     |
| 3 (2 int. each) | 20           | 9.44 Mbps  | 354060       | 354060000   | 825.12ms      |
| 4 (3 int. each) | 30           | 15.64 Mbps | 586393       | 586393000   | 694.23ms      |
| 5 (4 int. each) | 40           | 19.87 Mbps | 745088       | 745088000   | 579.22ms      |
| 6 (5 int. each) | 50           | 19.87 Mbps | 745092       | 745092000   | 580.43ms      |
| 7 (6 int. each) | 60           | 19.87 Mbps | 745090       | 745090000   | 579.17ms      |
| 8 (7 int. each) | 70           | 19.87 Mbps | 745087       | 745087000   | 580.61ms      |

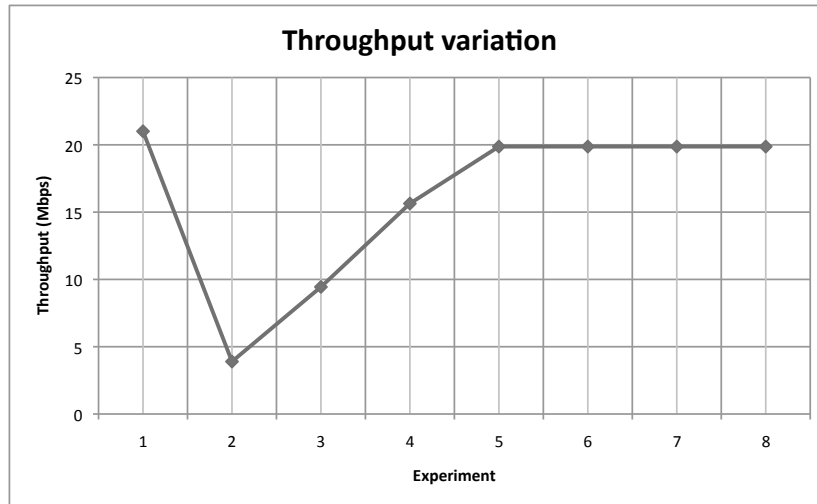


Figure 4.3 Variation of throughput : JRCA vs Fixed-radio scenarios

offers increased bandwidth. However, from experiment 5 onwards, the throughput remains the same. Although there is an increase in the number of interfaces, the degree of a mesh node (i.e the number of links emanating from that node) remains the same for every subsequent scenario. As a consequence, the MCCA algorithm is unable to use all the radio interfaces due to the limiting effect of the node degree. As a consequence, the throughput observed at the IGW remains the same.

The variation of average end-end delay for experiments 2 through 8 can be explained as follows: as the number of radios increases, the average end-end delay decreases but then flattens out. This trend, for experiments 2 through 5, is expected since multiple radios decrease the congestion level within the WMN due to increased bandwidth. The reduced congestion reduces the queuing delay and the number of packet retransmits, thus lowering the delay for packet transmission. The trend of the average end-end delay, from experiment 5 onwards, can be attributed to the same reason (node degree) used to explain the trend of throughput observed for experiments 5,6,7 and 8.

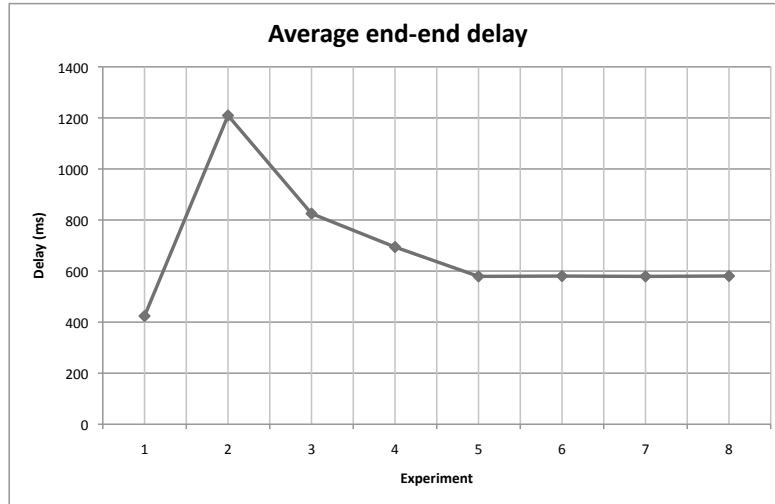


Figure 4.4 Variation of delay : JRCA vs Fixed-radio scenarios

#### 4.4.2 Study 2 : Trend of Throughput vs Client Traffic

In this study, we wanted to observe how the throughput varied when client traffic was varied, given the same WMN, setup using JRCA. To facilitate this, we used the JRCA result for the dense WMN (figure 4.2) and varied the client traffic (in table 4.1). Client sources at  $0.5X$  (table 4.4),  $0.75X$  (table 4.5),  $X$  (table 4.6),  $1.25X$  (table 4.7) and  $1.5X$  (table 4.8) were used to analyze the trend. Here  $X$  represents the total client traffic ( $X = 30Mbps$ ).

Table 4.4 Dense WMN - Client sources (Run 0.5X)

| Clients | Throughput | Total Bytes Sent | Total Packets Sent |
|---------|------------|------------------|--------------------|
| 5       | 6Mbps      | 225000000        | 225000             |
| 8       | 4Mbps      | 150000000        | 150000             |
| 9       | 3Mbps      | 112500000        | 112500             |
| A       | 2Mbps      | 75000000         | 75000              |

Table 4.5 Dense WMN - Client sources (Run 0.75X)

| Clients | Throughput | Total Bytes Sent | Total Packets Sent |
|---------|------------|------------------|--------------------|
| 5       | 9Mbps      | 337500000        | 337500             |
| 8       | 6Mbps      | 225000000        | 225000             |
| 9       | 4.5Mbps    | 168750000        | 168750             |
| A       | 3Mbps      | 112500000        | 112500             |

Table 4.6 Dense WMN - Client sources (Run X)

| <b>Clients</b> | <b>Throughput</b> | <b>Total Bytes Sent</b> | <b>Total Packets Sent</b> |
|----------------|-------------------|-------------------------|---------------------------|
| 5              | 12Mbps            | 450000000               | 450000                    |
| 8              | 8Mbps             | 300000000               | 300000                    |
| 9              | 6Mbps             | 225000000               | 225000                    |
| A              | 4Mbps             | 150000000               | 150000                    |

Table 4.7 Dense WMN - Client sources (Run 1.25X)

| <b>Clients</b> | <b>Throughput</b> | <b>Total Bytes Sent</b> | <b>Total Packets Sent</b> |
|----------------|-------------------|-------------------------|---------------------------|
| 5              | 15Mbps            | 562500000               | 562500                    |
| 8              | 10Mbps            | 375000000               | 375000                    |
| 9              | 7.5Mbps           | 281250000               | 281250                    |
| A              | 5Mbps             | 187500000               | 187500                    |

Table 4.8 Dense WMN - Client sources (Run 1.5X)

| <b>Clients</b> | <b>Throughput</b> | <b>Total Bytes Sent</b> | <b>Total Packets Sent</b> |
|----------------|-------------------|-------------------------|---------------------------|
| 5              | 18Mbps            | 675000000               | 675000                    |
| 8              | 12Mbps            | 450000000               | 450000                    |
| 9              | 9Mbps             | 337500000               | 337500                    |
| A              | 6Mbps             | 225000000               | 225000                    |

Table 4.9 shows the throughput, bytes received, packets received, average delay and throughput efficiency observed at the IGW. Throughput efficiency is calculated using the formula :

$$\text{Throughput Efficiency} = \frac{\text{IGW Throughput}}{\text{Client Traffic}}$$

From the data presented, it can be seen that as the client traffic increases, the throughput efficiency decreases. This is because the WMN was designed for client traffic X (30 Mbps)(table 4.1). When traffic higher than X is sent, it is expected that queues at each of the mesh nodes get saturated leading to packets being dropped at the mesh nodes. At every hop, this packet drop gets magnified, thus causing an overall decrease in the traffic observed at the IGW in comparison to the client traffic generated.

The average delay increases as the client traffic increases. The increased average delay can

Table 4.9 Study 2 - IGW Statistics (Client Traffic X=30 Mbps)

| Client Traffic | Throughput | Bytes Recvd | Packets Recvd | Average Delay | Efficiency |
|----------------|------------|-------------|---------------|---------------|------------|
| 0.5X           | 11.12 Mbps | 417097000   | 417097        | 393.43ms      | 74.15%     |
| 0.75X          | 16 Mbps    | 600011000   | 600011        | 410.15ms      | 71.11%     |
| X*             | 21.01 Mbps | 787782000   | 787782        | 423.78ms      | 70.03%     |
| 1.25X          | 23.27 Mbps | 872508000   | 872508        | 593.22ms      | 62.05%     |
| 1.50X          | 25.71 Mbps | 963985000   | 963985        | 697.17ms      | 57.13%     |

be attributed to the increase in packet drops at the mesh nodes. Since TCP guarantees reliable transmission, packets sent from the client must be *successfully delivered* to the receiver. By *successfully delivery*, we mean that the sending node must receive a positive acknowledgment from the receiver for the transmitted packet. However, due to an increase in client traffic, the congestion level increases, causing an increase in packet drops. So the sender is forced to resend packets till it receives a positive acknowledgments from the receiver. As a consequence, the average delay increases as shown in table 4.9.

Although designed for  $X$ , the dense WMN is only able to attain an efficiency of around 70%. Consider the results of JRCA for the dense WMN depicted in figure 4.2. There are a total of 18 active links within the WMN, which is greater than the number of non-overlapping channels (12 channels) used for the channel assignment. As a result, it is expected that more than one link is assigned to the same channel. For example, links (1,  $A$ ) and (3, 5) both use *channel1* while links (5, 6) and (2, 7) use *channel10*. When two links, in the interference range of each other, are assigned to the same channel, only one link can be active at a time, leading to a decrease in the network bandwidth. This explains why the efficiency of the dense WMN is 70%.

Based on the above observation, we expect that a sparse WMN will have a higher efficiency than a dense WMN. This observation is essentially the motivation for the final study, where we explore the performance of JRCA in dense and sparse WMNs.

#### 4.4.3 Study 3 : Trend of Throughput Efficiency vs Network Density

In this study, we compare the efficiency of JRCA in sparse and dense WMNs. To facilitate this, we use a sparse network (figure 4.5) consisting of only 7 mesh nodes - one IGW



(node 1) and 6 mesh routers and 8 active links. Three clients sources (3,5,7) are used to generate the traffic demands for this sparse network. The labels on the edges in figure 4.5 represents the traffic demands (in Mbps) flowing across an edge. Traffic sources listed in tables 4.10, 4.11, 4.12, 4.13 and 4.14 are used to setup sparse WMNs (using JRCA). For setting up dense WMNs (figure 4.1) (using JRCA), we use the traffic sources listed in tables 4.4, 4.5, 4.6, 4.7 and 4.8. Figures 4.6, 4.7, 4.8, 4.9 and 4.10 show the JRCA results for the sparse network for client traffic of 0.5X, 0.75X, X, 1.25X and 1.50X respectively (X=20 Mbps) while figures 4.11, 4.12, 4.2, 4.13 and 4.14 show the JRCA results for dense WMN for client traffic of 0.5X, 0.75X, X, 1.25X and 1.50X respectively (X=30 Mbps).

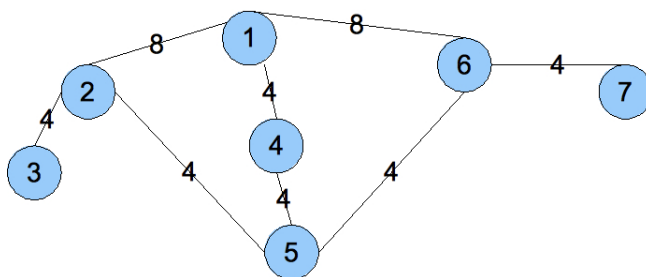


Figure 4.5 A Sparse network

Table 4.10 Sparse WMN - Client sources (Run 0.5X)

| Clients | Throughput | Total Bytes Sent | Total Packets Sent |
|---------|------------|------------------|--------------------|
| 3       | 2Mbps      | 75000000         | 75000              |
| 5       | 6Mbps      | 225000000        | 225000             |
| 7       | 2Mbps      | 75000000         | 75000              |

Table 4.11 Sparse WMN - Client sources (Run 0.75X)

| Clients | Throughput | Total Bytes Sent | Total Packets Sent |
|---------|------------|------------------|--------------------|
| 3       | 3Mbps      | 112500000        | 112500             |
| 5       | 9Mbps      | 337500000        | 337500             |
| 7       | 3Mbps      | 112500000        | 112500             |

Table 4.12 Sparse WMN - Client sources (Run X)

| <b>Clients</b> | <b>Throughput</b> | <b>Total Bytes Sent</b> | <b>Total Packets Sent</b> |
|----------------|-------------------|-------------------------|---------------------------|
| 3              | 4Mbps             | 150000000               | 150000                    |
| 5              | 12Mbps            | 450000000               | 450000                    |
| 7              | 4Mbps             | 150000000               | 150000                    |

Table 4.13 Sparse WMN - Client sources (Run 1.25X)

| <b>Clients</b> | <b>Throughput</b> | <b>Total Bytes Sent</b> | <b>Total Packets Sent</b> |
|----------------|-------------------|-------------------------|---------------------------|
| 3              | 5Mbps             | 187500000               | 187500                    |
| 5              | 15Mbps            | 562500000               | 562500                    |
| 7              | 5Mbps             | 187500000               | 187500                    |

Table 4.14 Sparse WMN - Client sources (Run 1.5X)

| <b>Clients</b> | <b>Throughput</b> | <b>Total Bytes Sent</b> | <b>Total Packets Sent</b> |
|----------------|-------------------|-------------------------|---------------------------|
| 3              | 6Mbps             | 225000000               | 225000                    |
| 5              | 18Mbps            | 675000000               | 675000                    |
| 7              | 6Mbps             | 225000000               | 225000                    |

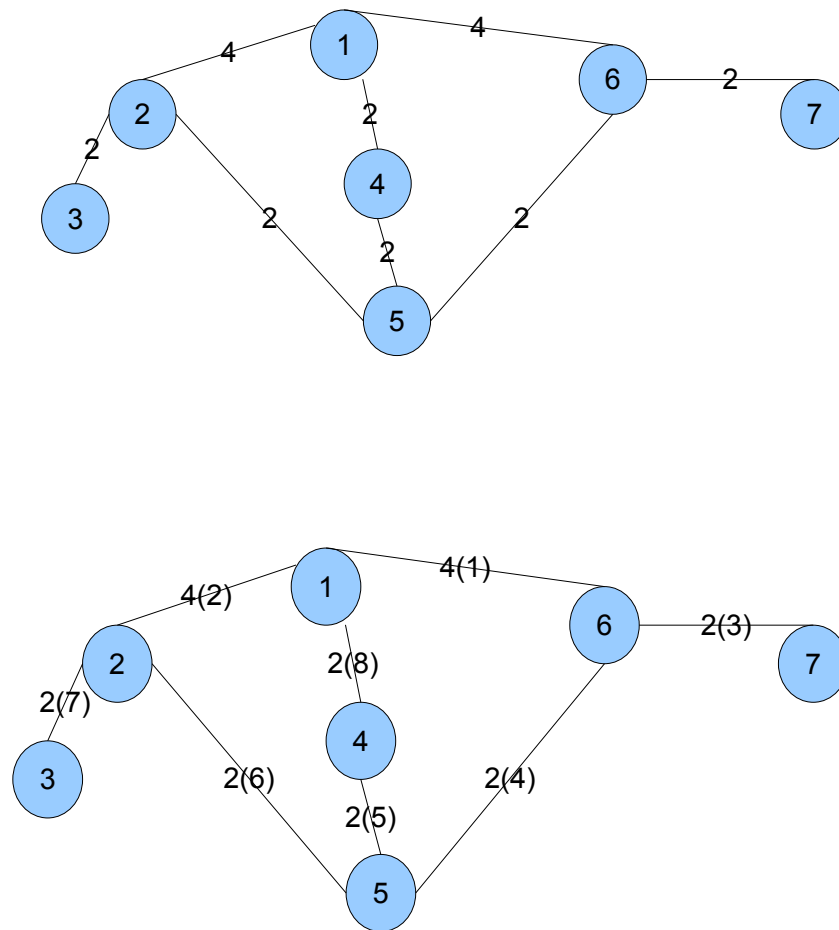


Figure 4.6 Sparse WMN - JRCAs result for traffic 0.5 X

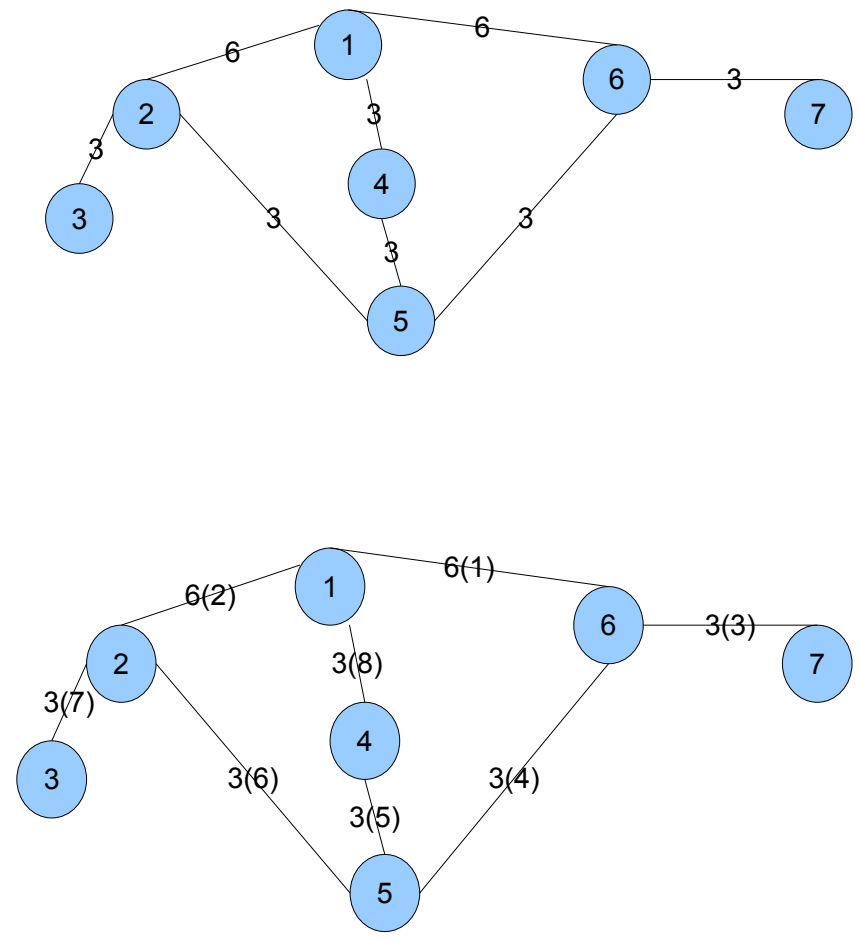


Figure 4.7 Sparse WMN - JRCA result for traffic 0.75 X

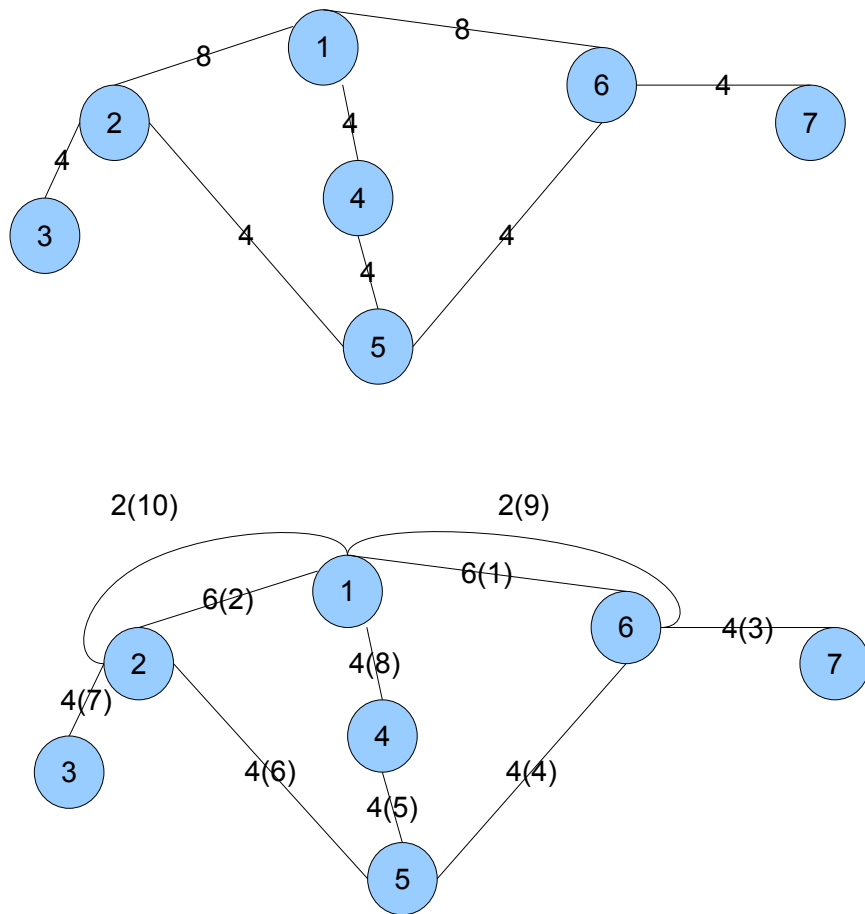


Figure 4.8 Sparse WMN - JRCA result for traffic X

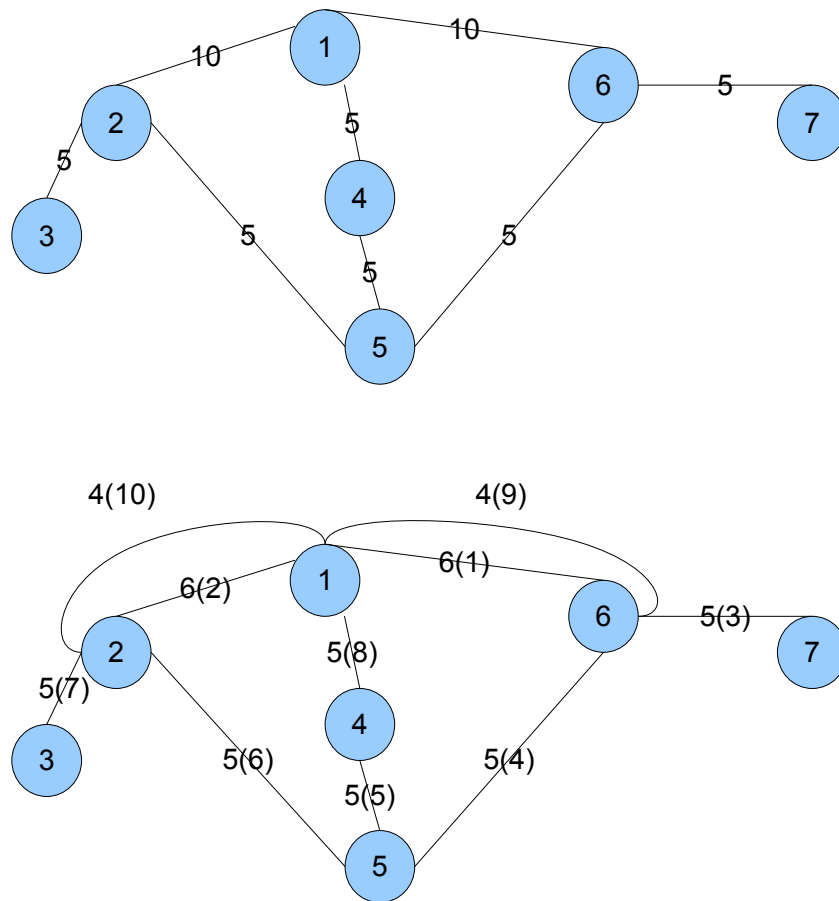


Figure 4.9 Sparse WMN - JRCA result for traffic 1.25 X

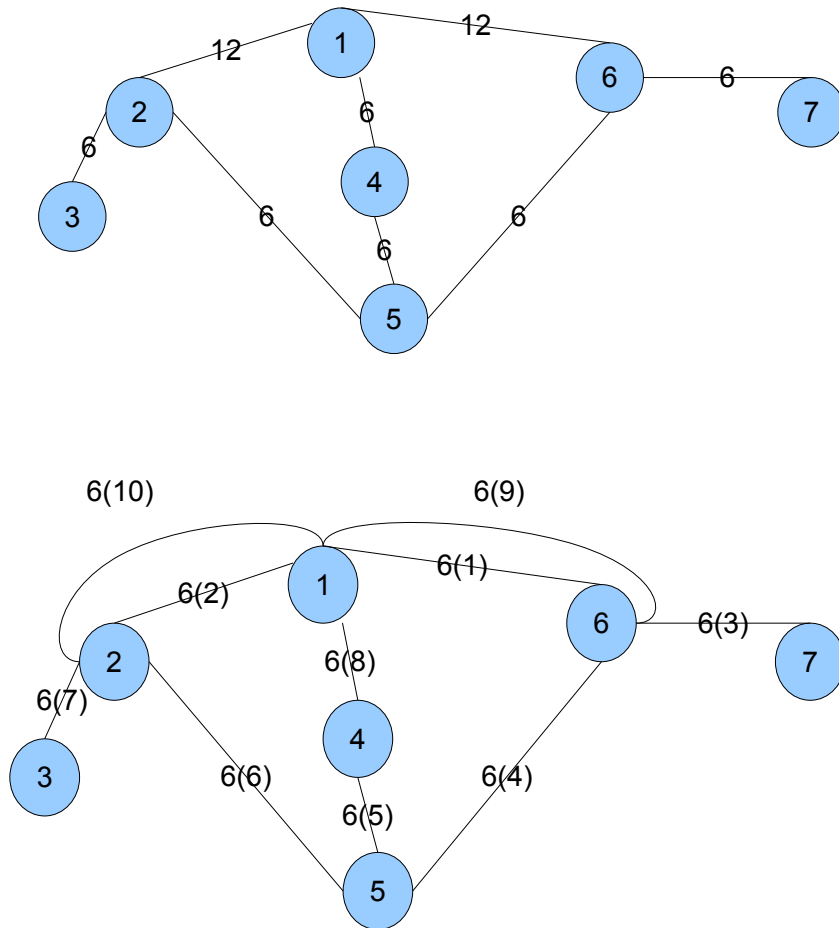


Figure 4.10 Sparse WMN - JRCAs result for traffic 1.50 X

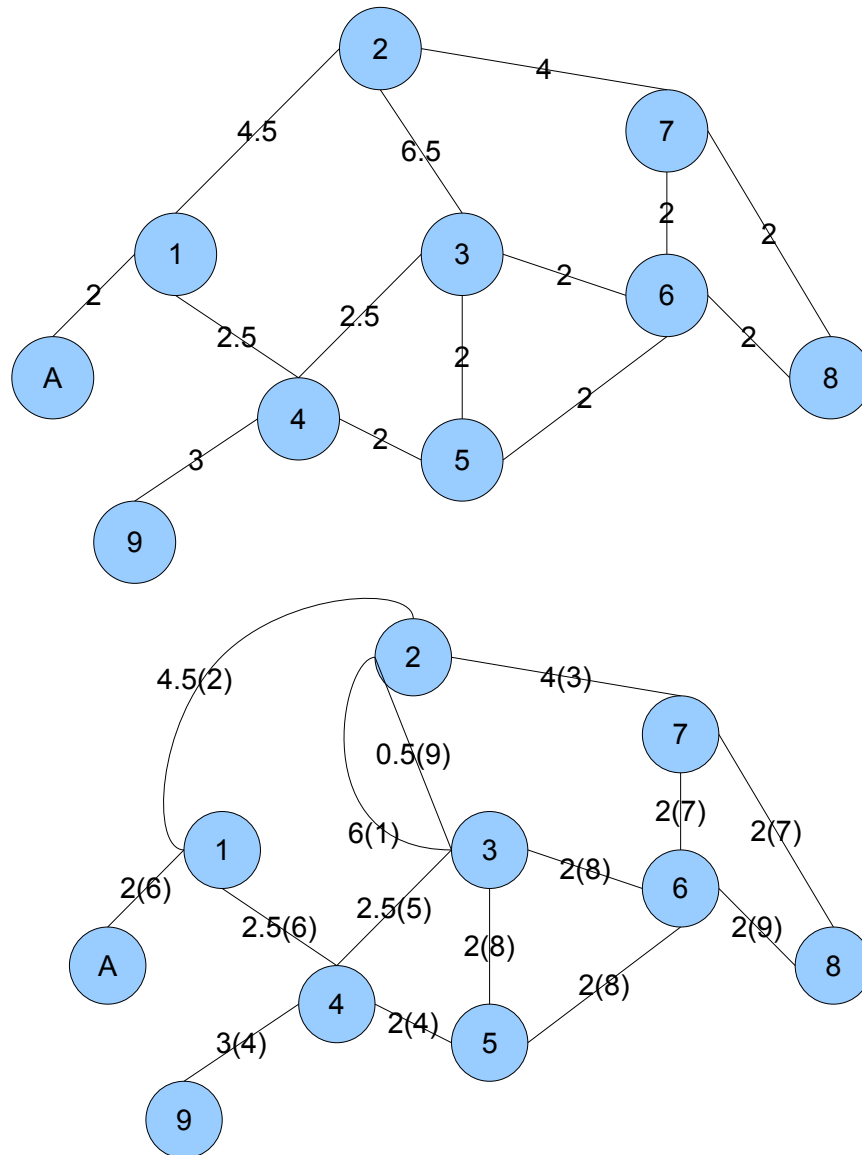


Figure 4.11 Dense WMN - JRCA result for traffic 0.5 X



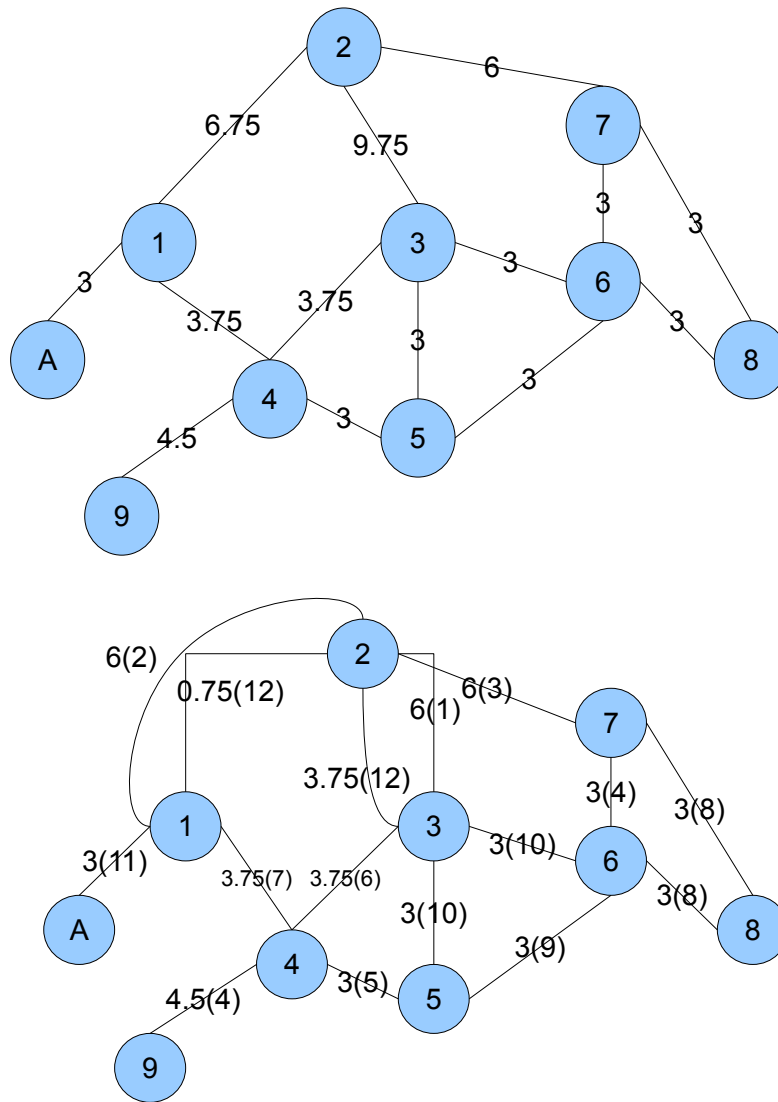


Figure 4.12 Dense WMN - JRCA result for traffic 0.75 X



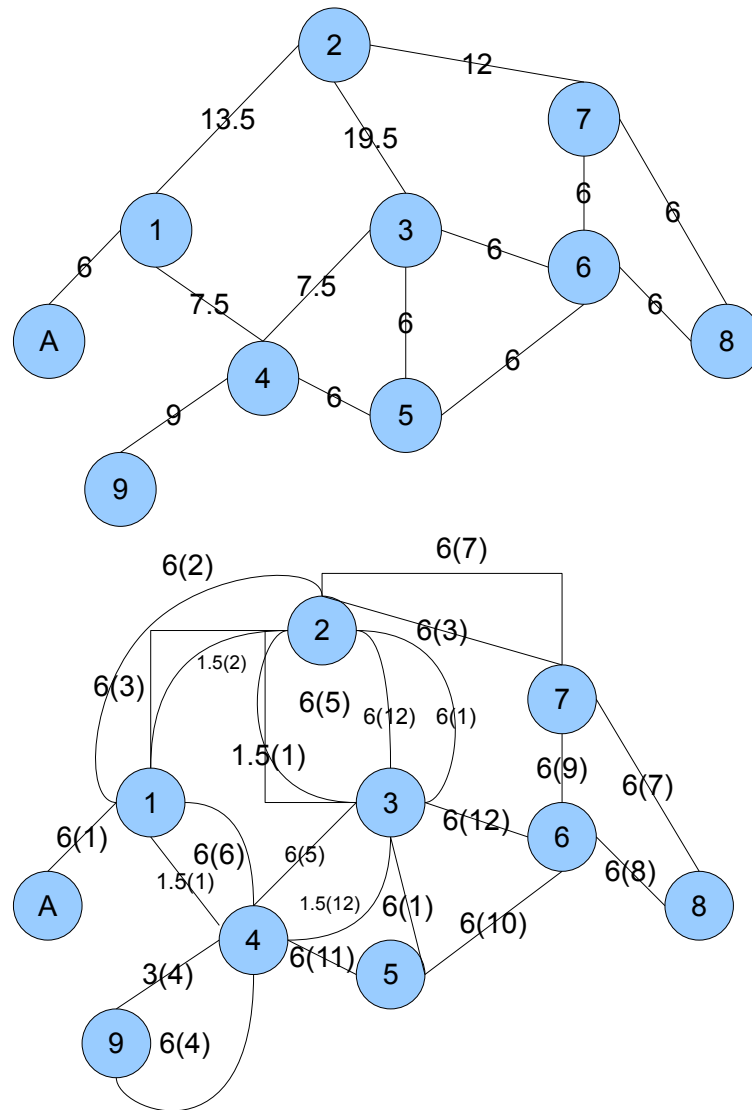


Figure 4.14 Dense WMN - JRCA result for traffic 1.50 X

Table 4.15 Study 3 - Throughput Efficiency vs Network Density

| <i>Scenario</i> | <b>1 (Dense WMN) (X=30 Mbps)</b> |              |              |               | <b>2 (Sparse WMN) (X=20 Mbps)</b> |              |              |               |
|-----------------|----------------------------------|--------------|--------------|---------------|-----------------------------------|--------------|--------------|---------------|
|                 | <b>Traffic</b>                   | <b>Links</b> | <b>Thput</b> | <b>Delay</b>  | <b>Efficiency</b>                 | <b>Links</b> | <b>Thput</b> | <b>Delay</b>  |
| 0.5X            | 15                               | 11.03 Mbps   | 375.42ms     | <b>73.53%</b> | 8                                 | 9.42 Mbps    | 98.97ms      | <b>94.2%</b>  |
| 0.75X           | 16                               | 16.42 Mbps   | 384.31ms     | <b>73%</b>    | 8                                 | 14.02 Mbps   | 105.19ms     | <b>93.47%</b> |
| 1.0X            | 18                               | 21.01 Mbps   | 423.78ms     | <b>70.03%</b> | 10                                | 18.55 Mbps   | 110.27ms     | <b>92.74%</b> |
| 1.25X           | 21                               | 26.24 Mbps   | 433.45ms     | <b>70%</b>    | 10                                | 22.77 Mbps   | 131.42ms     | <b>91.08%</b> |
| 1.50X           | 23                               | 31.00 Mbps   | 502.33ms     | <b>68.89%</b> | 10                                | 27.28 Mbps   | 167.13ms     | <b>90.93%</b> |

Table 4.15 shows the results of the performance of JRCA in dense and sparse WMNs. From the data presented, it can be observed that JRCA does very well in sparse WMNs delivering upto 94% of the client traffic at the IGW while in dense WMNs, JRCA is only able to achieve moderate efficiency (upto 74% of client's traffic).

The drop in the throughput efficiency in case of dense WMNs can be explained as follows: In dense WMNs, where the number of links are high (i.e. the number of links exceed the number of orthogonal channels available for channel assignment), there is a high probability that two links, in the interference range of each other, are assigned to the same channel. This leads to channel interference (inter-flow and intra-flow interference) and consequently decreases the network capacity (since only one link can be active at a time) and increases the packet loss.

We also make the following observations:

- *When the number of active links increases, the throughput efficiency decreases and the delay increases.* As the number of active links increases, the channel interference increases. Increased interference leads to a decrease in network capacity and consequently decreases the throughput efficiency. With decreased network capacity, the congestion level within the WMN increases. As the congestion level increases, the delay increases.
- *If the number of links are the same, when client traffic increases, the throughput efficiency decreases and the delay increases.* As the client traffic increases, the congestion level within the WMN increases. When the congestion level increases, the delay increases and consequently decreases the throughput efficiency.

## CHAPTER 5. CONCLUSIONS AND FUTURE WORK

Nowadays, typical wireless mesh installations generally utilize mesh routers equipped with a fixed number of radios. This approach is not adaptive to the bandwidth requirements of a 802.11-based wireless mesh network. To solve this problem, we need to be able to equip mesh nodes with as many radios as the net traffic flowing through that mesh node. However, this does not guarantee that we will obtain the maximum bandwidth just yet. We also need to assign channels to these radio interfaces in such a way that minimizes channel interference.

In this work, we have addressed this issue by proposing a scheme that can allocate both radio interfaces and assign channels to them, in such a manner that captures the client's traffic demands while minimizing channel interference. In particular, the JRCA scheme uses a clever heuristic to create new links between mesh nodes by considering surrounding channels and load distribution.

During the course of our study, we have examined the quality of our scheme by studying various parameters directly or indirectly influencing JRCA's performance. Firstly, we performed a comprehensive study that compared our scheme against scenarios which contained mesh nodes, each of which were equipped with a fixed number of radio interfaces. We discovered that our scheme performs very well against these fixed-radio scenarios. Secondly, we examined how the throughput varied when client traffic was varied. Lastly, we studied the trend of throughput efficiency vs network density, which revealed to us that JRCA's performance is directly contingent on the density of the wireless mesh network configuration.

Thus, our work has demonstrated the fact that by equipping mesh routers with a variable number (depending upon the net traffic flowing through the node) of radio interfaces and assigning them channels in a manner that minimizes interference is indeed more advantageous

than the naive strategy of equipping each router with the number of radios equal to the number of orthogonal channels. In addition to the scheme's several advantages, we believe that our work, to our best knowledge, is the first of its kind that couples the problem of radio allocation and channel assignment in a joint manner. With costs of 802.11-enabled radio interfaces plummeting, we believe that wireless mesh networks are becoming more of a commonplace. With our scheme, a network deployment engineer can easily and quickly install a 802.11-based wireless mesh network that caters to the ISPs' customer's needs.

As a part of future work, we would like to extend JRCA to use overlapping channels. Rate control for each of radios of a mesh router is another additional feature that we are planning to factor into our scheme for the future. Using rate control, we will be able to control the interference range of the radios thus allowing JRCA to minimize the co-channel interference to a greater degree.

**BIBLIOGRAPHY**

- [1] IEEE Standards Association  
*IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*  
<http://standards.ieee.org/getieee802/download/802.11-2007.pdf>
  
- [2] I.F.Akyildiz, X.Wang and W.Wang  
*Wireless Mesh Networks: a survey*  
Computer Networks and ISDN Systems, 2005, 47(4): pp. 445-487
  
- [3] *IEEE 802.11 LAN/MAN Standards Committee*  
<http://ieee802.org/11/>
  
- [4] Matthew S. Gast  
*802.11 Wireless Networks - The Definitive Guide*  
Second Edition, O'Reilly, ISBN-10 : 0-596010052-3
  
- [5] William Stallings  
*Wireless communications and Networks*  
Prentice Hall, 2005
  
- [6] Microsoft TechNet  
*802.11 Wireless Technical Reference*  
<http://technet.microsoft.com/en-us/library/cc737818.aspx>

- [7] Z. Chen, C. Qiao, Z. Chen, C. Qiao, J. Xu, T. Lee J. Xu, T. Lee  
*A Constant Approximation Algorithm for Interference Aware Broadcast in Wireless Networks*  
IEEE INFOCOM 2007, pp. 740-748
- [8] B. N. Clark, C. J. Colbourn, and D. S. Johnson  
*Unit diskgraphs*  
DiscreteMath., 1990, vol. 86, no. 1-3, pp. 165-177
- [9] Ashikur Rahman and Pawel Gburzynski  
*Hidden Problems with the Hidden Node Problem*  
23rd Biennial Symposium on Communications, 2006, pp. 270-273
- [10] Qualnet Simulator 4.5  
*Qualnet 4.5 User's Guide Scalable Network Technologies, Inc 2008*  
<http://www.scalable-networks.com/>
- [11] Ashish Raniwala, Kartik Gopalan, Tzi-cker Chiueh  
*Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks*  
SIGMOBILE 2004, pp. 50-65
- [12] KN Ramachandran, EM Belding, KC Almeroth, MM Buddhikot  
*Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks*  
INFOCOM 2006, pp. 1-12
- [13] S. Avallone and I. F. Akyildiz  
*A Channel Assignment Algorithm for Multi-Radio Wireless Mesh Networks*  
ICCCN 2007, pp. 1034-1039



- [14] Anand Prabhu Subramanian, Himanshu Gupta, and Samir R. Das  
*Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks*  
SECON 2007, pp. 1459-1473
- [15] J.Tang, G. Xue, and W. Zhang  
*Interference Aware Topology Control and QoS Routing in Multi-channel Wireless Mesh Network*  
ACM MobiHoc 2005, pp. 68-77
- [16] Ford, L. R., Fulkerson, D. R. (1956)  
*Maximal flow through a network.*  
Canadian Journal of Mathematics Vol 8, pp. 399-404
- [17] Ian F. Akyildiz, Xudong Wang  
*A Survey on Wireless Mesh Networks*  
IEEE Radio Communications, September 2005, pp. S23- S30
- [18] R. Draves, J. Padhye, B. Zill  
*Routing in Multi-Radio, Multi-hop Wireless Mesh Networks*  
Mobicom 2004, pp. 114-28
- [19] E.M. Belding-Royer  
*Multi-level Hierarchies for Scalable ad hoc Routing*  
ACM WINET 2003, Vol 9, no.5, pp. 461-78
- [20] H. Frey  
*Scalable Geographical Routing Algorithms for Wireless Ad Hoc Networks*  
IEEE Network Mag., July/Aug 2004, pp. 18-22
- [21] Liang Ma, Denko M.K.  
*A Routing Metric for Load-Balancing in Wireless Mesh Networks*  
AINAW 2007, Volume: 2, pp. 409-414

- [22] Deepti Nandiraju, Lakshmi Santhanam, Nagesh Nandiraju, Dharma P. Agrawal  
*Achieving Load Balancing in Wireless Mesh Networks Through Multiple Gateways*  
MOBHOC 2006, pp. 807-812
- [23] A Raniwala, T Chiueh  
*Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network*  
IEEE INFOCOM 2005, Vol 3, pp. 2223-2234
- [24] PH Hsiao, A Hwang, HT Kung, D Vlah  
*Load-balancing routing for wireless access networks*  
IEEE INFOCOM 2001, pp. 986-995
- [25] Y Yang, J Wang, R Kravets  
*Load-balanced routing for mesh networks*  
ACM SIGMOBILE Mobile Computing and Communications Review, 2006, pp. 3-5
- [26] AK Das, HMK Alazemi, R Vijayakumar, S Roy  
*Optimization models for fixed channel assignment in wireless mesh networks with multiple radios*  
IEEE SECON 2005, pp. 463-474
- [27] P Kyasanur, NH Vaidya  
*Routing and interface assignment in multi-channel multi-interface wireless networks*  
IEEE Wireless Communications and Networking Conference, 2005, pp. 31-43
- [28] MK Marina, SR Das  
*A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks*  
2nd International Conference on Broadband Networks, 2005, pp. 381-390

- [29] BJ Ko, V Misra, J Padhye, D Rubenstein  
*Distributed channel assignment in multi-radio 802.11 mesh networks*  
Wireless Communications and Networking Conference (WCNC), 2007, pp. 3978-3983
- [30] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy  
*Experimenting with a Multi-Radio Mesh Testbed*  
WiNMee 2005
- [31] A. Hertz and D. de Werra  
*Using Tabu Search Techniques for Graph Coloring*  
Computing, vol. 39, no.4, 1987, pp. 345-351
- [32] Michel X. Goemans and David P. Williamson.  
*Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.*  
Journal of ACM, 42(6), 1995, pp. 1115-1145.