# CHANNEL ASSIGNMENT FOR THROUGHPUT IMPROVEMENT IN MULTI-RADIO WIRELESS MESH NETWORKS

by

Shivaram Venkata Tenneti

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

November 2007

APPROVAL

of a thesis submitted by

Shivaram Venkata Tenneti

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Division of Graduate Education.

Dr. Jian Tang

Approved for the Department of Computer Science

Dr. John Paxton

Approved for the Division of Graduate Education

Dr. Carl A Fox

## STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U. S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Shivaram Venkata Tenneti

November 2007

# ACKNOWLEGEMENT

I would like to thank Dr. Tang, my academic advisor for his help and guidance throughout the project. I am indebted to him for his patience and guidance over the past year. I am grateful to Dr. Rocky Ross for his advice during my initial semester at MSU, and also for serving on my committee.

Special thanks to Dr. Richard Wolff who introduced me to the wonderful world of networking and exposed me to fiber optics and wireless networks during my early days at MSU. I thank him for taking time to provide me with his valuable input and serving on my committee.

I thank all my friends at CS and EE for helping with my project, providing useful tips and tricks in designing some of the components. Their contributions are truly appreciated.

Finally, I want to thank my family and friends for the infinite support and encouragement which is incredibly hard to express in words. My Mom, Dad, Shanti, Micheal, and rest of the family for their never ending love, support and encouragement, my friends Amit, Arnav, Rajini, the list goes on, who have all provided me support during though times and finally I thank dear GOD, with whom all things are possible. Never say die!

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

# ABSTRACT

Wireless mesh networks offer many advantages in terms of connectivity and reliability. They provide multiple paths between nodes and are self healing. Traditionally, wireless mesh networks were typically used with nodes equipped with a single radio. There are however, limitations in single radio wireless mesh network, such as lower throughput and its limited use of the available wireless channels. This thesis focuses on the Hybrid Channel Allocation scheme which efficiently utilizes multiple wireless interfaces to achieve better throughput thereby increasing the network capacity. In this thesis we introduce and evaluate different methods to improve the network throughput of a multi radio wireless mesh network. We present the Random and Greedy channel assignment protocols which utilize multiple radio interfaces to improve the throughput and minimize the radio interference of the wireless network. We also implement the Superimposed Code based channel assignment proposed in [5], to evaluate and compare its performance with the Random and Greedy protocols. These channel assignment protocols allow different nodes in the same network to communicate with each other without causing too much interference to their neighbors.

Network scenarios have been designed using the discrete event simulator Opnet Modeler 11.5$^{\text{TM}}$. These network scenarios have been created to compare and evaluate the performance of the channel assignment protocols under different conditions. Simulation results are presented and discussed.

# INTRODUCTION

Traditionally in wireless networks, nodes[1] were operating with a single radio[2], due to the cost associated with having multiple radios on a node, which was high. Several methods were proposed which aimed to improve the network throughput, for single radio wireless networks. However, with lowering costs, it has become possible to equip a node with multiple radios. Having multiple radios on a node opens up several possibilities and options as to how these radios can be utilized to improve some of the important characteristics of the nodes. Several interesting studies have been performed on multi radio nodes in [14, 15, 9] and have concluded that in some cases using multiple radios improves the throughput.

In this thesis we us the concept of a multi radio mesh node to analyze the performance of a mesh network in different conditions with different protocols. We look at new ways to try and improve the network throughput in a wireless mesh network. Wireless mesh networks (WMN) are gaining popularity and are by far one of the favorite network topologies for wireless networks which are required to cover a large area. Several commercial establishments have deployed wireless mesh networks using the very popular IEEE 802.11 standard to provide network connectivity over a large geographical area. An example of such a mesh network can be found here [19].

---

[1] Wireless device
[2] Network interface card

## Overview of 802.11 and Wireless Mesh Network

The IEEE 802.11 standard [10] also know as “Wi-Fi” (Wireless Fidelity) is a popular wireless technology which is extensively used in a lot of establishments. The expense to deploy the 802.11 networks is very cost effective thus is very attractive to both consumers and for commercial establishments. It is therefore a natural choice for research and innovation to improve the performance of networks using the 802.11 technology.

### IEEE 802.11 Standard

The 802.11 comes in several different versions, the most popular being a/b/g. The chief difference between them is that ‘b/g’ versions operate on the 2.4GHz spectrum and ‘a’ operates on the 5.8GHz spectrum. The various versions 802.11 standard differ in the physical characteristics that determine a nodes operation in a wireless local area network or LAN. The 802.11 standard[10] defines specifications such as the channel characteristics including the frequency of operation and the channel bandwidth, modulation scheme, the transmission power which determines the transmission range of a node, etc.

The 802.11b standard was specified to use either Frequency hopping spread spectrum (FHSS) or Direct Sequence Spread Spectrum (DSSS) and operates in the frequency range of frequency range of 2.400 to 2.4835GHz. This band is divided into 11 overlapping channels, each 30MHz wide and the allowed data rates are 1Mbps, 2Mbps, 5.5Mbps and 11Mbps. The 802.11b has become the most popular 802.11 technology for wireless LAN's and has been widely deployed to support a wide variety of applications such as file transfer, video and voice streaming, etc.

There are however, limitations in the 802.11b network. Although there are 11 channels available for communication in the 802.11b, there are only three non overlapping channels. Hence, there are only three channels 1, 6 and 11 which provides for an interference free communication in a network. The maximum data rate allowed is 11Mbps, which was very inadequate for the kind of applications that were supported.

The 802.11g standard was introduced to address the data rate limitations in 802.11b. The 'g' standard is backward compatible with 'b' and operates on the same 2.4Ghz frequency spectrum and the maximum data rate allowed is 54Mbps. The drawback of using the 802.11g is the transmission range is decreased to around 90m, when compared to 400m for the 802.11b.

The IEEE 802.11a was introduced to include a higher throughput of up to 54 Mbps [13], by using up to 13 non overlapping channels operating in the 5 GHz band, which is achieved by using Orthogonal Frequency Division Multiplexing (OFDM) [3].

## Wireless Mesh Networks (WMN)

Mesh networks are very robust, which offer reliable communication between nodes in the network. Mesh networks are available in two configurations, fully connected mesh network and a partially connected mesh networks. A fully mesh network typically consists of several nodes which are connected to each other. In a partial mesh

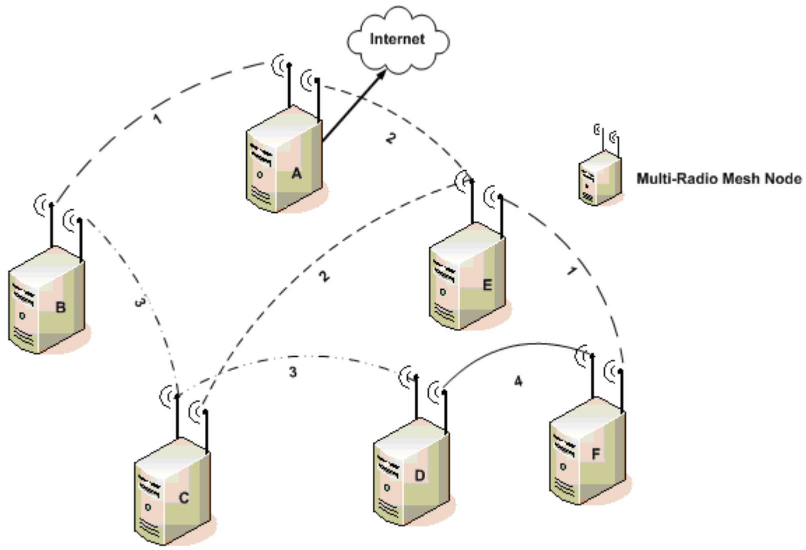


Figure 1. A Multi-Radio WMN using four channels.

network, some nodes are connected to all the others, but some are connected only to those other nodes with which they exchange data. Fig. 1, shows a partially connected mesh network in which all nodes have multiple radios operating on four channels.

In a multi-radio multi-channel (MR-MC) network, mesh routers are equipped with multiple radios. Each radio on a router can be tuned to any of the available multiple channels as shown in fig. 1. Some mesh nodes can also be gateways to other networks such as the internet and so the other nodes in the network can connect to an external network through the gateway node. Two neighbor nodes wishing to communicate establish a wireless link between them by tuning at least one of their interfaces to the same channel. For example in fig. 1, if node C wants to initiate communication with node E, then C will tune its interface to channel 2 and establish a wireless link from C to E. Similarly, if node B intends to communicate with the internet, it needs to establish a link to the gateway node A by tuning its interface to channel 1. Hence, every node in the network can communicate with each other simply by tuning the radio interfaces of the communicating nodes to the same channel.

The major advantage of mesh networks over other networks is that a destination can be reached via multiple paths. They are self healing, i.e., if a node fails the network can still operate by finding an alternate path making it very reliable. The main drawback of the mesh topology in a wired network is that its expensive, due to the large number of cables and connections required. However, in the case of wireless networks, a mesh topology is very inexpensive and very effective due to the absence of wires and its self healing property. A wireless mesh topology can increase the performance and reliability of the network by providing multiple paths between the source and the destination nodes.

The problem of implementing a wireless network such as that in fig. 1, is very challenging chiefly due to radio interference. Unlike in a wired network, one of the main reasons for sub par network performance in a wireless network is radio interference. When two neighbor nodes are operating on the same channel they interfere with each others communication. A wireless network which experiences radio interference will degrade the performance of the network.

## Channel Assignment in WMN

Wireless Technologies such as the 802.11a, provides several non overlapping channels, which means that nodes can be transmitting or receiving at the same time on different channels without interfering with each other. However, to ensure such interference free communication, all the nodes within each other's interference range must be on different channels. This problem of making sure that all the interfering nodes are assigned different channels is known as the *Channel Assignment Problem*.

The Channel Assignment Problem becomes more challenging to solve when each node is equipped with multiple radios. This is because when a node is equipped with multiple radios, if both the radios operate on the same channel then there is interference from a nodes own radio which leads to packet collisions. However, if it can be ensured that in a node, if each of the multiple radio interfaces operate on different channels then this will improve the networks performance. The benefits of using Multi-Radio Multi-Channel (MR-MC) nodes are studied in [16, 17] which consider channel assignment.

If all the radios in a MR node are on the same channel it will lead to packet collisions due to interference from one of the radios on the same node. There is also another case of interference, which is interference from the neighbor nodes, which also has significant impact on the network performance. Hence, the problem of assigning different channels to multiple radios in a node and the neighbor node becomes very important in order to minimize interference. These two cases together are the main focus in [4, 7, 8] which attempt to solve this "Channel Assignment" problem in different ways.

There are several channel assignment strategies, that consider channel assignment to the radio interfaces in a MR node so as to minimize interference. In [5], channel assignment to all radios is considered in a static fashion. Their channel assignment algorithms are localized and designed for a mesh network with a more general peer-to-peer traffic pattern. They present two algorithms one of which for a general broadcast and the other for unicast data transmissions. In [18, 9], a common default channel is introduced to facilitate channel negotiation for data communication. The following gives a brief description on some of the channel assignment strategies.

Static Assignment

In a static assignment strategy, each interface is assigned to a channel for long time durations. There are two different approaches to in this strategy. The first is the common channel approach in which the radio interfaces of all the nodes in the network are assigned to common channels. In the second approach, the radio interfaces in different nodes may be assigned to different channels, which is the varying channel approach.

In static assignment strategy, nodes sharing a common channel on at least one of their radio interfaces can communicate with each other, while other cannot. Hence, deciding which nodes can communicate in the network can affect the network performance.

Dynamic Assignment

In this strategy any interface can be assigned to any channel and interfaces are allowed to switch from from one channel to another. A network using such a strategy needs some kind of synchronization mechanisms to enable communication between nodes in the network. An example of such mechanism is that nodes can periodically visit a common channel [21] and tune the interfaces accordingly to establish a communication link.

The main hurdle in the dynamic strategy is to decide which channel to switch the interface and also when to switching needs to occur so that interference free communication is ensured.

Hybrid Channel Assignment Scheme

In the hybrid strtegy all the nodes are MR nodes, in which the multiple radios are divided into two groups. One group of radios are assigned fixed channels for receiving packets there by ensuring connectivity, and switchable channels are assigned to the other group [4]. Whenever data transmissions are required the switchable radio of the source node switches to the fixed channel of the destination node. Thus, the channel assignment for the fixed radios are the most important aspect of the hybrid strategy.

## Motivation for the Research

The motivation for using multiple radios on node is to achieve higher throughput. A network with higher throughput will deliver better quality of service for bandwidth intensive applications such as video and voice streaming, voip[1], etc [1, 2].

A high throughput capacity WMN can support several users running bandwidth intensive applications simultaneously. The other advantage of using a WMN is that the geographical range of the network can be increased without disturbing the existing network topology. This is particularly desirable in network designed for small town or a campus, which may grow at some point in the future. If there is a need to increase the geographical range of a WMN then there is no need for a total network redesign. Instead, a simple solution for this problem is to add another, or a few, new mesh node(s) depending on how much area needs to be covered.

Hence, WMN's provide an attractive solution towards providing network connectivity over an unlimited range. There are several commercial vendors who specialize in providing mesh networking solutions. Thus, there has been a lot of interest and activity in this are with many interesting ideas suggesting solution for some of the challenging problems.

Traditionally in wireless networks nodes with single radios were used for data communication. However, using single radio for multiple channels is not a feasible solution. For example, in a single radio multi channel network, if two communicating

[1] Voice over IP

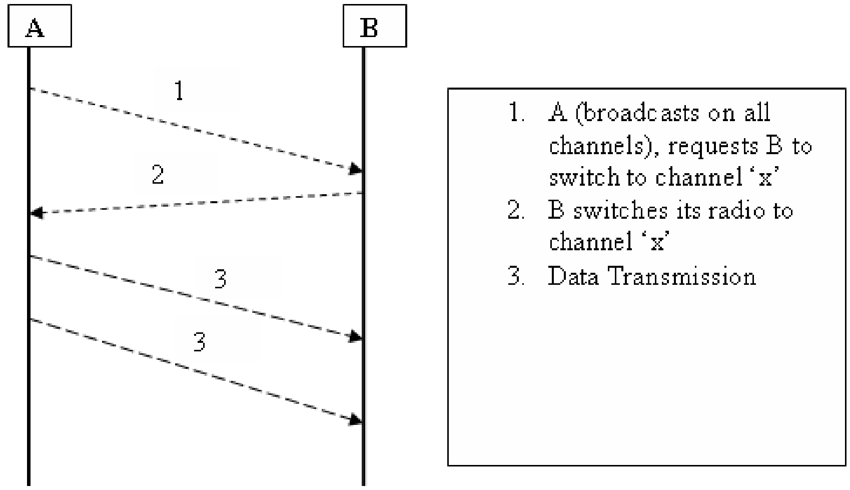


Figure 2. An example of channel synchronization message exchanges.

nodes are on different channels then they cannot communicate. For data communication to take place between two nodes they have to be on the same channel. Hence, for data communication to take place there has to be some mechanism to ensure synchronization between the communicating nodes.

An example of the control message exchange between the source and destination nodes before data transmission begins is given in fig. 2. Node A is the source node which intends to initiate data communication with node B. However, node A may not know the channel on which node B is operating at any given time. Hence it broadcasts a control message to B on all channels requesting B to switch its radio to channel 'x'. Once B receives the request from A, it switches to channel 'x' and replies back to A on channel 'x'. Data transmission begins once A recieves the reply from B.

Considering the above example, if the data transmission takes place on a multi hop network, the above approach is very inefficient due to the amount of control message exchanges that take place between the nodes and also the time to switch to different channels. Also, there is no way to know if there is interference from the neighbors of the communicating nodes. If the neighbors are on the same channel, then it will drastically reduce the throughput due to interference from neighbor nodes. Also, in a single radio network the nodes are half duplex, so they cannot send and receive data in parallel.

Now consider the same case for a multi radio network using the hybrid scheme describe in [4]. For simplification purpose, assume that each node in the network has two radios. One radio will send data and the other radio will receive data. Assume that all the nodes in the network know the channel on which their neighbor nodes will receive data. Control packets, such as a HELLO packet, can be used to exchange this information among neighbor nodes. This is a fair assumption, because nodes can broadcast this information to all its neighbors.

Now, if node A wants to initiate data communication with node B, it will switch the channel on its transmitting radio to the channel on which B will receive. As we can see, there is no need for A to broadcast a control message to requesting B to change channels. Also, both nodes A and B can receive and send data from and to other nodes without interrupting the data transmission between them.

Thus the benefit of using multiple radios for such networks is twofold. One is that the necessity for the control message exchange between the source and destination nodes is eliminated. Second, is the ability to send and receive data in parallel make them full duplex.

However, when using multiple radios, it must be ensured that the interference between neighbor nodes must be at a minimum to ensure maximum throughput, if not the performance of a multi radio network will degrade and may also perform worse than a single radio network. In this thesis we attempt to solve this problem using different approaches to ensure that the throughput is maximized in a MR-MC network.

The benefits of using Multi-Radio Multi-Channel (MR-MC) nodes are studied in [16, 17]. Load aware channel assignment is considered in [15]. In [4], channel assignment is considered using the hybrid strategy. However, in this approach the nodes are assigned channels at random. In [5], channel assignment to all radios is considered in a static fashion. However, with static channel assignment it is very difficult for a network to reconfigure itself in case of a node failure or a channel change by a node in the network. There are also several ideas which make use of a MR-MC nodes in a WMN which will boost the network performance, some of them have been studied in [22]. This thesis is aimed at contributing towards this effort.

## Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we describe the system model and formally define the problem followed by an introduction to the channel assignment protocols. Chapter 3 gives an introduction to OPNET Modeler and we present the simulation scenarios that were used to evaluate the performance of the algorithms, followed by the results obtained from the simulation. Chapter 4 summarizes and concludes the thesis.

# CHANNEL ASSIGNMENT PROTOCOLS

In this chapter we introduce the Random and Greedy channel assignment protocols that we use in this thesis. We also implement the Superimposed code based channel assignment protocol proposed in [5] to compute the channel for a unicast communication between two nodes. We describe the system model and formally define the problem. We also list the all the assumptions.

In Random Channel Assignment, every node selects a random channel and assigns it to its fixed interface(s). All the neighbor nodes are aware of each other fixed channels based on the information exchanged using a *'HELLO'* packet. The format of a HELLO packet is shown in fig. 3.

In the Greedy Channel Assignment, every node, instead of straight away selecting and assigning a channel to is fixed interface, waits for a random time period. If during

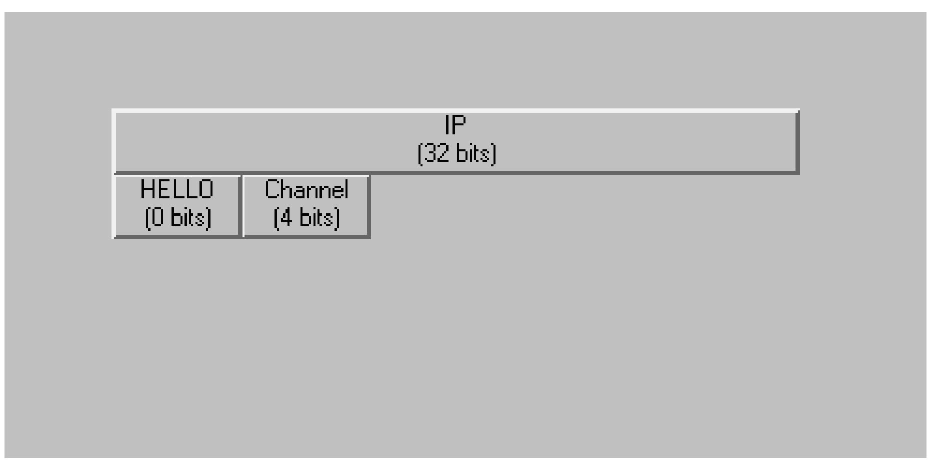


Figure 3. HELLO Packet Format.

this time period, all the neighbor nodes pick their fixed channels, then the node will determine the least used channel among its neighbors and assign that channel to its fixed interface. The channel information among the nodes is exchanged using the HELLO packet shown in fig. 3.

The Superimposed Code based Channel Assignment proposed in [5], is a unique approach towards solving the channel assignment problem. In this approach, every node has a unique codeword, whose length is equal to the number of available channels. The codewords consist of a series of 1's and 0's, which indicate primary and secondary channels. A node always favors its primary channel to assign it to its fixed interface.

## System Model and Problem Formulation

Consider a multi-radio wireless mesh network with $N$ stationary wireless mesh nodes in which there are C non-overlapping channels. The number of radios on each node $v$ is $R_v$ $(2 \leq R_v < 4)$. We consider a hybrid assignment scheme as in [4], and assume that once a channel is assigned to its fixed interface it will not change. We assume the network consists of several static nodes[1] communicating over a wireless link. We define throughput as follows.

[1] The location of the nodes is fixed

*Definition 1* (Throughput). We define throughput as the number of data bits received by the destination node during a unicast data communication between a source node and a destination node in the network.

*Definition 2* (Interference). For any node $u \in N$, a node $v \in N$ is an interferer of u if v's transmission interferes with u's transmission.

Hence, we define the problem as follows,

*Definition 3.* Given a set of nodes N=$[n_1, n_2 ... n_k]$, and set of channels C=$[c_1, c_2 ... c_k]$ (C < N), the goal is to assign channels to all the nodes in N such that the throughput is maximized and the radio interference is minimized.

Once a node is assigned a channel it broadcasts this information in a HELLO packet to all it neighbors in a two hop neighborhood. For data transmission, each node randomly picks a node from its neighbor table. Since we don't consider routing in this thesis, data transmissions occur over a nodes one hop neighborhood.

## Random Channel Assignment

The random channel assignment algorithm is a straight forward algorithm in which the input for is a set of available channels from which a node picks a channel at random assigns it to its fixed wireless interface. The pseudo code is as follows,

Once channel assignment for a node is complete, the node broadcasts this information over its one hop neighborhood in a HELLO packet. When a node receives a HELLO packet from its neighbor node, it updates its neighbor table with the ip

**Algorithm 1** Random Channel Assignment

```
Node(u) ← ∅
for all i ∈ C do
    i ← rand(C) //Pick a channel From C at random
    Node(u) ← i //Assign the channel to u
end for
```

address and the channel contained in the HELLO packet. This information helps the node to determine what channel it needs to assign to its switchable radio when data transmission is initiated with the destination node. The pseudo code for the HELLO packet broadcast is as follows,

**Algorithm 2** HELLO Packet Broadcast

```
List_neighbor(u) ← ∅
time ← ∅
for every 5 seconds do
    HELLO_channel,ip,hops(u) //Broadcast HELLO packet with channel information
end for
if HELLO_channel,ip,hops(v) then
    List_neighbor(u) ← HELLO_ip(v)
    List_neighbor(u) ← HELLO_channel(v)
    List_neighbor(u) ← HELLO_hops(v)
end if
```

The structure of the neighbor table contains a list of (Ip address, channel, hops). The node's own ip-address and randomly picked channel is always the first entry in the list. The node then broadcast's this channel information to its entire neighborhood which is specified in *hops* on all channels through a HELLO packet. A timer is set to broadcast the Hello packet every 5 seconds.

When a node receives a HELLO packet, it retrieves the source ip-address and channel fields from the HELLO packet and adds it to its Neighbor Table, if the entry does not exist in its Neighbor Table. When the neighbor table is populated with information from the neighbors, the node randomly picks an ip address from the Neighbor Table and initiates data transmission.

In a real world case, some external event would trigger the data transmission, such as web browsing, voice or video streaming, etc.

## Greedy Channel Assignment

In the basic channel assignment algorithm, since all the nodes in the network pick a channel at random, there is no guarantee that the channel picked by all the nodes in the network will be different. In the worst case, we can assume that using algorithm 1 all the nodes in the network can pick the same channel. The interference in this case is maximum and the network will lot of packet collisions which will result in worse throughput.

To avoid such a situation, we introduce a different approach to solve the channel assignment problem. Instead of picking a channel at random right away, each node will wait for a random time period. The reason for the wait period is that, during this time there is a possibility that some node might already pick a channel and then broadcast this information to its neighbor. Therefore, when the waiting time is over a nodes channel table may have entries from some of its neighbors. So, instead of picking a channel at random it can now pick a channel that is least used by its neighbors. The pseudo code for the Greedy Channel Assignment is shown in algorithm 3.

**Algorithm 3** Greedy Channel Assignment

$C(u) \leftarrow \emptyset$
$List_{channel}(u) \leftarrow \emptyset$
$List_{neighbor}(u) \leftarrow \emptyset$
**while** $timeout \neq 0$ **do**
  **if** $Packet_{HELLO}(v)$ **then**
    $List_{neighbor} \leftarrow IP_{address}(v)$
    $List_{neighbor} \leftarrow Channel(v)$
    $List_{channel} \leftarrow Channel(v)$
  **end if**
**end while**
**if** $List \neq 0$ **then**
  **for all** $Channel(v) \in List$ **do**
    $SORT_{List_{channel}}$
    $C(u) \leftarrow List_{channel}(0)$
  **end for**
**else**
  **for all** $i \in C$ **do**
    $i \leftarrow rand()$
    $C(u) \leftarrow i$ *//Assign a random channel from the set of channels*
  **end for**
**end if**

Consider the example network as shown in fig. 4. We will explain the Greedy Channel Assignment from Node A's perspective. Initially node A waits for a random time period during which it accepts any incoming HELLO packets from its neighbor nodes. Next, it initializes the Neighbor Table and Channel List.

- The *Neighbor Table* is a list of IP Addresses and Channel numbers. The first entry in the Neighbor Table is a nodes own IP Address and its fixed channel.

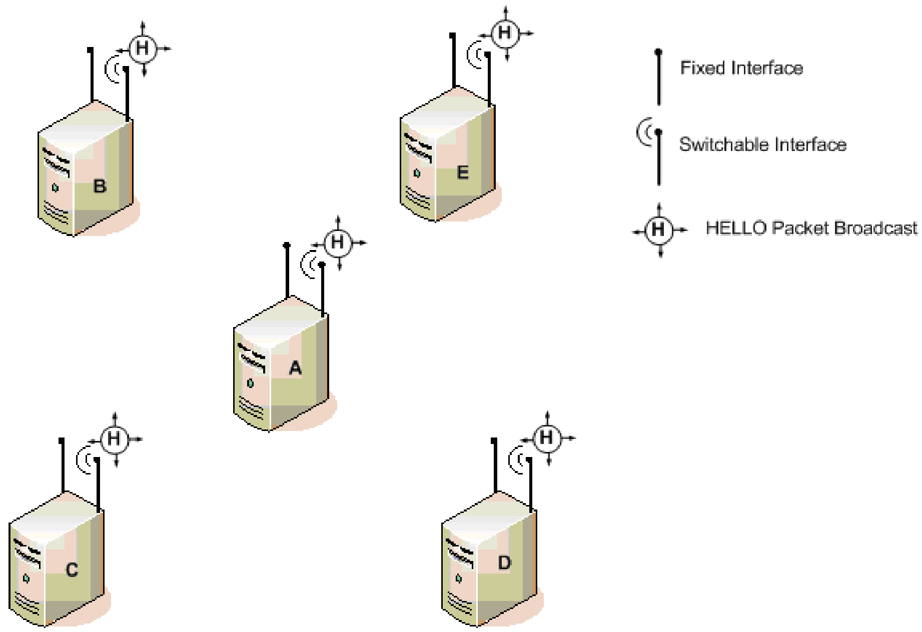


Figure 4. HELLO packet broadcast.

Therefore, the first entry in node A's neighbor table will contains its own IP Address and its fixed channel number.

- The *Channel List* contains a list of channel numbers on which every node is operating on and count which keeps track of the number of nodes operating on the same channel. For example, once node A selects its fixed channel, it will enter the channel number into its Channel List and increment the value of count to 1. If there is another node operating on the same channel as node A's then the count will be incremented to 2, and so on.

During the wait period node A will listen/accept any incoming HELLO packet from its neighbor nodes. If node A receives a HELLO packet, it populates its Neighbor

Table by retrieving the source ip address and the channel number from the packet and also adds the channel number into its channel list as described above. An example of the Neighbor Table and the Channel List is shown in table 1.

Once, the wait period has ended, node A will then look into its Neighbor Table for any entries. If the Neighbor Table is populated then, node A will pick the least used channel by looking into its Channel List and selecting a channel that has the least count value. If two channels have the least count, then the node is free to pick one among them. In our case we sort the list in ascending order on count and select the very first entry in the list.

If at the end of the wait period, if node A's Neighbor Table is empty, then it will randomly select a channel, add it to its neighbor Table and Channel List and assign the channel to its fixed interface. Once, node A selects a channel it broadcasts this information using a HELLO packet to all its two hop neighbors.

Let us, for example, assume that node A selects channel 5 as its fixed channel. Then enters this information into its tables, and broadcasts this information using a HELLO packet to all its neighbor nodes as shown in fig. 4. Once, node A starts receiving HELLO packets from its neighbors it builds its tables using the information in the HELLO packets which is shown in table 1.

Once there are enough entries in the Neighbor Table, node A will initiate data transmission by selecting one of its one hop neighbors at random. Fig. 5 shows the unicast data transmissions between A and its neighbors. In the fig. 5, if node

Table 1. Tables at Node A.

| Neighbor Table | | Channel List | |
|---|---|---|---|
| IP Address | **Channel** | Channel | **Count** |
| 192.168.1.1 | **5** | 5 | **1** |
| 192.168.1.2 | **2** | 2 | **1** |
| 192.168.1.3 | **3** | 3 | **2** |
| 192.168.1.4 | **4** | 4 | **1** |
| 192.168.1.5 | **3** | | |

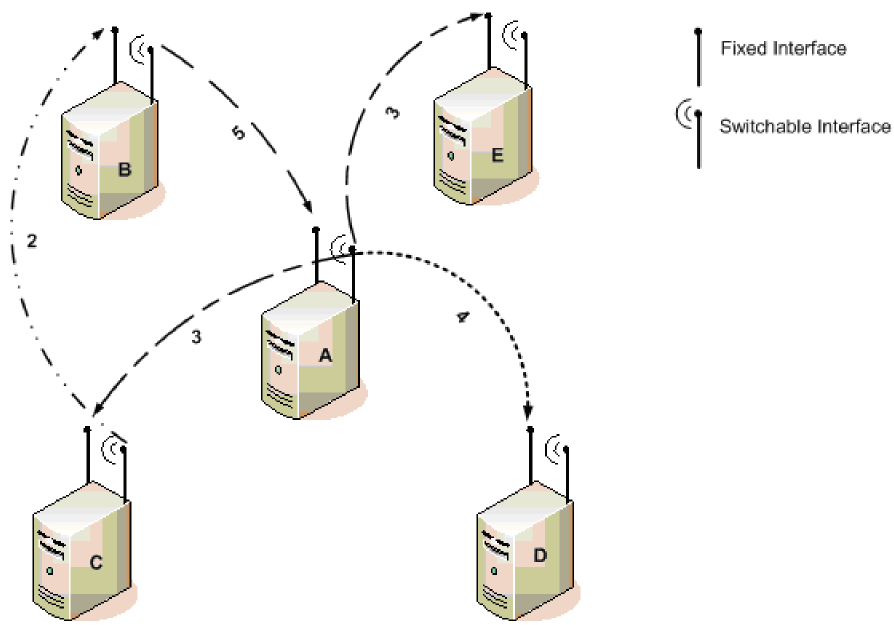


Figure 5. Data Transmission.

A intends to initiate unicast data communication to node E, then it will tune its switchable interface to node E's fixed channel, which is channel 3, and requests unicast communication. If node E is not already involved in a unicast communication with another node, it accepts node A's request and then the data communication begins. Notice, that node C and B are also locked in a unicast data communication on Channel 2, and does not interfere in the communication of A and E. Also, node A can be receiving data from node B on channel 5 hence making it a full duplex node.

We assume that unicast data communication between two nodes takes place for a long time, which is a fair assumption, since a mesh network mainly serves as a backbone network where the nodes can be access points. Also, another reason is that if node intends to communicate with several nodes frequently then it has to switch channels on its switchable interface frequently. This frequent switching of channels by a node introduces a delay known as 'switching delay' [4]. The switching delay can degrade the performance of the network and thus we have avoided it.

However, in a real world scenario, there can be a situation in which node A might have to terminate its unicast communication with node E and initiate a new unicast link with another node for example node D in fig. 5.

In the approach described above, if all the nodes are not to pick the same channel, then care must be taken to ensure that the wait period in the beginning is random. Once, we can make sure about that, it is almost impossible for the nodes in the network to pick the same channel. Hence, we can avoid the worst case scenario that can occur in algorithm 1.

However, when the network size is large there may be a possibility that the number of nodes operating on the same channel can be high depending on the node density. But it still is better than the channel assignment scheme described in algorithm 1.

## Superimposed Code based Channel Assignment

In this section we will talk about a unique approach towards solving the channel assignment problem. In this method, every node in the network generates unique codeword [5], whose length is equal to the number of channels that are available. When the codeword's from all the nodes are combined to form a matrix, they should satisfy the property of a superimposed code.

This unique approach to solve the channel assignment problem has been proposed in [5], which assigns channels to the fixed radios based on codewords. For each node, the available channels are divided into primary and secondary channels.Thus, a codeword contains a series of 1's and 0's, which indicate the primary and the secondary channel for a node. Nodes always prefer the primary channels, that is secondary to all its interferers, for the fixed channel assignment and use the secondary channel only when the primary channels are not available.

The basic idea in [5] is that all nodes generate a codeword that is unique and when the codewords are combined together to form a matrix they must satisfy the

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{pmatrix}_{13 \times 13}$$

Figure 6. A (3,1,13) superimposed code [5].

property of a superimposed code. A superimposed code is formally defined as

**Superimposed code (SC)**: A N $\times$ t binary matrix X is called a superimposed code of length N, size t, strength s, and listsize = L - 1 if the Boolean sum of any s-subset3 of the codewords of X covers no more than L - 1 codewords that are not components of the s-subset. This code is also called a (s,L,N)-code of size t.[5]

Fig. 6, is a superimposed code matrix and by the definition it is a (3,1,13) code if size 13. The rows of the matrix indicate the number of channels, which in this case is 13, and the columns indicate the number or nodes in the network, which is also 13. Therefore the size of the codeword for each node is 13 in this case.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

Figure 7. A (2,1,13) superimposed code.

There is no mention of how the codewords are actually generated in [5, 11]. We attempt to generate our own codewords which satisfy the property of the superimposed code for different network sizes for the channel assignment algorithms.

Generating a superimposed code for this thesis has been very challenging, which included try and test methods. It was however, not possible to generate a superimposed code matrix for large networks, because the computations required to generate such a matrix consumed a lot of time, and most of the matrices generated did not satisfy the superimposed code property. We therefore, use small network sizes for analyzing the performance of this method. Fig 7, is a 13 $\times$ 20 matrix which is a (2,1,13) superimposed code. It was not possible for us to generate a matrix for large networks and for this reason the largest network size is 40.

**Algorithm 4** Superimposed Code based Channel Assignment

$Interferers(v) \leftarrow \emptyset$
**while** $!HELLO(v)$ **do**
  **for all** $c \in C(u)$ **do**
    $HELLO_{REQ}(u) \rightarrow v$ *//Request interferers list from v*
  **end for**
**end while**

Since we generated the superimposed codes matrix separately, we assume that all the nodes are aware of the codewords for nodes in the network. There is thus no need for any HELLO packet exchanges to broadcast a node's codeword to its neighbor. Therefore, nodes can immediately start the data communication. For unicast data communication, we use the algorithm described in [5]. However, in [5] they assume that the source nodes are aware of the destination nodes interferers. We however, do not make such an assumption, therefore we use the HELLO packet exchanges to request the destination node to disclose all its interfering nodes. We use, the unicast algorithm described in [5] as a subroutine which will be invoked only when we know the interferes of the destination node.

Consider the network in fig. 8. Assume that node A intends to initiate data communication with node E. Assume that, node B is the interferer of node E, and nodes C and D are interferes of node A. Now, node A broadcasts a unicast request to

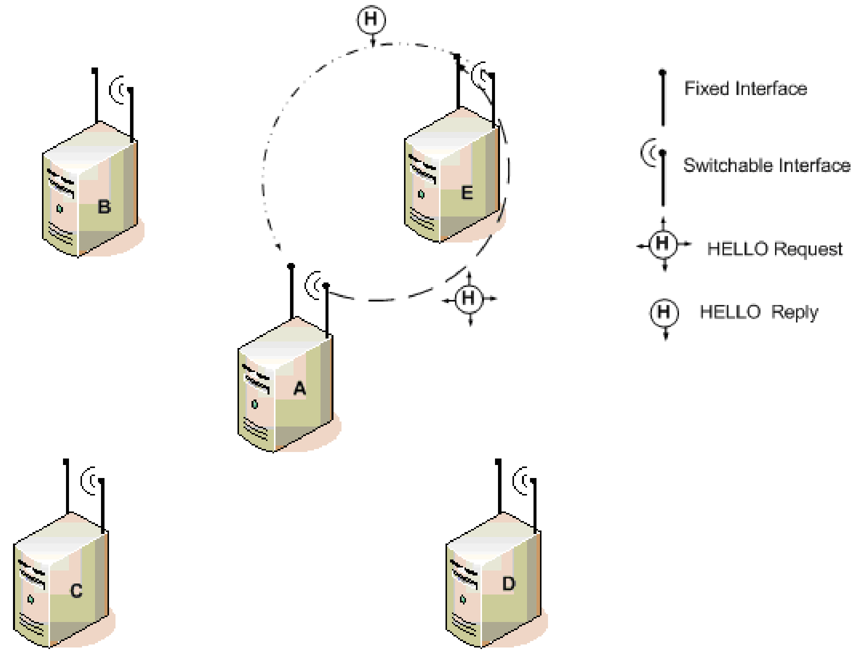


Figure 8. Data Communication request using a HELLO packet.

node E in a HELLO packet. When node E received the request from A it responds to the request by sending its interferer list to node A in a HELLO packet.

When node A received a response from node E, it will compute the channel on which it can initiate data communication with node E using the process described in [5]. Once a channel is selected, node A will again broadcast the channel number to node E in a HELLO packet. Node E, upon receiving the packet will assign its fixed interface to the channel computed by node A and send a reply to node A indicating the switch.

The data transmission will begin once node A received the clear signal from E as shown in fig. 9. The trade off with this protocol is the number of HELLO packet

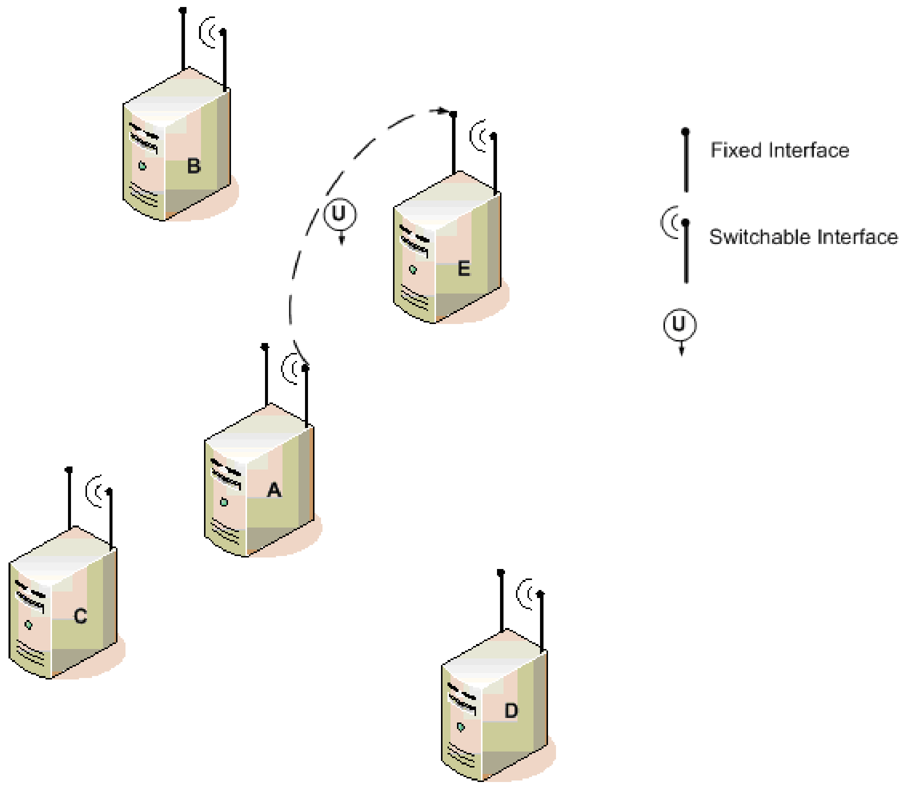


Figure 9. Unicast data communication between A and E.

exchanges required to initiate a unicast data communication between two nodes is considerably higher. In the next chapter we give the analyze the simulation results and provide some trade off's on the channel assignment protocols described here.

# MODELLING AND SIMULATION

In this chapter we will introduce the models and the simulations settings that were designed in the discrete event simulator OPNET Modeler 11.5$^{TM}$, which is used to perform simulations for this thesis.

The OPNET Modeler is a popular software for conducting simulations on a wide range of networks. It has a rich set of features to deal with most the network technologies available. ranging from software. It provides a several editors each of which enable us to change characteristics such as the network size, node model, etc. We used the project editor in the Modeler to create simulation scenarios.

We designed four different scenarios which we use to analyzing the performance of a network operating on the channel assignment algorithms described in chapter 2.

Scenario's 1 and 2 operate on the 802.11a, which provides 13 non overlapping channels, so we evaluate the performance of all the three algorithms in these two scenarios, with the network size varying from 13 nodes to 120 nodes. We also change the number of interfaces in scenario 2, to three from two in scenario 1. When using 3 interfaces on a node, there are two different cases of utilizing the interfaces. In the first case, we make 2 interfaces fixed and one switchable, and in the second case we make one interface fixed and two switchable. In scenario 3, we analyze the performance of the Random and the Greedy channel assignment on 802.11b using 3 non overlapping channels. We cannot use the Superimposed code based algorithm in this case since, generating small codewords is not possible. In scenario 4, we change the packet arrival rate to analyze network saturation.

## Node Model

In this section we describe the multi radio node model used for the simulations. Fig. 10, shows a node with three radios designed in the Opnet Modeler software. The node models in the modeler are implemented in the form of modules and interaction between modules(grey boxes in fig. 10 takes place with the help of statistical wires (red and blue lines in fig. 10). Each module has properties described in the network stack. For example the 'ip' module contains all the network layer functionality, and handles all packet routing based on ip address besides other things.

To efficiently handle all the channel assignment protocols described in this thesis, the implementation has been done across ip, arp and wlan modules. Most of the channel assignment code resides in the dsr_rte process model, which is a child process of the ip module and can be invoked from the ip module code. We will describe the role of each module using fig. 11. As shown in the figure, each module is labelled as A, B and C, which denote the various functions performed by the ip, arp and wireless_mac modules.

Most of the channel assignment protocol is implemented in the ip module labelled as A in fig. 11. The ip module selects a channel based on the protocol being used, and also designates the fixed and switchable interfaces. Once a channel is selected

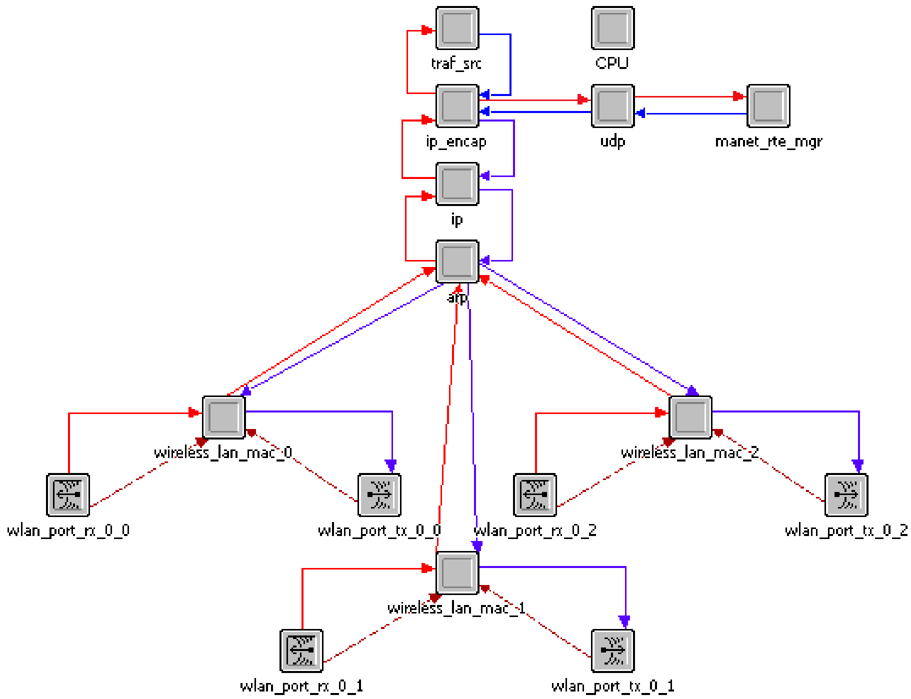


Figure 10. Multi-Radio Wireless Node Model.

in the ip module this information is passed to the arp module, which will then set the switch the appropriate wireless interface to the fixed channel. Once, the interface is assigned to the fixed channel, A HELLO packet is created for broadcast over the neighborhood. A multi hop broadcast can be created by setting the hop count to the required hops. For example for a two hop broadcast the hop count will be set to 2 and so on. The data structure for the node such as the Neighbor Table and the Channel List are also handled in this module. The data packets are also generated at this module. The size of a data packet if fixed to 1024 bits which is used for unicast

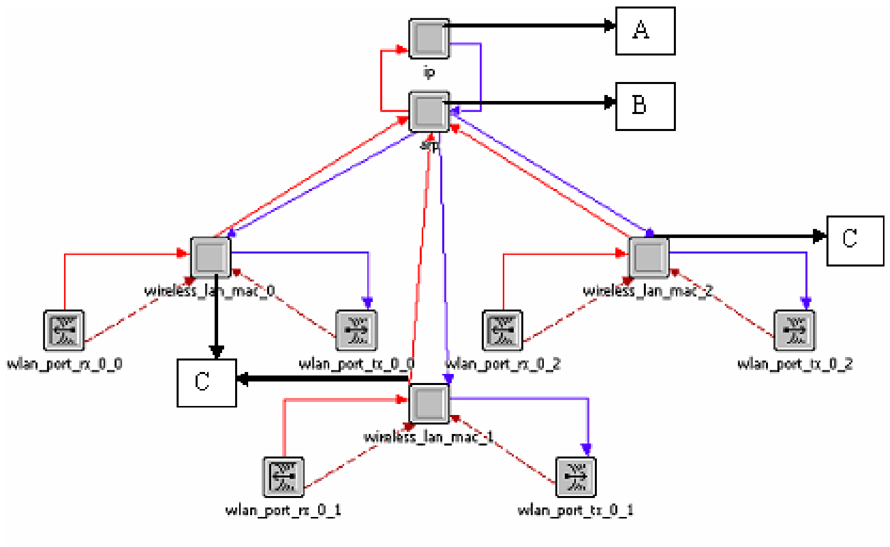


Figure 11. Internal Workings of the Node Model.

data communications. The following are the brief list of functions performed at the ip module,

- Select a channel based on the protocol being used.
- Select the fixed interface(s).
- Send the fixed channel and interface(s) information to the arp module.
- Initialize the required data structures.
- Create the HELLO packet and broadcast.
- Generate data packets for unicast communication.

The role of the arp module is to make sure that the packets coming from the ip module are sent to the correct wireless interface. When a packet arrives at this module, it checks for the interface number and sends the packet to the correct interface. These are the functions performed at the arp module,

- Check for the interface number and send information to the correct interface.
- If packet(s) arrive from the wireless_mac module(s), send it to the ip module.

When a packet arrives from the arp module to the wireless_mac module, a check is made to see if the interface is the fixed interface. When a outgoing packet, such as the HELLO packet arrives at this module, interface switches to the specified channel and then sends out the packet. These are the functions performed at the wireless_mac module,

- Check if it is designated as the fixed interface and set the channel.
- If it s not the fixed interface, switch channels accordingly and send packets.

## Simulation Settings

All the scenarios used in this thesis are designed in Opnet Modeler to make use of its rich set of features and libraries that it offers. The Modeler provides several options such as choosing the type and area of geographical area, node placements in the area, etc. An example of a scenario with 20 nodes created in the Modeler is shown in fig 12. We have created several scenarios with varying network sizes, so that we can simulate sparse and dense networks.

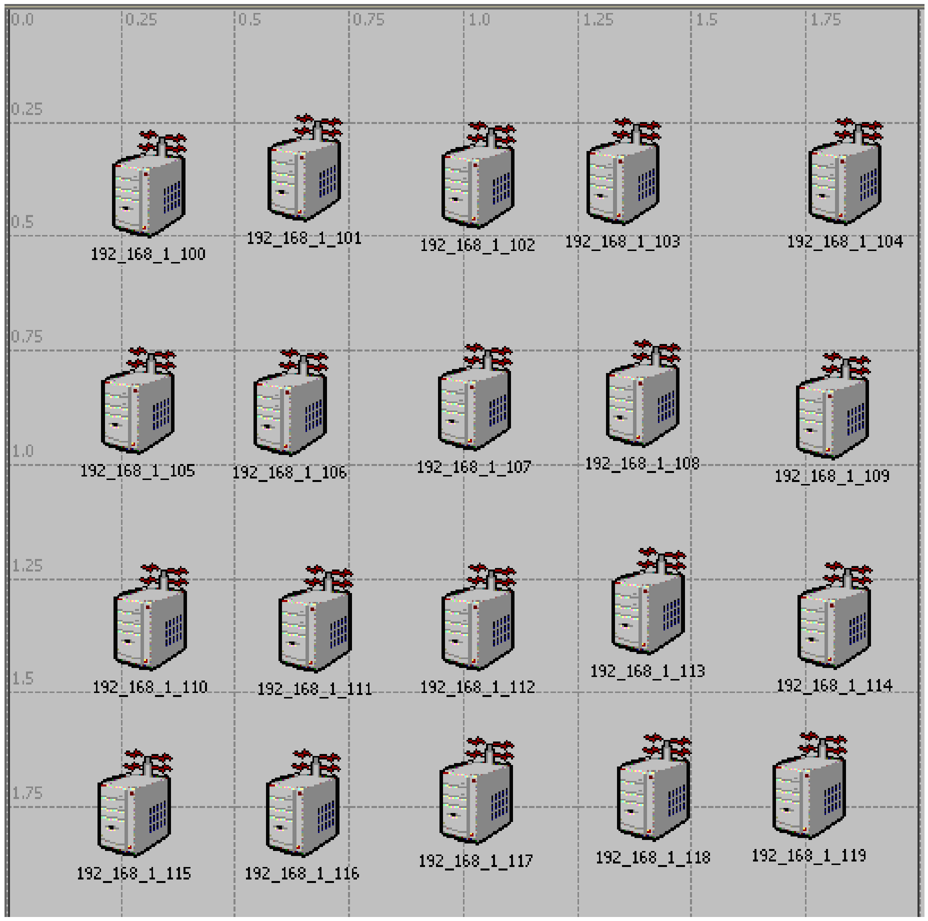


Figure 12. Simulation Scenario Setup for 20 nodes.

The Modeler provides a set of attributes associated with a node model. These attributes include assigning a channel to the nodes radio interface, setting the power level, setting the size of the packets, etc. However, we do not make use of these menu options in the Modeler, since most of the settings for the nodes in the scenario networks are controlled from the procedures inside the modules described above. For example setting the channel on a radio, the size of a packet, etc are all handled in one of the modules using the procedures described in chapter 2.

## Simulation Results and Analysis

In this section we analyze the simulation results and evaluate the performance of the three protocols. The results were averaged over multiple simulation runs. We will now give some definitions of the metrics which we use to evaluate the performance.

- *Overhead*: Overhead is obtained by the formula

$$Overhead = \frac{Total\ HELLO\ packets}{Total\ Data\ Packets} \times 100 \tag{3.1}$$

- *Throughput*: Throughout is defined as the total number of unicast data communication bits.

- *Delay*: The time difference between a packet sent from the source node to the destination node.

  $Delay = Packet_{Time_{Sent}} - Packet_{Time_{Received}}$

- *Node Degree*: Node degree is defined as the average number of nodes on a given channel(s). A simple example of this will be the number of nodes operating on one channel, or two channels. This can be obtained by printing nodes Channel List and then averaging them. The pseudocode given in algorithm 5 can be used to obtain the node degree.

The parameters for scenario 1 are give in table 2. As, shown in the table, for this scenario, the number if interfaces for each node is set to two. The network size is varied from 13 nodes up to 120 nodes with 13 available channels .

**Algorithm 5** Node Degree

$node = \emptyset$
**for** $i = 0$ *to* $C$ **do**
  **for all** $j \in N$ **do**
    $node = node + Channel\_List(i)$
    **if** $j = N$ **then**
      $print(avg\_node = node/N)$
    **end if**
  **end for**
**end for**

Table 2. Scenario 1 Settings.

| | |
|---|---|
| Network Interface Cards (k) | 2 |
| Packet Interarrival Time (r) | 2 sec |
| Transmission Range (R) | 250m |
| Network Size (N) | 13 to 120 nodes |
| Channels (C) | 13 |

We used the metrics described above to calculate the throughput, delay, overhead and the node degree for this scenario. Figs. 13 and 14 show the network throughput and the average delay for the network. The Superimposed code (SC) based channel assignment performs better than the Greedy and the Random channel assignment protocols. However, the tradeoff is that the overhead is higher for SC protocol as shown in fig 15. This is due to the fact that the number of HELLO packet exchanges that take place before a unicast data communication begins is very high in the SC protocol.

The choice of channel assignment protocol to be used in a network depends on a lot of factors such as the network topology, robustness of the nodes in the network,

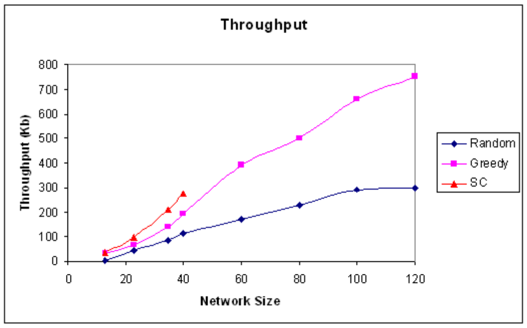


Figure 13. Scenario 1: Throughput.

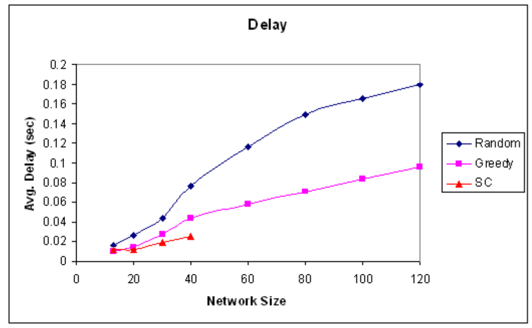


Figure 14. Scenario 1: Delay.

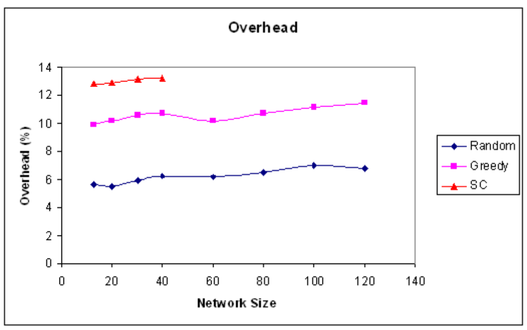


Figure 15. Scenario 1: Overhead.

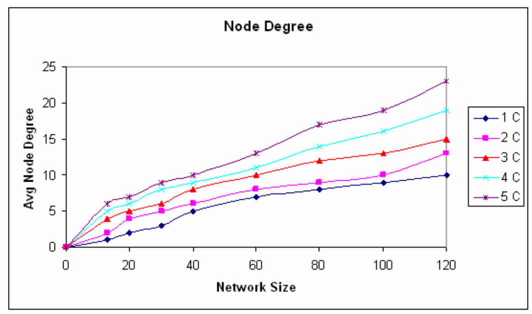


Figure 16. Scenario 1: Node Degree.

etc. For example, in a community mesh network, the SC protocol is a good choice because it will yield the maximum throughput compared to the other two protocols. If we assume that all the nodes are stationary and are constantly supplied with power, then probability of a node failure due to loss of power is very low. Therefore, for such a network the SC protocol will be a very good choice. Also, since then the nodes are stationary, number of HELLO packet exchanges can be very low and can be used over long time intervals to check if the network topology has not changed.

If we consider a sensor network deployed in an open area, where every node is powered by a battery, then the most of the nodes power must be utilized in data packet exchange. In such a network the probability of a node failure due to loss of battery power is very high. In this case, there will be a trade off between HELLO packet exchanges that are required and the data transmissions. If the goal of the sensor network is to constantly provide some information then the data transmission should be higher, so the SC protocol would be an ideal choice for such situations. On the other hand, if network requires data transmission occasionally then the Greedy protocol is better suited, since the HELLO packet exchanges can be used to check if there is any change in the network topology, so that when a node is ready for data transmission it do so without having to search for its neighbors.

Random channel assignment is effective when the network size is very small. Ideally, this protocol would work efficiently when the number of nodes in the network are smaller than the number of available channels. For example, if we consider a network designed for a large room, in which a small number of nodes are placed in different parts of the room. In such cases, the Random channel protocol may be efficient since the number of nodes in the network is very small.

In scenario 2, we changed the number of radio interfaces on a node to 3. For a node equipped with three radios there are two possibilities. In the first case one radio is assigned to the fixed channel and the remaining two radio are switchable. This is indicated in the figures as '2+1'. In the second case two radios are fixed and one is switchable which is indicated as '1 + 2'. As shown in fig. 17, the throughput for in both cases is almost same.

Table 3. Scenario 2 Settings.

| | |
|---|---|
| Network Interface Cards (k) | 3 |
| Packet Interarrival Time (r) | 2 sec |
| Transmission Range (R) | 250m |
| Network Size (N) | 20 to 120 nodes |
| Channels (C) | 13 |

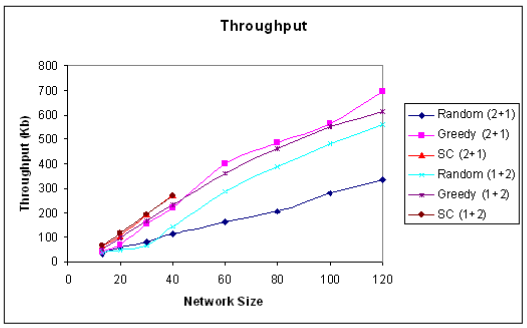


Figure 17. Scenario 2: Throughput.

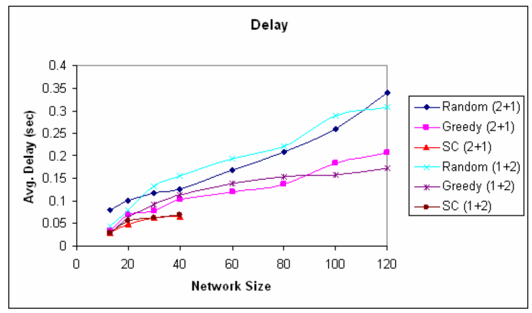


Figure 18. Scenario 2: Delay.

If the packet inter arrival time is shorter, then the time difference between packets arriving at the destination node is very small. Therefore in this case there can be a possibility that two packets may overlap in time and therefore the destination node may not be able to read them correctly and therefore will request for a retransmission, which will cause a drop in the network performance. If the packet inter arrival time is larger, then packets will not overlap in time, but it will also mean that the destination node receives packets at a much slower rate which will also cause a drop in network throughput.

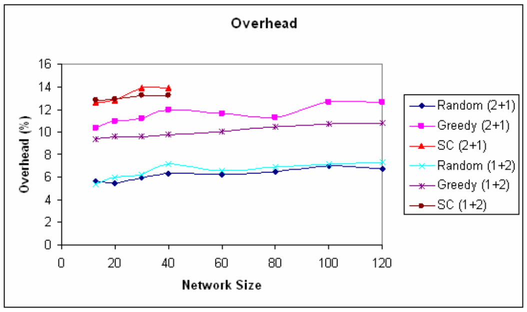


Figure 19. Scenario 2: Overhead.

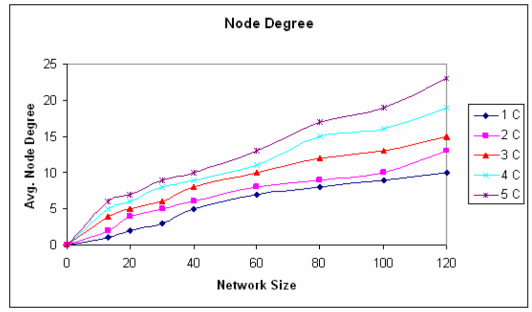


Figure 20. Scenario 2: Node Degree.

Table 4. Scenario 3 Settings.

| | |
|---|---|
| Network Interface Cards (k) | 2 |
| Packet Interarrival Time (r) | 2 sec |
| Transmission Range (R) | 250m |
| Network Size (N) | 20 to 120 nodes |
| Channels (C) | 3 |

The overhead for the network is highest for the Superimposed code based channel assignment as shown in fig 19, due the large number of HELLO packet exchanges that are required to initiate a unicast data communication.

For scenario 3, we consider that nodes are operating on three non overlapping channels. We only analyze the performance of the Random and Greedy protocols in this scenario, since it is not possible to generate superimposed code matrix for such a small channel set operating on large networks.

As can be seen in fig 21, the throughput drops as the network size increases. This can be attributed to the fact that since there are only three channels to select from, once the number of nodes increases, the number of nodes operating on the

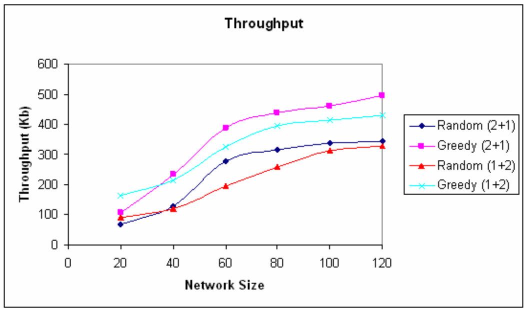


Figure 21. Scenario 3: Throughput.

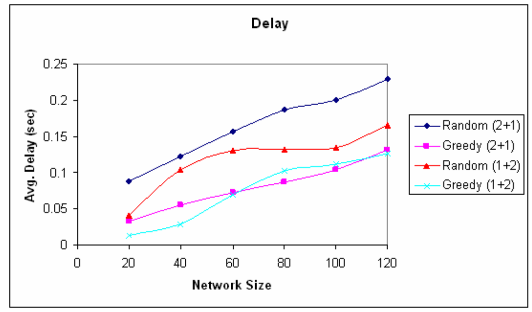


Figure 22. Scenario 3: Delay.

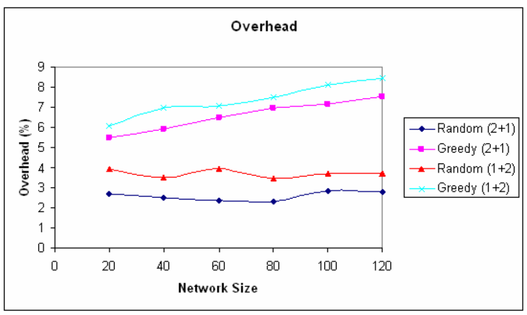


Figure 23. Scenario 3: Overhead.

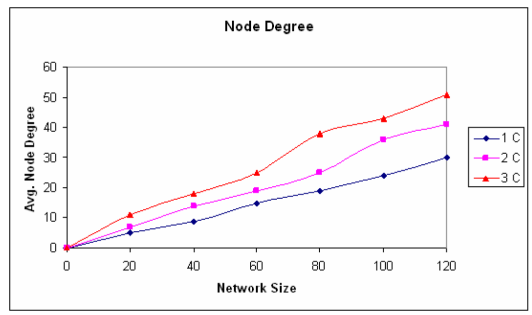


Figure 24. Scenario 3: Node Degree.

same channel will increase. This will result in a lot of packet collisions and therefore requires lot of retransmissions and thus will result in more delay.

Such a case where the number of channels is low, is ideal for a smaller network size. An example of such a network is a home network, where the mesh node acts as an access point and there are few devices which connect with an outside network such as the internet using the mesh node. It can be noticed that the throughput drops even for the Greedy protocol in this scenario and the network can experience large delays due to node interference.

Table 5. Scenario 4 Settings.

| | |
|---|---|
| Network Interface Cards (k) | 2 |
| Packet Interarrival Time (r) | 2 sec |
| Transmission Range (R) | 250m |
| Network Size (N) | 20 to 120 nodes |
| Channels (C) | 13 |

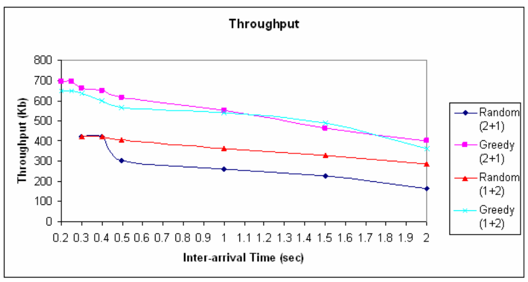


Figure 25. Scenario 4: Throughput.

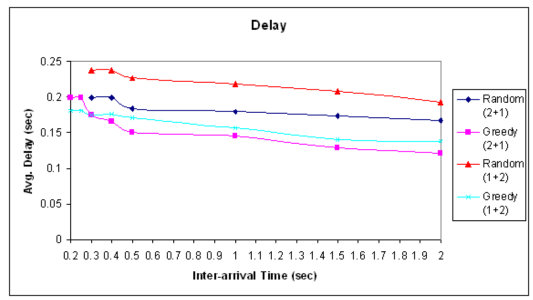


Figure 26. Scenario 4: Delay.

One aspect to notice here is that the average number of nodes that are operating on a channel is very large as indicated in fig. 24. This is due to the fact that the number of channels on which nodes can operate is very less. Hence, as mentioned above, such a scenario is ideal for smaller sized networks.

In scenario 4, we keep the network size fixed to sixty nodes and vary the packet inter arrival time. Lowering the packet inter arrival time at a node will result in faster packet generation. This will give indicate how a network can handle faster data requests without saturating.

Network saturation is also one of the reasons which will degrade the performance of a network. Hence, when a network is designed it is very important that the

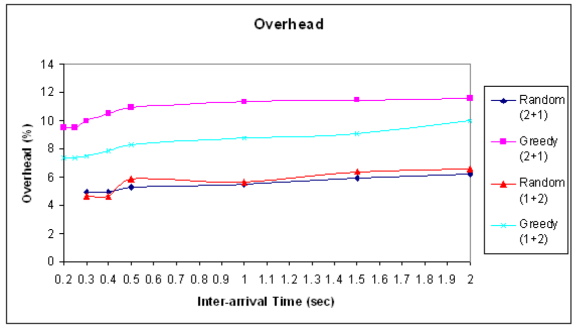


Figure 27. Scenario 4: Overhead.

saturation limit for a network is established so that precautions can be taken to make sure a network never saturates. For example consider a simple network having 10 nodes. If one of the nodes is communicating with an external network, and it generates packets at a faster rate, then most of the network bandwidth is consumed by that node. It is important to note that this packet generation rate is mostly user initiated. For example, if in a file sharing application, there are multiple files being downloaded from the network then the number packets generated is higher. Such a situation will lead to network saturation which will degrade the performance. This is one of reasons for limiting the amount of bandwidth available to the users if a network. Thus lower packet inter arrival time will degrade the network performance even in smaller networks.

# CONCLUSIONS

This thesis has introduced two different approaches to solve the channel assignment problem in a MR-MC network that maximizes the network throughput. We have presented the trade offs when different approaches are used in different cases.

The channel assignment problem based on a Greedy approach is appropriate for networks where power is an important factor, such as a sensor network. There is a however an interesting trade off in the frequency of HELLO packet exchanges which depends on the function of the network. If data transmission is more important then the frequency of the HELLO packet exchanges can be decreased. If data transmission is less frequent then the frequency of the HELLO packet exchanges can be increased so that a node is aware of any changes in the network topology.

The Superimposed code based channel assignment can improve the network throughput, but at the cost of high overhead. This approach is efficient in networks where the data communication will take place for long time periods. Once, two nodes are involved in a data transmission then there is no further need for HELLO packet exchanges and therefore result in a much lower overhead.

However, extending the superimposed code based protocol to large networks is not possible due to the fact that there is no effective way to generate unique codewords for all the nodes in a large networks. If such a method to generate codewords is available, it will be interesting to see how the codeword based channel assignment will perform on large networks.

# BIBLIOGRAPHY

[1] Andrew S Tanenbaum, *Computer Networks*, Prentice Hall 2000

[2] James F. Kurose, Keith W. Ross , *Computer Networking*, Pearson Education 2004

[3] Mischa Schwartz, *Mobile Wireless Communications*, Cambridge University Press 2005

[4] Pradeep Kyasanur, Nitin H. Vaidya , *Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks*, SIGMOBILE Mobile Computing and Comm. Rev., vol. 10, no. 1, pp. 31 -43, 2006.

[5] Kai Xing, Xiuzhen Cheng, Liran Ma, Qilian Liang, *Superimposed Code Based Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks*, MobiCom 2007

[6] Shin-Ming Cheng, Phone Lin, Di-Wei Huang, Shun-Ren Yang, *A Study on Distributed/Centralized Scheduling for Wireless Mesh Network*, WCMC 2006

[7] Richard Draves, Jitendra Padhye, Brian Zill, *Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks*, MobiCom 2004

[8] Haitao Wu, Fan Yang, Kun Tan, Jie Chen, Qian Zhang, Zhensheng Zhang , *Distributed Channel Assignment and Routing in Multiradio Multichannel Multihop Wireless Networks* , IEEE 2006

[9] Bong-Jun Ko, Vishal Misra, Jitendra Padhye, Dan Rubenstein *Distributed Channel Assignment in Multi-Radio 802.11 Mesh Networks*, available via WWW at url: *http://research.microsoft.com/ padhye/publications/wcnc2007.pdf*

[10] The Institute of Electrical and Electronics Engineers, Inc. IEEE Std 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications*, The Institute of Electrical and Electronics Engineers, Inc. 1997.

[11] W. H. KAUTZ, R. C. SINGLETON , *Nonrandom Binary Superimposed Codes*, IEEE 1963

[12] Opnet Technologies , *www.opnet.com*

[13] 802.11 Drafts available via WWW at url: *http://www.ieee802.org/11/*

[14] Allen Miu, Hari Balakrishnan, Can Emre Koksal, *Multi-Radio Diversity in Wireless Networks*, MIT Computer Science and Artificial Intelligence Laboratory

[15] Anand Prabhu Subramanian, Milind M. Buddhikot, Scott Miller *Interference Aware Routing in Multi-Radio Wireless Mesh Networks*

[16] M. Alicherry, R. Bhatia, and L. E. Li, *Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks*, MobiCom 05

[17] A. P. Subramanian, R. Krishnan, S. R. Das, and H. Gupta, *Minimum-interference channel assignment in multi-radio wireless mesh networks.*

[18] Krishna N. Ramachandran, Elizabeth M. Belding, Kevin C. Almeroth, Milind M. Buddhikot, *Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks*, Infocom 06

[19] Strix Systems , *Wireless Network in Blankenberge, Belgium*, Press Release accessed via WWW at: *http://www.strixsystems.com/press/blankenberge.asp*

[20] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng and Jang-Ping Sheu, *A New Multi Channel MAC Protocol with On Demand Channel Assignment for Mobile Ad Hoc Networks*, ISPAN 2000

[21] Jungmin So and Nitin H. Vaidya, *MultiChannel MAC for Ad Hoc Networks: Handling MultiChannel Hidden Terminals Using A Single Transceiver*, Mobihoc 2004

[22] Ian F. Akyildiz, Xudong Wang and Weilin Wang, *Wireless mesh networks: a survey*, available via WWW at *http://www.comet.columbia.edu/ campbell/e6951/papers/meshsurvey2005.pdf*