Fast Channel Allocation for multimedia communication in CRAHNs under Byzantine Attack

Thesis report submitted to

Indian Institute of Technology (ISM), Dhanbad

in partial fulfilment for the award of the degree of

Master of Technology

in

Computer Science & Engineering

by

Aadity Aabha Patel

(17MT001980)

Under the supervision of

Dr. Ansuman Bhattacharya

Department of Computer Science & Engineering

Indian Institute of Technology (ISM), Dhanbad

Dhanbad - 826004, India

Year, 2018-19

Certificate

This is to certify that Miss Aadity Aabha Patel (Admission No. 17MT001980), a student of M.Tech. (Computer Science and Engineering), Department of Computer Science, Indian Institute of Technology (Indian School of Mines), Dhanbad has worked under my guidance and completed her Dissertation entitled **Fast Channel Allocation for multimedia communication in CRAHNs under Byzantine Attack** in partial fulfillment of the requirement for award of degree of M.Tech. in Department of Computer Science and Engineering from Indian Institute of Technology (Indian School of Mines), Dhanbad.

This work has not been submitted for any other degree, award, or distinction elsewhere to the best of my knowledge and belief. She is solely responsible for the technical data and information provided in this work.

(Prof. Ansuman Bhattacharya)
Assistant Professor and Guide
Department of CSE
IIT (ISM), Dhanbad

FORWARDED BY:

(Haider Banka)
Head of the Department
Department of CSE
IIT (ISM), Dhanbad

DECLARATION

The Dissertation titled **"Fast Channel Allocation for multimedia communication in CRAHNs under Byzantine Attack"** is a presentation of my original research work and is not copied or reproduced or imitated from any other person published or unpublished work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions, as may be applicable. Every effort is made to give proper citation to the published/unpublished work of others, if it is referred to in the Dissertation.

To eliminate the scope of academic misconduct and plagiarism, I declare that I have read and understood the UGC (Promotion of Academic Integrity and Prevention of Plagiarism in Higher Educational Institutions) Regulations, 2018. These Regulations have been notified in the Official Gazette of India on 31 st July, 2018. I confirm that this Dissertation has been checked with the online plagiarism detector tool Turnitin (http://www.turnitin.com) provided by IIT (ISM) Dhanbad and a copy of the summary report/report, showing Similarities in content and its potential source (if any), generated online through Turnitin is enclosed at the end of the Dissertation. I hereby declare that the Dissertation shows less than 10% similarity as per the report generated by Turnitin and meets the standards as per MHRD/UGC Regulations and rules of the Institute regarding plagiarism.

I further state that no part of the Dissertation and its data will be published without the consent of my guide. I also confirm that this Dissertation work, carried out under the guidance of **Dr. Ansuman Bhattacharya**, Department of Computer Science and Engineering, IIT (ISM) Dhanbad, has not been previously submitted for assessment for the purpose of award of a Degree either at IIT (ISM) Dhanbad or elsewhere to the best of my knowledge and belief.

Dr. Ansuman Bhattacharya
Assistant Professor
Department of CSE
IIT (ISM), Dhanbad

Aadity Aabha Patel
Adm. No.:**17MT002001**
M.Tech (CSE-IS)
Department of CSE
IIT (ISM), Dhanbad

Abstract

Cognitive Radio (CR) technology in wireless networks solves the issues which exist in the wireless network due to scarcely available spectrum and inefficient spectrum utilization. CR technology exploits the existing wireless spectrum opportunistically. The basic idea of the Cognitive Radio Networks (CRNs) is to allow the Unlicensed or Secondary Users to make use of the spectrum of a Primary User on the condition that the secondary user has to vacate the band once the primary user starts transmitting the packets.A rising trend of multimedia communication as text, audio, still image as well as video in various applications including *CRN* has been observed [3].Distinct bandwidths are required for various types of signals to maintain the *quality-of-service (QoS)* during multimedia communication over a *CRN*. For multimedia communication, therefore, the bandwidth necessity of the channel varies. Security is one of the critical attributes of any communication network. The presence of an attacker in the Cognitive Radio Ad Hoc Networks (CRAHN s) leads to disruption in the reliable CR communications. The Byzantine Attack denotes the attack during spectrum sensing which is a very crucial factor in reliable CR communications. Under such attack, proper channel allocation is to be done. Majority Voting algorithm is proposed for fast channel allocation for multimedia communication under such attacks in distributed CR networks.

Acknowledgements

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Ansuman Bhattacharya, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Cognitive radio ad hoc networks and multimedia channel allocation and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this report work. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I thank our H.O.D. Prof. Haider Banka and other faculties of Computer Science Department of IIT Dhanbad for their encouraging words and valuable suggestions. I am also thankful to our college for providing good facilities for us in completing our research work.

At last but not the least I am in debt to my family to support me regularly during my hard times.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Date: (Aadity Aabha Patel)
Place:

Contents

Certificate i

Declaration ii

Abstract iii

Acknowledgements iv

Contents v

List of Figures vii

List of Tables viii

Abbreviations ix

1 Introduction 1

1.1 Security Issues in Physical Layer of CRAHN . . . 3

1.1.1 Primary User Emulation . . . 3

1.1.2 Objective Function Attack . . . 4

1.1.3 Jamming . . . 4

1.2 Security Issues in Link Layer of CRAHN . . . 5

1.2.1 Spectrum Sensing Data Falsification or Byzantine Attack . . . 5

1.2.2 Control Channel Saturation DoS Attack . . . 5

1.2.3 Selfish Channel Negotiation . . . 6

1.3 Security Issues in Network Layer of CRAHN . . . 6

1.3.1 Sinkhole Attack . . . 6

1.3.2 Selective Forwarding . . . 6

1.3.3 HELLO Flood Attack . . . 7

1.3.4 Wormhole Attack . . . 7

1.3.5 Sybil Attack . . . 7

1.3.6 Black Hole Attack . . . 7

2 Literature Review 9

2.1 Byzantine Fault Model 9
2.2 Agreement Protocol 11
2.2.1 Byzantine Agreement Problem 11
2.2.2 Consensus Problem 11
2.2.3 Interactive Consistency Problem 12
2.3 Fault Tolerant Model 12
2.3.1 Commit protocols 13
2.3.2 Voting protocols 13
2.4 Multimedia channel allocation 14

3 Problem Statement and Contribution 16
3.1 Problem Statement 16
3.1.1 Channel Allocation Problem 16
3.2 Contribution 17

4 Proposed Methodology 20
4.1 Source Node 20
4.1.1 Channel Allocation for multimedia data 21
4.1.2 Data Transmission 25
4.2 Destination Node 26
4.3 Non Faulty Node 27
4.4 Faulty Node 29

5 Result analysis 31
5.1 Distribution of Malicious Nodes in Single Hop 33
5.1.1 Method I 33
5.1.2 Method II 33
5.1.3 Method III 34

6 Conclusion 41

Bibliography 42

List of Figures

2.1 Network with four node 10

3.1 Four node network 17
3.2 State Diagram 18

4.1 RREQ message format. 21
4.2 RREP message format. 21
4.3 CheckOneHop message format. 22
4.4 FreeOneHop message format. 24
4.5 BusyOneHop message format. 24
4.6 SrcFree message format. 24
4.7 Data Packet format. 25
4.8 Channel Deallocation message format. 26
4.9 DestFree message format. 26
4.10 DestBusy message format. 26
4.11 SUCCESS message format. 27
4.12 DATA_ACK message format. 27
4.13 DATA_NACK message format. 27

List of Tables

5.1 Assumptions 32
5.2 Majority Voting 35
5.3 Existing Method 36
5.4 Majority Voting 37
5.5 Existing Method 38
5.6 Majority Voting 39
5.7 Existing Method 40

Abbreviations

QoS	**Q**uality of **S**ervice
CRAHNs	**C**ognitive **R**adio **A**d **H**oc **N**etworks
PU	**P**rimary **U**ser
SU	**S**econdary **U**ser

Chapter 1

Introduction

Wireless applications and services are allocated with certain spectrum bandwidth by governmental agencies as per the *Fixed Spectrum Access (FSA)* policy. According to it, *Primary Users (PUs)* or licensed users and *Secondary Users (SUs)* or unlicensed users utilizes and exploits the spectrum bands assigned to them. The unlicensed users operate on the unlicensed bands, such as *Industrial, Scientific, and Medical (ISM)* bands. Because of the rapid rise in the usage of wireless applications and cell phones over the recent years, the unlicensed bands are getting congested. According to the recent studies conducted by *Federal Communications Commission (FCC)*, the allocated spectrum to the licensed users through the *FSA* policy was found to be mostly underutilized. The average utilization of the licensed spectrum bands was found to be varying between 15% to 85%. Thus, *Dynamic Spectrum Access (DSA)* techniques brought forward to exploit the licensed spectrum by the unlicensed users in an efficient and non-interfering manner [12][13]. *Cognitive Radio (CR)* is a technology that facilitates the utilization of radio spectrum efficiently by changing its radio parameters in correspondence to its environment. *CR* senses the variation in its environment and dynamically re-configures its radio parameters accordingly.

The rise in demand for high-quality multimedia services has been leading to the technological evolution of high bandwidth wireless/mobile communications systems

and standards. Thus a rising trend of multimedia communication as text, audio, still image as well as video in various applications including *CRN* has been observed [3]. Distinct bandwidths are required for various types of signals to maintain the *quality-of-service (QoS)* during multimedia communication over a *CRN*. For multimedia communication, therefore, the bandwidth necessity of the channel varies.

Suppose, the minimum bandwidth requirement be B_{min} for all these channels. Say B_{min} is equal to 64 kbps for a voice signal. Assuming that the complete range of the spectrum is partitioned into units of B_{min} and then allocating channels, each of width B_{min} as much needed by the different multimedia signals. As a result, we are supposed to require 128 kbps bandwidth for data, 256 kbps bandwidth for still image, 384 kbps bandwidth for video, and 512 kbps bandwidth for online streaming data. Hence, allocating 1 channel to data, 2 channels to still image, 4 channels to voice, 6 channels to video and 8 channels online streaming data[12]. Where each channel width is B_{min}.

Basic idea of *CR* networks is to overcome the spectrum scarcity problem by utilizing the temporarily unused spectrum or Spectrum Holes in a licensed band. The unlicensed devices operate in the analyzed unused spectrum in the licensed bands and vacate the band whenever a licensed device arrives.

Cognitive Radio Networks (CRNs) are organized in different ways based on its architecture [2]. The *Infrastructure based CRNs* contains the *Base Stations (BSs)* or *Access Points (APs)* as a central network entity, through which the communication between the *CR* devices occur. All the decisions of the network are taken by the central control entity. On the other hand, with the help of ad hoc connections in *Cognitive Radio Ad Hoc Networks (CRAHNs)* the *CR* users can communicate with each other. In an ad hoc model, each *CR* users must have all the cognitive capabilities. Two component parts of the *cognitive radio ad hoc network (CRAHN)* architectures seem to be the primary network and the *CR* network.

A cognitive cycle is formed through various spectrum management functionalities of the *CRAHNs* firstly includes *Spectrum Sensing* then *Spectrum Decision* after this *Spectrum Sharing* and finally *Spectrum Mobility* [9][1][11]. In *CRAHNs*, the nodes only have the local observations and therefore, to form a decision, cooperation schemes are required for exchanging observed information among the nodes. *Cooperative Spectrum Sensing (CSS)* is a cooperation scheme where the final decision on the availability of spectrum holes is made by considering the spectrum sensing information of all the *CR* nodes in the network.

CR networks are prone to many security issues. The typical working nature of *CRN* makes it vulnerable to many new kinds of security attacks. These attacks can affect the spectrum management functionalities in the *CRN*.

Security issues can be categorized in *CRN* based on the targeted layers of the protocol stack.

1.1 Security Issues in Physical Layer of CRAHN

1.1.1 Primary User Emulation

During the sensing process, if a *SU* detects another *SU* node in the network, certain mechanisms are implemented to share the particular channel among the users. In *Primary User Emulation (PUE)* attack, a *SU* node behaves as *PU* in the network and may try to obtain all the resources of the channel [5]. The nodes act maliciously in order to acquire full-spectrum bands without intending to share it. During sensing, the malicious nodes may transmit signals and the received signal strength at other sensing nodes will be beyond a certain threshold. It makes other nodes believe it is a primary user in this way. Aim of the attacker is to acquire a maximum share of the resources or to harm other *SUs* by blocking them from using the spectrum

opportunities. It can be categorized as *Malicious PUE attack* and *Selfish PUE attack*.

1.1.2 Objective Function Attack

The objective function attack aims the reconfiguration ability of the *CR* node [6]. For adapting the radio parameters according to the changing spectral environment, the nodes reconfigure the parameters by computing certain objective functions. During the process, an attacker may hamper the objective function by altering the radio parameters in such a way that it is configured according to the interest of the attacker and thus, the attacker gains from the alteration.

1.1.3 Jamming

It is a kind of *DoS* attack [17]. The attacker sends junk data packets which give the impression of a channel being busy. It can transmit maliciously in the licensed band making it unable to use by both *PU* and a *CR* user. It can also send the packets to the *CR* nodes continuously to hinder the communication. The attacker can send packets to the *CCC* and disrupt its functioning. *Physical* and *MAC* layer both are prone to jamming. Four types of jammers are there. They are *Constant Jammer, Deceptive Jammer, Random Jammer* and *Reactive Jammer*.

1.2 Security Issues in Link Layer of CRAHN

1.2.1 Spectrum Sensing Data Falsification or Byzantine Attack

Spectrum sensing is important for determining the spectrum access opportunities in the bands. Thus, the spectrum sensing attack can cause an immense effect on the successful implementation of *CRN*. Spectrum sensing should be done in an efficient way. The collaboration of sensing information from different *CR* users needs to be done for the accuracy of the sensing information. The detection performance is improved with the help of The *Cooperative Spectrum Sensing (CSS)*. The *CSS* process implementation depends on the type of *CR* architecture.In such networks, several types of attacks can be introduced, one of them being *Byzantine Attacks* or *Spectrum Sensing Data Falsification (SSDF)* [10]. Here, a malicious *CR* user participating in the network provides with false channel sensing information thus creating disruption in the decision making. In distributed *CRN*, the false information propagates fast thus this attack is more damaging in such networks.

1.2.2 Control Channel Saturation DoS Attack

The attack aims the *CCC* by saturating it with junk *MAC* control frames during the decision making for the appropriate channel for data transmission [4]. Interchanging the control frames between the nodes so that the selected channels can be reserved by the node for transmission. As the *CCC* accommodates only limited *CR* users, the attacker saturates it with the control frames so that the performance of the network can be degraded.

1.2.3 Selfish Channel Negotiation

In this, the attacker node refuses to moves ahead of the packets to other nodes for conserving its energy and degrading the throughput of the network [4].

1.3 Security Issues in Network Layer of CRAHN

The network layer is also prone to attacks affecting the data packet routing to the destination [14]. Some of the network layer attacks are discussed below.

1.3.1 Sinkhole Attack

The attacker declares in this that it is the best path for forwarding packets to a specific destination [10]. It tempts the neighboring nodes to forward packets through it. The attacker can use a high level of power for sending packets to the destination advertising itself as one hop count away from the destination. After creating a trust base, the attacker launches attacks such as eavesdropping. By just providing a route with high-quality to the destination, this can be launched for such networks where all packets share the same destination such as in networks with only one *Base Station (BSS)*.

1.3.2 Selective Forwarding

For multi-hop networks, the relay nodes may selectively forward the packet by modifying or discarding the packets from any node in the network [10].

1.3.3 HELLO Flood Attack

The attacker sends a broadcast message to all nodes throughout a network with sufficient power to convince their neighbor[10]. Thus, the packets will be lost. Even if the nodes detect the attack, it cannot send packets to any of its neighbors as all of them will be using the same route. There are some protocols which require local information exchange among neighbor nodes so that it can maintain topology or flow control such protocols are prone to this kind of attack.

1.3.4 Wormhole Attack

In this, one part of the network receives an adversary tunnels messages over a link with small latency and replay in another part [10]. The attacker situated in the path can make other nodes to route the packets through it by colluding the distance between them. It provides with the information that it is the best node, thus creating a sinkhole.

1.3.5 Sybil Attack

A single node displays multiple profiles to other nodes throughout the network in a Sybil attack [8]. This attack is more dangerous for fault-tolerant strategies such as multi-path routing,distributed storage, dispersion, and construction of topology. It may pose a risk to protocols of geographic routing.

1.3.6 Black Hole Attack

It's a sort of [7] Denial of Service attack. A malicious node attempts to trick other nodes in such types of attack by describing itself as the shortest route to the destination.When the *RREQ* packet is received, it responds to the destination with

the highest sequence number and the lowest number of hops, tricking other nodes into believing that the node's distance is closer to the destination. Thus, the source node is tricked to send the data packets towards this node. The packets are therefore either totally lost or sent to an unknown address.

Chapter 2

Literature Review

Since *CRAHNs* are vulnerable to various kinds of security attacks, Byzantine attacks can be launched in the network which hampers the proper channel allocation. In this chapter, the fault model of the network has been discussed. The following section also includes the detailed description of different agreement protocols, fault tolerant model, the system model of our proposed protocol.

In addition to all these, this section includes a brief overview of channel allocation for multimedia communication.

2.1 Byzantine Fault Model

The *CR* ad-hoc network model is assumed to be having reliable communication medium.However, it is assumed that nodes throughout the network are prone to failure. If there is a malicious node in the network, it can create problems in the node communication *CR*. A node can fail in three modes: *Crash fault, Omission fault* and *Malicious fault* [16]. A node stops working in a *Crash fault* and it fails to resume. In a *Omission fault*, a node may or may not send messages to other nodes. For example, in the event of a broadcast, some endpoints in the network may not

receive the broadcast message out of a source node. In case of a *Malicious fault* or *Byzantine fault*, a node behaves randomly. A node may send fictitious messages to other nodes to confuse them. In such cases, a non-faulty node may never receive the correct message from a faulty node.

In this work, the Byzantine fault model with Authenticated Message System is considered. It is assumed that a malicious node (Byzantine Faulty Node) can modify its own sensing result but it cannot modify or alter the sensing result of other nodes in the network. It can provide any random result regarding channel sensing.

The range of sensing and transmission in each node is assumed to be the same. Node B and node D are the one-hop neighbor of node A, and node C is the one-hop neighbor of node B, according to Figure 2.1. But node C is and is hidden from node A's two-hop neighbor. Therefore, node A cannot sense the channel being used for transmission by node C. Through cooperative sensing, node A can query about the channel availability in its two-hop range by asking node B to sense in its one hop.

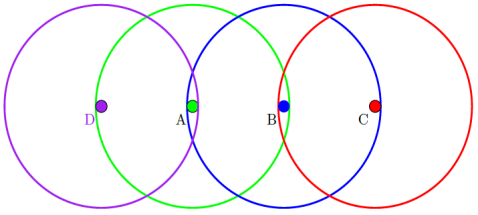

FIGURE 2.1: Network with four node

Node B replies with either 0 or 1 according to the information it gained from its own local observation. Here, 0 denotes that the queried channel is free and 1 denotes that the channel is busy and is being used by some other nodes in its one hop. The sensing result given by node B depends on the channel being used for transmission by node C. If node C is transmitting on the same channel as node A, it can lead to Hidden Node Problem and can cause a collision at B.

2.2 Agreement Protocol

Three agreement protocol problem is defined namely: *Byzantine Agreement Problem, Consensus Problem* and *Interactive Consistency Problem* [16].

2.2.1 Byzantine Agreement Problem

The Byzantine Agreement problem chooses an arbitrary source node which broadcasts the sensing information to all other nodes.
The following conditions should be met by a solution:

- **Agreement:** The same value is agreed on by all non-faulty nodes.
- **Validity:** If the source node is non-faulty, then all non-faulty node's common agreed value should be the source's initial value.

2.2.2 Consensus Problem

In the Consensus problem, every node broadcasts the sensing information to all other nodes.
The following conditions should be met by a solution:

- **Agreement:** The same value is agreed on by all non-faulty nodes.
- **Validity:** If the initial value of each non-default node is the same, then all non-faulty nodes will have the same common agreed value.

2.2.3 Interactive Consistency Problem

In Interactive Consistency problem, every node broadcasts the sensing information to all other nodes.
The following conditions should be met by a solution:

- **Agreement:** All non-faulty nodes agree on a common set of values.
- **Validity:** If the node i^{th} is non-faulty and its initial value is v, then the value i^{th} for all non-faulty nodes must be v.

The problem of Byzantine agreement is a special case of the problem of interactive consistency in which the original value of only one node is considered. The consensus problem can be solved using the solution of interactive consistency problem as all non-faulty node can compute the value that is to be agreed upon for taking the majority value of the common vector. The solutions to the problem of interactive consistency and consensus can, therefore, be derived from the solutions to the problem of *Byzantine Agreement*.

2.3 Fault Tolerant Model

To avoid disruptions due to failures and to improve availability, systems are designed to be fault-tolerant [16]. The fault tolerant system could be structured in two ways. A system might conceal failures or a system might show some well-defined actions of failure in the event of failure. *Commit protocols* and *voting protocols*, are widely used in the design of fault-tolerant systems. *Commit protocols* implement well-defined actions when failure occurs. *Voting protocols*, on the other hand, mask failures in a system.

2.3.1 Commit protocols

In distributed database systems, a transaction must be processed at every site or at none of the sites to maintain the integrity of the database. This is referred to as global atomicity. The protocols that enforce global atomicity are referred to as commit protocols. Given that each site has a recovery strategy at the local level,commit protocols ensure that the transaction is aborted unanimously by all sites, even within the existence of multiple and repeated failures.

The commit protocol is classified into two categories:

(a) Blocking or Two-Phase Commit Protocol and

(b) Non-blocking Commit Protocol

2.3.2 Voting protocols

The voting mechanism is a technique used to manage replicated data. Each replica is assigned a number of votes with the voting mechanism, and Before replication can be accessed, a majority number of votes will be collected out of a processThe mechanism for voting seems to be more tolerant than those of a commit protocol because it allows data access under network partitions, website failures, and message losses without sacrificing data integrity.

The voting mechanism is classified into two categories:

(a) Static Voting

(b) Dynamic Voting

(a) **Static Voting**

Basic Idea: A certain number of votes are designated to each replica. This information is kept in the storage table. A read or write operation is allowed if the

requesting process collects some defined number of votes, write quorum or read quorum, respectively.

(**b**) **Dynamic voting protocols**

Dynamic voting protocols solve the problem of complete unavailability of the system by adjusting the number of votes or sites that may constitute a quorum, to the system's changing state due to site failures and communication failures.
From the previously proposed dynamic protocols, two approaches to enhance availability can be identified.
(**1**) **Majority based approach -** A set of sites that can make up a majority to allow access to replicated changes in data with changing system status.
(**2**) **Dynamic vote reassignment -** The percentage of the vote allocated to the site dynamically keeps changing.

2.4 Multimedia channel allocation

In the [14], the work concentrates on a single channel allocation for plain data for *CRAHNs* under *Byzantine Attack*. An introduction to security issues of *CRAHNs* and how the malicious nodes can propagate false sensing result to its neighbors which leads them to take wrong decisions about the channel availability during the CSS in *CR* ad hoc networks. *Majority Voting* algorithm is considered there which leads to fast channel allocation for plain data as compared to the existing method.

To allocate channels to a particular user of the *CRN* necessitating some kind of multimedia interaction, it is rather feasible that the whole spectrum is sufficiently divided so there will be many small unused portions (white spaces) neither of which will individually provide the desired bandwidth to a channel. The channel allocation method using multiplexing-*frequency division multiple access (FDM-FDMA)* provides a means of communication even in circumstances where even the white spaces

in the spectrum will not provide sufficiently large adjacent bandwidth to maintain the QoS multimedia signals. Such a method is based on first finding a collection of non-contiguous white spaces whose cumulative width is equivalent to the desired multimedia signal bandwidth, then subdividing (shuffling) the bits from the original time domain signal, creating sub-packets with all these bits subsets and forwarding those sub-packets via the collection of channels so discovered. Using *FDM-FDMA* together with PU understanding for one hop connectivity, communication protocols among both two nodes was proposed.

Chapter 3

Problem Statement and Contribution

Multimedia channel allocation had not been done in the [14] instead only single channel allocation has been done. This chapter describes the problem statement and a brief description of our contributed work.

3.1 Problem Statement

The following section describes the channel allocation problem for multimedia data in *CRAHNs* under the *Byzantine attack*.

3.1.1 Channel Allocation Problem

Different bandwidths are needed for multimedia communication for distinct types of data to be transmitted over a *CRN*. Throughout this process, *Miss-Detection* and *False Alarm* occur.

When there is a malicious node in the network, it can mislead other nodes regarding channel availability by providing with conflicting results. For example, as depicted in Figure 3.1, if node B provides with a false sensing result to node A, it may hinder the proper channel allocation for node A. Suppose, node B gives free(0) as a reply on receiving the result, node A will transmit on the channel believing it is free which will gradually lead to *Miss-Detection*.

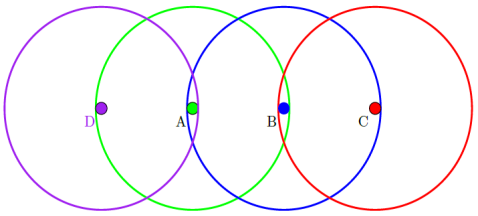

FIGURE 3.1: Four node network

On the other hand, if it replies with 1 when the actual result is 0, the result may refrain node A from transmitting on that particular channel. Thus, the outcome may lead the nodes with no channels to transmit, thus causing *False Alarm*. In such an environment, reliable channel allocation needs to be done in *CRAHNs*.

3.2 Contribution

In this work, a previously existing method for single channel allocation through majority voting have been extended for multimedia data where multiple channel allocation is done depending upon the type of data being transmitted. Suppose, the minimum bandwidth requirement be B_{min} for all these channels. Say B_{min} is equal to 64 kbps for a voice signal. Assuming the entire spectrum is divided into B*min* units and the necessary amount of channels is assigned to distinct multimedia signals for each of B*min* width. As an outcome, for data we may require 128 kbps, for still image we may require 256 kbps , for video we may require 384 kbps but also for internet streaming data we may require 512 kbps bandwidth. Therefore, we devote 1

for data, 2 for still image, 4 for voice, 6 for video and 8 channels for internet streaming data (each Bmin width). And afterward, the problem situation is compared to the problem of *Byzantine Agreement*. When the outcome is conflicting in the network, the source node must agree on a common value. The ultimate decision on the availability of the channel is made by *Majority Voting* for a node. Finally, the results of the computation and the efficiency of the algorithm for majority voting are compared to the existing algorithm.

The basic state transition diagram of the system can be depicted as in Figure 3.2. If a node wants data packets to be transmitted, it goes from Idle to Sensing state. After sensing is done within its two-hop range, it checks whether the required set of channels are available or not through Majority Voting algorithm. If most of the nodes reply that the channels are free, it Allocates the channels. If the channels are not free, it does the sensing again till a specified maximum limit.

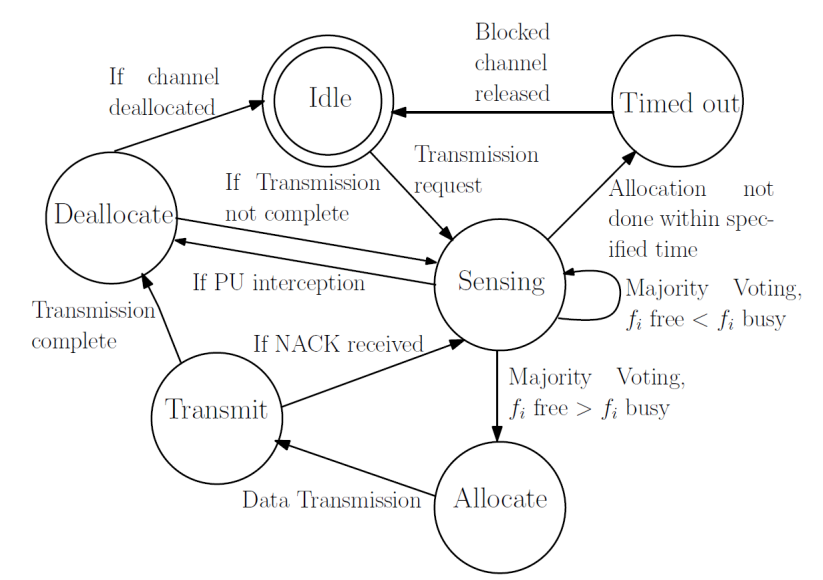

FIGURE 3.2: State Diagram

If the channel allocation is not done within a specified time or the maximum limit, firstly it went to the Timed Out state and then goes to the Idle state. Thus, the selected channels cannot be used for transmission. After the allocation of channels, it moves to the Transmit state where it transmits the data packets to the destination

through the allocated channels. Once the transmission is finished, it went to the Deallocate state and deallocates the selected channels and then goes to the Idle state. If a NACK message is received from the destination during the transmission, it goes to the sensing state again. If a PU comes in the selected channels during the transmission, it goes to the Deallocate state and does the sensing again if the transmission is not completed.

Chapter 4

Proposed Methodology

In this chapter, the protocols for fast channel allocation for multimedia data with varying bandwidth in *CR* ad hoc networks under the *Byzantine attack* has been described in detail. The channel allocation from source to destination is done using the *Majority Voting* algorithm.

4.1 Source Node

In order to find out the route from source to destination, the source node broadcasts message of the *Route REQuest (RREQ)* to neighboring nodes *CCC*. The message *RREQ* contains the information as shown in the Figure 4.1: *Source Id, Destination Id, Timestamp* and *a Tag field*. The *Source Id* and *Destination Id* specifies the Source and Destination node address, respectively. *Timestamp* denotes the global clock time. It ensures routing loop freeness and avoids the problem of count to infinity [15]. The *Tag Field* of *RREQ* will have the flag status as 0.

Source Id	Destination Id	Timestamp	0

FIGURE 4.1: RREQ message format.

After sending the *RREQ* message, it waits for Δ_{time} for the *Route REPly (RREP)* message as shown in Algorithm 1. If it is received within the specified time, the *Channel Allocation* procedure is called and the channels are allocated for source and destination node data transmission. If the *RREP* packet is not received within Δ_{time}, *R_attempt* is incremented and the *RREQ* packet is re-transmitted till a specified maximum re-transmission attempt, i.e., *MAX_R_ATTE- MPT*. If the required re-transmission attempts goes beyond *MAX_R_ATTEMPT*, the transmission between source and destination becomes unsuccessful. The *RREP* packet contains the information as shown in Figure 4.2: *Source Id, Destination Id, Timestamp* and *Tag Field*. Here, the *Tag Field* will contain the flag status 1. The *Tag field* is required to differentiate the *RREQ* and the *RREP* message.

Source Id	Destination Id	Timestamp	1

FIGURE 4.2: RREP message format.

As the *RREP* packet is transmitted from destination to source, in a uni-cast reverse path order, the *Source Id* in the *RREP* message format will denote the destination node address. The *RREP* message is transmitted to the node that transmitted the *RREQ* message. Thus, the source node gets the route information to the destination node.

4.1.1 Channel Allocation for multimedia data

In the *Channel Allocation* algorithm, the data transmission between source and destination node takes place by considering the frequencies used by all the nodes in its one hop and two hop range. If the selected set of channels are free, it is used for

Algorithm 1: Source Node

```
while (R_attempt < MAX_R_ATTEMPT) do
    Broadcast Route_Request;
    Wait for Δ_time;
    if (Route_Reply received) then
        Call Channel Allocation procedure;
    else
        R_attempt++;
    end
end
```

data transmission. The algorithm used for frequency selection in source node is as shown in Algorithm 2.

Here, a set of channels are selected randomly and there availability is checked in its one hop and two hop adjoining nodes. As a node cannot sense the nodes beyond its one hop range, the one hop neighbors are asked to sense its own one hop for the channels availability. When the message *CheckOneHop*, as shown in Figure 4.3, is sent to all the one hop neighbors through *CCC*, they check for the channels availability in their vicinity. It waits for Δ_{time} for the reply to come. Here, it should be noted that $\Delta_{time} << \delta_{time}$. Δ_{time} is the time required for message passing within one hop neighbors. Whereas, δ_{time} denotes the time taken to do the message passing within multiple hops in the network. It is necessary to find the paths from a source node to a destination node in the network. Therefore, it can vary from a single hop to hops equivalent to the network size. Thus, Δ_{time} can be significantly greater than δ_{time}.

FIGURE 4.3: CheckOneHop message format.

The output of the nodes can be either *FreeOneHop* (channels are free) or *BusyOneHop* (channels are busy) as shown in Figure 4.4 and 4.5. The message format of both the messages contains the *Source Id, frequency* f_i *such that* $f_i \epsilon S_n$ *where* S_n *denotes set of frequencies selected by source* and *Tag field*. The tag field

Algorithm 2: Channel Allocation for multimedia data

```
1  n = Required_Frequencies ( Required_Frequencies = 1 for text, 2 for audio, 4 for
     voice, 8 for video )
2  count = 0 S_n = φ
3  while (|S_n| < Required_Frequencies AND F_attempt < MAX_R_ATTEMPT) do
4      Select set of frequencies F_n randomly ( |F_n| = n);
5      For each f_i ∈ F_n do in parallel
6      Sense f_i within One Hop;
7      if (f_i==free within One Hop)) then
8          Transmit CheckOneHop to all of its one hop neighbors; /* search f_n is free
             within Two Hop */;
9          Wait for δ_time;
10         Received FreeOneHop or BusyOneHop from one hop neighbors;
11         if (FreeOneHop > BusyOneHop) then
12             Send SrcFree to destination;
13             Wait for δ_time; /*Δ_time >> δ_time*/
14             if (DestFree received) then
15                 count++;
16                 n = Required_Frequencies - count;
17                 add f_i to S_n;
18             else
19                 F_attempt++;
20             end
21         else
22             F_attempt++;
23         end
24     else
25         F_attempt++;
26     end
27 end
28 if (|S_n| = Required_Frequencies) then
29     Select F_n for data transmission;
30     Send SUCCESS to destination;
31     Call Data Transmission procedure;
32 end
```

for *FreeOneHop* message and *BusyOneHop* message will be 00 and 11, respectively. If the *FreeOneHop* message received is more than the *BusyOneHop* message, it denotes that the selected channels can be used for data transmission from source side, otherwise not. These messages are sent through *CCC*.

FIGURE 4.4: FreeOneHop message format.

FIGURE 4.5: BusyOneHop message format.

If the channels are free within source's two hop range, message *SrcFree* is sent to the destination through *CCC* and is asked to sense the channels in its two hop. The message format includes the *Source Id* and *Destination Id* with the selected Frequencies S_n as shown in Figure 4.6. It again waits for δ_{time} for the reply from destination. If the channels are free in the one hop and two hop range of destination, it sends the message *DestFree*. If the channels are busy within two hop range of either source or destination node, *F_attempt* is incremented and some other channels will be selected randomly and the whole process of channel allocation is repeated again till a maximum attempt *MAX_F_ATTEMPT*.

FIGURE 4.6: SrcFree message format.

If the channels are free in both the ends, a *SUCCESS* message will be sent to the destination. The message format of *SUCCESS* message is shown in Figure 4.11. It contains *Source Id, Destination Id, Frequencies* f_i and *Tag field*. The *Tag field* for *SUCCESS* message has flag status 01. After this, the Data Transmission procedure is called and the selected channels are used for data transmission between the source and destination.

4.1.2 Data Transmission

After the route and channel selection procedure, the data transmission between the nodes are done as shown in Algorithm 3.

Algorithm 3: Data Transmission procedure

```
while (B_tx != φ) do
    Transmit Data Packet;
    if (Within δ_time DATA_ACK received)) then
        Packet_Number++;
    else
        Sense f_i;
        if (f_i Busy) then
            Free f_i;
            Transmit Channel Deallocation message;
            Call Channel Allocation Procedure;
        end
    end
end
```

The data packets are kept in a local buffer and it waits for the transmission by the *First In First Out (FIFO)* principle. The packet structure is as shown in Figure 4.7. The packet structure contains: *Source Id, Destination Id, Packet Number* and *Data*. The data packets are sent via the selected Sn data channels to the destination. The data field size can be up to 1 KB.

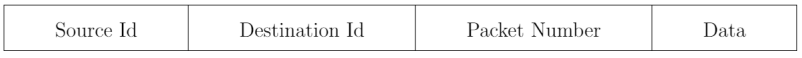

FIGURE 4.7: Data Packet format.

After sending the data packets to the destination, if an acknowledgment *DATA_ACK* is received within δ_{time}, the packet number is incremented and the next data packet is sent till the buffer is empty. If in the specified period of time the data packets are not received, the selected channels f_n are sensed again. If the channels are sensed to be busy or if its being used by some other *CR* or *PU* nodes, the channels are deallocated and a *Deallocation* message is sent to the destination. The

structure of Channel *Deallocation* message is as shown in Figure 4.8. The *Channel Deallocation* message will have 10 as its tag field. After this, for continued data transmission, reallocation of the channels are done again by calling the *Channel Allocation* procedure.

FIGURE 4.8: Channel Deallocation message format.

4.2 Destination Node

For destination node, as shown in Algorithm 4, the same procedure of majority voting is followed. First, when the *RREQ* message is received, the destination responds with the *RREP* message to the node through which the request message is received. It is forwarded till it reaches the source. The destination senses its one hop neighbors when it receives the *SrcFree* message from the source node. If the channels are not free, it sends *BUSY* message to the source. If the channels are free in its one hop, it senses its two hop neighbors by sending *CheckOneHop* message. If the channels are free in its two hop range, it sends a *DestFree* message to the source, otherwise *DestBusy*.

FIGURE 4.9: DestFree message format.

FIGURE 4.10: DestBusy message format.

When the *SUCCESS* message is received from source node, it selects the specified channels for data reception. On transmission of data packets from source, if the data packets are received properly at the destination, the destination sends the *DATA_ACK* message, otherwise, the *DATA_NACK* message is sent to the source

node. Here, the *DATA_ACK* and *DATA_NACK* messages are sent through the selected data channels. The message structure for *DATA_ACK* and *DATA_NACK* is as shown in Figures 4.12 and 4.13, respectively. The structure contains: *Source Id, Destination Id, Packet Number* and *Tag field*. The tag field differentiates the *DATA_ACK* and *DATA_NACK* message. If the *Channel Deallocation* message is received from source, it frees the selected channels.

Source Id	Destination Id	f_i	01

FIGURE 4.11: SUCCESS message format.

Source Id	Destination Id	Packet Number	1

FIGURE 4.12: DATA_ACK message format.

Source Id	Destination Id	Packet Number	0

FIGURE 4.13: DATA_NACK message format.

4.3 Non Faulty Node

In the network, apart from the source and destination node, the other non faulty nodes senses for the channel availability in its one hop and two hop range and replies with the correct sensing result. As shown in the Algorithm 5, on receiving the *RREQ* message, the non faulty nodes search its routing table for the specified destination node. If the nodes do not have any information regarding the destination node, it updates its routing table and forwards the *RREQ* to other nodes till it reaches the destination. If any of the intermediate nodes contains the information of the destination node, it sends the *RREP* message to the node from which it received the *RREQ* packet, instead of forwarding it. All the nodes in the network discards the duplicate packets received in this process. On receiving the *CheckOneHop* message from the source node, the non faulty nodes senses its one hop and replies with either *FreeOneHop* or *BusyOneHop* message correctly. On receiving the data packets, the

Algorithm 4: Destination Node

```
1  if (Route_Request obtained) then
2  |  Sending Route_Reply message to node from whom it received Route_Request
   |   message;
3  end
4  if (SrcFree obtained) then
5  |  For each f_i ∈ S_n do in parallel
6  |  Sense f_i within One Hop;
7  |  if (f_i==free within One Hop) then
8  |  |  Transmit CheckOneHop to all of its one hop neighbors;
9  |  |  Wait for δ_time;
10 |  |  Received FreeOneHop or BusyOneHop from one hop neighbors;
11 |  |  if (FreeOneHop > BusyOneHop) then
12 |  |  |  Send DestFree;
13 |  |  else
14 |  |  |  Send DestBusy;
15 |  |  end
16 |  else
17 |  |  Send BUSY ;
18 |  end
19 end
20 if (SUCCESS received) then
21 |  Select f_i for data reception;
22 end
23 if (Data Packet Received) then
24 |  Transmit DATA_ACK;
25 else
26 |  Transmit DATA_NACK;
27 end
28 if (Channel Deallocation message received) then
29 |  Free f_i;
30 end
```

nodes forward it to other nodes till it reaches the destination. If *DATA_ACK* or *DATA_NACK* message is received, it forwards the message to the specified id.

Algorithm 5: Other Non Faulty Node

```
1  if (Route_Request obtained)) then
2  |  Broadcast Route_Request;
3  end
4  if (Route_Reply obtained) then
5  |  Sending Route_Reply message to node from whom it received Route_Request
   |   message;
6  end
7  if (CheckOneHop obtained) then
8  |  Sense f_i within One Hop;
9  |  if (f_i==free) then
10 |  |  FreeOneHop;
11 |  else
12 |  |  BusyOneHop;
13 |  end
14 end
15 if (Data Packet Received) then
16 |  Forward Data Packet;
17 end
18 if (DATA_ACK Received) then
19 |  Forward DATA_ACK;
20 end
21 if (DATA_NACK Received) then
22 |  Forward DATA_NACK;
23 end
24 if (Channel Deallocation Message received) then
25 |  Forward Channel Deallocation message;
26 end
```

4.4 Faulty Node

For data transmission in a network, the problem arises when the nodes become faulty. Faulty or malicious nodes can provide with false sensing results causing confusion in making the correct decision. As the source or the destination cannot sense channels beyond its one hop, faulty nodes can provide the network with false sensing results. Here, the source and destination are assumed to be non-faulty. As shown in Algorithm 6, when the *CheckOneHop* message is received from the source, a faulty node can randomly provide with either *FreeOneHop* or *BusyOneHop* message

as a reply. For the faulty nodes in the network, *Lamport-Shostak-Pease* algorithm is considered where the number of faulty nodes in the network is within b(n - 1)/3c where n is the total number of nodes in the network.

Algorithm 6: Malicious/Faulty Node

```
1  if (Route_Request obtained)) then
2  |  Broadcast Route_Request;
3  end
4  if (Route_Reply obtained) then
5  |  Sending Route_Reply message to node from whom it received Route_Request
   |    message;
6  end
7  if (CheckOneHop obtained) then
8  |  Send either FreeOneHop or BusyOneHop randomly;
9  end
10 if (Data Packet Received) then
11 |  Forward Data Packet;
12 end
13 if (DATA_ACK Received) then
14 |  Forward DATA_ACK;
15 end
16 if (DATA_NACK Received) then
17 |  Forward DATA_NACK;
18 end
19 if (Channel Deallocation Message received) then
20 |  Forward Channel Deallocation message;
21 end
```

Chapter 5

Result analysis

This chapter includes the simulation results of the proposed protocols and its comparisons with other existing algorithms are described.

The simulation of the proposed algorithm requires certain parameters as shown in Table 5.1. The number of nodes in the network are considered step by step ranging from 20 to 100 *CR* nodes in the network size of 100 × 100. The *PU* nodes are also considered to be present in the network. The number of *PU* nodes is considered to be $1/10^{th}$ of the number of *SU* nodes. The sensing and transmission range of each node in the network is 10 in radius. The maximum attempt limit in channel allocation is considered to be 15.

Basically, two types of protocols are considered: *Majority Voting* and *Existing Method*. The *Majority Voting* considers the most common results given by the neighboring nodes. Whereas, in the *Existing Method*, the node leaves the selected channel if any one of the neighboring nodes replies with a busy message. The parameters considered in the comparison of both the methods are:

- **Nodes:** It denotes the total number of nodes in the network.

Table 5.1: Assumptions

NUMBER OF NODES:	20, 40, 60, 80, 100
RANGE OF A NODE:	40
NETWORK SIZE:	100 × 100
MAX ATTEMPT:	15
DATA FIELD SIZE:	1KB
LINK DELAY:	1 millisecond (ms)
CHANNEL BANDWIDTH:	1 Mbps
MEAN INTERVAL TIME:	2-20 seconds (sec)

- **Nodes:** It denotes the success percentage of the network in terms of channel allocation for source to destination path. It depends on the number of attempts required to allocate a channel for the path. If the attempt exceeds the MAX_ATTEMPT, i.e., 15, the success status of the path becomes 0 and no channel is allocated to the path.

- **Average Attempt:** t denotes the average number of attempts required for the allocation of channels for a specified path. The attempts considered here are only for the nodes or paths which have been successfully allocated channels.

- **Average Miss Detection:** It denotes the average number of miss detection occurred during the channel allocation for a path in the network.

- **Average False Alarm:** It denotes the average number of false alarm occurred during the channel allocation for a path in the network.

The assignment of faulty nodes in the network plays an important role in the channel allocation. It effects the efficiency of the algorithms to allocate channels for a path. Here, three types of faulty node assignment method is considered which is described in the section below.

5.1 Distribution of Malicious Nodes in Single Hop

The methods discussed here considers a single Source-Destination pair nodes. It should be noted that the *PU* nodes are not involved in the process of cooperative spectrum sensing.

5.1.1 Method I

In this method, the faulty nodes in the network are assigned in such a way that the total nodes in the network are taken as n and the number of faulty nodes are considered to be (n - 1)/3. The faulty nodes are then assigned randomly in the network. As we are considering a random network, the placement of faulty nodes in the network will play an important role in the performance of the network. In Table 5.2 and 5.3, given the arrangement of the network, the result of performance of the network is shown for majority voting method and existing method, respectively.

5.1.2 Method II

Here, the faulty nodes are assigned around each nodes in the network by considering its one hop neighbor nodes. As it is a random network, if node A has n number of one hop neighbors, it considers at least b(n - 1)/3c number of faulty nodes to be assigned in its one hop. Thus, for each nodes, a certain number of faulty nodes are assigned. This method of faulty node assignment is considered for both majority voting and existing method algorithms and the results are given in Table 5.4 and 5.5.

5.1.3 Method III

In this method, the faulty nodes are assigned in the network by considering the common nodes between a source and destination node pair. Firstly, the maximum number of faulty nodes to be assigned i.e., b(n - 1)/3c, is calculated for the node (source or destination)that have at least one hop neighbors. Then, the common nodes of the source-destination pair are taken and is assigned faulty. If there is no or few common nodes between the pair, then nodes in the one hop of source or destination, apart from the common nodes, are taken and are assigned faulty. The results are shown in Table 5.6 and 5.7.

Table 5.2: Majority Voting

Nodes			Type	Success %	Average Attempt	Avg. Miss Detection	Avg. False Alarm	Average Delay
SU	PU	Total						
20	2	22	Text	99.4	2.6	0.6	0.2	105
			Image	99.8	1.6	0.2	0.2	505
			Audio	99	6.4	2.4	0.8	1012.2
			Video	96.6	249	76.6	16	2751
40	4	44	Text	99.8	1.8	0.2	0.2	105.2
			Image	99.6	1.8	0.6	0	507.2
			Audio	99.4	2.2	0.8	0.4	1007.4
			Video	98	5.2	1.2	1.8	2508.2
60	6	66	Text	98.2	2	0.8	0	108
			Image	97.8	2.4	0.8	0.2	508
			Audio	97.4	4	2.2	0	1006.2
			Video	96.8	5.6	4.4	0.4	2509
80	8	88	Text	94.8	1.6	0.2	0	105.2
			Image	94.8	2.4	0.8	0.2	507.4
			Audio	93.2	3.6	1.4	0	1006.4
			Video	92.4	3.8	1.8	0	2506.2
100	10	110	Text	92.8	2.4	0.4	0.2	106.8
			Image	92.6	3	1	0.6	506.6
			Audio	91.4	4.2	1.8	0.4	1007
			Video	89.8	8	3.8	0.4	2511.4

Table 5.3: Existing Method

Nodes			Type	Success %	Average Attempt	Avg. Miss Detection	Avg. False Alarm	Average Delay
SU	PU	Total						
20	2	22	Text	98.2	2.6	0.2	0.2	105
			Image	99.2	7.2	0	2.4	510.2
			Audio	93.6	12.2	0	3.6	1016.4
			Video	88.4	1160.4	2	69.4	3662.8
40	4	44	Text	92	2.4	0	0.8	105.8
			Image	92	3.4	0	1.4	506.4
			Audio	99.6	9.2	0	8	1014.4
			Video	98	32.6	0.2	58	2536.4
60	6	66	Text	93.6	115.8	0	50.6	220.8
			Image	92.2	82.8	0.6	65.8	586.2
			Audio	88.6	77.8	0.2	71.2	1066
			Video	97.8	274.4	0.8	309.2	2757.4
80	8	88	Text	88.4	355.2	0	159	459.6
			Image	88.2	52.4	0	33	555.8
			Audio	85.6	456.6	0	340.4	1459.6
			Video	82.6	1804.6	0	2951.6	4308
100	10	110	Text	85.4	1852	0	831.2	1959
			Image	87	759.6	0	419.4	1261.2
			Audio	82.8	1468.6	0	1492.8	2473.6
			Video	87	1636.6	0.2	3066	4139.4

Table 5.4: Majority Voting

Nodes			Type	Success %	Average Attempt	Avg. Miss Detection	Avg. False Alarm	Average Delay
SU	PU	Total						
20	2	22	Text	99.6	3.8	0.4	1	107.8
			Image	99.8	3.6	0.2	3.4	507.8
			Audio	99.2	17	5.2	3.8	1022.6
			Video	96.6	380.6	73.4	32.8	2883.6
40	4	44	Text	99.8	2	0.2	0.4	104.4
			Image	99.6	1.8	0.4	0	505.8
			Audio	99.4	10.4	0.2	9.2	1016
			Video	99	4	0.4	0.6	2508.4
60	6	66	Text	98.6	2.2	0.4	0.2	105
			Image	97.2	4	1.4	0.4	507.8
			Audio	97	5	2.6	0.2	1008.8
			Video	96	7.2	3.6	0.6	2511
80	8	88	Text	96.8	1.4	0.4	0	106
			Image	95.8	2.4	0.6	0	507.2
			Audio	94.6	2.6	0.8	0.2	1006.6
			Video	92	4.4	1.4	0.4	2508.6
100	10	110	Text	96.2	1.6	0.2	0.2	106
			Image	94.6	2.2	0.4	0.4	505
			Audio	92.2	3.4	1.4	0.6	1006.4
			Video	90	7.6	4.2	6.4	2513.2

TABLE 5.5: Existing Method

Nodes			Type	Success %	Average Attempt	Avg. Miss Detection	Avg. False Alarm	Average Delay
SU	PU	Total						
20	2	22	Text	98.2	88.4	0	47.8	194
			Image	98.3	100	0	50.2	606.6
			Audio	97	282.8	0.2	120.6	1287.8
			Video	92.2	1613.6	3.6	815.8	4117.8
40	4	44	Text	92.2	470	0	331.6	574.8
			Image	91	840.6	0	598	1343
			Audio	87.6	198.4	0	224.6	1200.8
			Video	87	322.4	0	747	2825.2
60	6	66	Text	58.6	10306.8	0	4646.6	10412
			Image	79.4	4605.6	0	4460.6	5110.4
			Audio	75.4	2576	0	3203.2	3579
			Video	77.2	4321.2	0.4	7413.6	6825
80	8	88	Text	23.8	46381	0	19832	46485.8
			Image	75.4	1841.2	0	1714.4	2344.8
			Audio	43.8	39870	0	80716	40873.6
			Video	33	21907	0	73144.8	24410.2
100	10	110	Text	31	213990	0	122880	214094
			Image	51.8	16568	0.4	15139.2	17071.4
			Audio	69.6	540	0	761.8	1544.4
			Video	32	64730.2	0	236432	67233

Table 5.6: Majority Voting

Nodes			Type	Success %	Average Attempt	Avg. Miss Detection	Avg. False Alarm	Average Delay
SU	PU	Total						
20	2	22	Text	99.8	7.8	0	3.8	110
			Image	99.8	4	0.2	3.4	509.2
			Audio	99.2	16.2	2.2	4.6	1021
			Video	96.6	518.8	66.8	43	3022.8
40	4	44	Text	99.6	4.2	0.4	1.2	106.4
			Image	99.8	3	0.2	1.6	505.6
			Audio	99.8	2	0.4	0.8	1005
			Video	99	11.8	1	9.2	2514.4
60	6	66	Text	98.2	3.2	0.8	0	104.8
			Image	98	4.4	1.6	0.6	508.2
			Audio	96.4	3.8	1.2	0.6	1007
			Video	95	4.6	2.8	0.6	2506.8
80	8	88	Text	95.2	3	0.8	0.2	105.2
			Image	94.8	1.6	0.2	0.4	505.2
			Audio	95.4	3.2	1.4	0.4	1007.4
			Video	92.2	4.6	3.4	0.4	2508.6
100	10	110	Text	95	1.6	0.4	0	104.4
			Image	94.4	3	1.2	0.4	507
			Audio	92.6	3	1.6	0	1006.6
			Video	89.8	4.8	3.4	0.2	2508.4

Table 5.7: Existing Method

Nodes			Type	Success %	Average Attempt	Avg. Miss Detection	Avg. False Alarm	Average Delay
SU	PU	Total						
20	2	22	Text	73	770	0	400	873.4
			Image	89.4	1679.8	0	712	2184
			Audio	57.4	8076.4	1.2	8580.4	9079.8
			Video	79.6	4135.6	2.8	5655.2	6639.4
40	4	44	Text	81.2	499.2	0	340.8	602.6
			Image	65.4	6502.4	0	5297.2	7005.8
			Audio	63.2	818	0.6	745	1821.6
			Video	45.6	20465.8	0.2	85475.2	22968
60	6	66	Text	67	7123.2	0	3191	7226.2
			Image	74	2206.4	0.6	1941	2710.6
			Audio	58	7412.2	1.8	11233.2	8415.2
			Video	51.8	12458.6	1.8	32426.8	14961.2
80	8	88	Text	14.6	148957	0.2	72434.4	149061
			Image	27.6	18949.8	0	14995.6	19435.2
			Audio	41.4	29871.6	0.2	52905	30873.4
			Video	28	54351.2	0	211476	56856.2
100	10	110	Text	25.8	14265	0.4	5756	14369
			Image	52.4	7046.8	0.2	5723.2	7551
			Audio	69.2	5134.6	0.6	7470.6	6138
			Video	46.6	18826.4	1	63801	21322.4

Chapter 6

Conclusion

CRN provides increased communication options for users. The rising demand for high-quality multimedia services has been leading to the technological evolution of high bandwidth wireless/mobile communications systems and standards. Security plays a major role in such networks as such networks can be susceptible to manipulation of adversaries. Many security issues can affect a CR network's protocol stacks. For such networks, spectrum sensing attacks or Byzantine attacks may prove important. It affects the correct allocation of channels in the CRAHNs. In this thesis, which helps in fast channel allocation for multimedia communication, Majority Voting algorithm is implemented. Results of simulation are shown and it concludes that the Majority Voting algorithm provides a better result in a distributed CR network than other existing algorithms. The analysis is performed in the network with three different types of faulty node distribution. The delay time analysis for the channel reallocation is also done.

Bibliography

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. "next generation/dynamic spectrum access/cognitive radio wireless networks: a survey". *Computer networks*, 50(13):2127–2159, 2006.

[2] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury. Crahns: Cognitive radio ad hoc networks. *Ad Hoc Networks*, 7(05):810–836, 2009.

[3] A. Bhattacharya, R. Ghosh, K. Sinha, D. Datta, and P. B. Sinha. Multimedia channel allocation in cognitive radio networks using fdm-fdma and ofdm-fdma. *IEEE Transactions on Cognitive Communications and Networking*.

[4] K. Bian and J. Park. "mac-layer misbehaviors in multi-hop cognitive radio networks,". *US-Korea Conference on Science, Technology, and Entrepreneurship (UKC2006)*, 2006.

[5] R. Chen and J. M. Park. "ensuring trustworthy spectrum sensing in cognitive radio networks,". *1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, pages 110–119, 2006.

[6] T. C. Clancy and N. Goergen. "security in cognitive radio networks: Threats and mitigation,". *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, pages 1–8, 2008.

[7] H. Deng, W. Li, and D. P. Agrawal. "routing security in wireless ad hoc networks,". *IEEE Communications Magazine*, 40:70–75, 2002.

[8] J. R. Douceur. "the sybil attack,". *Revised Papers from the First International Workshop on Peer-to-Peer Systems, IPTPS '01, (London, UK, UK)*, pages 251–260, 2002.

[9] S. Haykin. "cognitive radio: brain-empowered wireless communications". *IEEE journal on selected areas in communicationss*, 23(2):201–220, 2005.

[10] C. Karlof and D. Wagner. "secure routing in wireless sensor networks: Attacks and countermeasures,". *Ad hoc networks,*, 1(2):293–315, 2003.

[11] J. Mitola. "cognitive radio for flexible mobile multimedia communications". *Mobile Multimedia Communications, 1999.(MoMuC'99) 1999 IEEE International Workshop*, pages 3–10, 1999.

[12] J. Mitola. "cognitive radio an integrated agent architecture for software defined radio,". *Int. J. Communication Networks and Distributed Systems*, 2000.

[13] J. Mitola and G. Q. Maguire. "cognitive radio: making software radios more personal," , vol. 6, no. 4, pp. 13–18, 1999. *IEEE personal communications*, 6 (4):13–18, 1999.

[14] A. Narayanan. "fast channel allocation and optimum route selection in crahns under byzantine attack". 2017.

[15] C. Perkins, E. B. Royer, and S. Das. "ad hoc on-demand distance vector (aodv) routing,". *Ad hoc networks,*, 2003.

[16] M. Singhal and N. G. Shivaratri. "advanced concepts in operating systems". pages 178–401, 2001.

[17] W. Xu, W. Trappe, Y. Zhang, and T. Wood. "the feasibility of launching and detecting jamming attacks in wireless networks,". *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '05*, pages 46–57, 2005.