Depth First Search Tree based Routing Protocol for Multi-hop and Multi-channel Cognitive Radio Networks

A Submitted
By

Md. Sharif Mollah
ID: 1103078

Adiba Sarker
ID: 1102034

Under the Supervision of
Name of Supervisor
Mamun Ahmed
Department of Computer Science and Engineering
Bangladesh Army International University of Science and Technology

Bachelor of Science in Computer Science and Engineering
Bangladesh Army International University of Science and Technology
Choose an item., Choose an item. Semester

Acknowledgement

In this section, I want thank Allha first and after that my parents and teacher for all support.

MD. Sharif Mollah

Adiba Sarker

Contents

List of Figures

Chapter 1
Introduction

1.1 Introduction

In recent years, the growth of wireless data has taken place an breathtaking matter. Recently a spectrum crisis has created. It industrial named is looming spectrum crisis. Without proper use of available spectrum, we cannot solve the present crisis. Cognitive Radio Network (CRN) consents dynamic spectrum access in multiple channels. So that, if we want to use more spectrum, it is better to respond to increased spectrum demand Bandwidth will focus on getting more unused spectrum Already allocated.

In this article, we attempt to conduct a survey of radio networks for channel allocation by using cognitive radio networks (CRN) in cooperation with major users.

1.2 Conventional Wireless Network

A wireless network is a network that uses electromagnetic waves to propagate in free space to connect workstations or devices. wireless network. Contrary to the guiding media of copper cables and fiber optic cables used in traditional wired networks, wireless transmission is said to use non-guiding media.

Wireless mesh network (WMN) has become a promising technology Used in next-generation wireless networks. Contrary to traditional networks Rely on a small number of wired access points or wireless hotspots In order to connect users, WMN uses wireless mesh network to extend network coverage Routers that communicate with each other through multi-hop wireless communication [1, 2]. The use of WMN include broadband homes, Communities, community and corporate networks, building automation, Health and medical system, public safety and security system, smart Transportation system, Google homepage, Iridium constellation, smart phone Plan, one laptop for each child, emergency or disaster network system and Broadband Internet access in metropolitan areas.

Communication in wireless networks is guided by events. whenever The event triggers the wireless sensor (WS) node to create busy traffic.

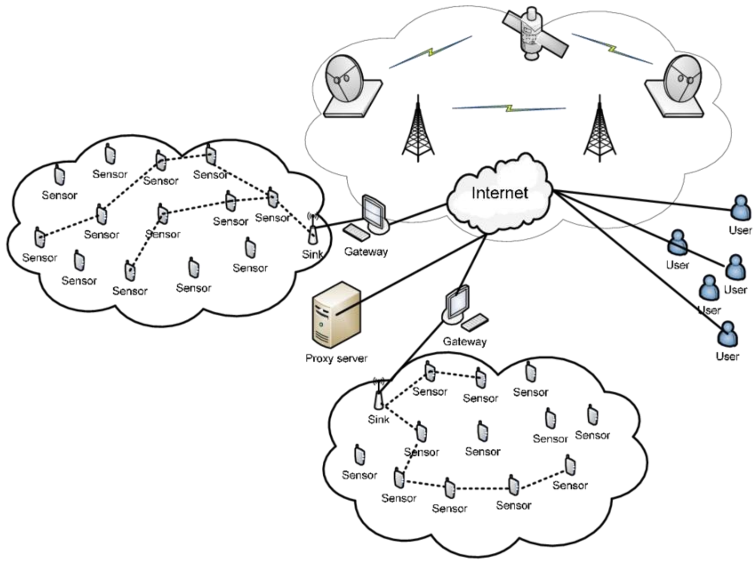


Figure 1.1: Conventional wireless network

Researchers and industry are working hard to improve product performance WSN in terms of cost, energy consumption, data rate, robustness, and network Throughput, service quality and security, etc. Operable hardware and software have been implemented in recent years Improve network performance.

1.3 Multi-Hop Cognitive Radio Network (MHCRN)

Cognitive radio (CR) is a wireless network system. The transceiver can intelligently detect the communication channel being used and the communication channel not being used, and immediately move to the idle channel and close the assumed channel at the same time. By minimizing interference to other users, it optimizes the use of available radio spectrum. The cognitive radio network (CRN) is an opportunistic network which facilities secondary user to use available spectrum.

There are two main types of cognitive radio, full cognitive radio and spectrum-aware cognitive radio. Full cognitive radio takes into account all the parameters known to the wireless network. Detect channels in the radio spectrum and use spectrum-aware cognitive radio.

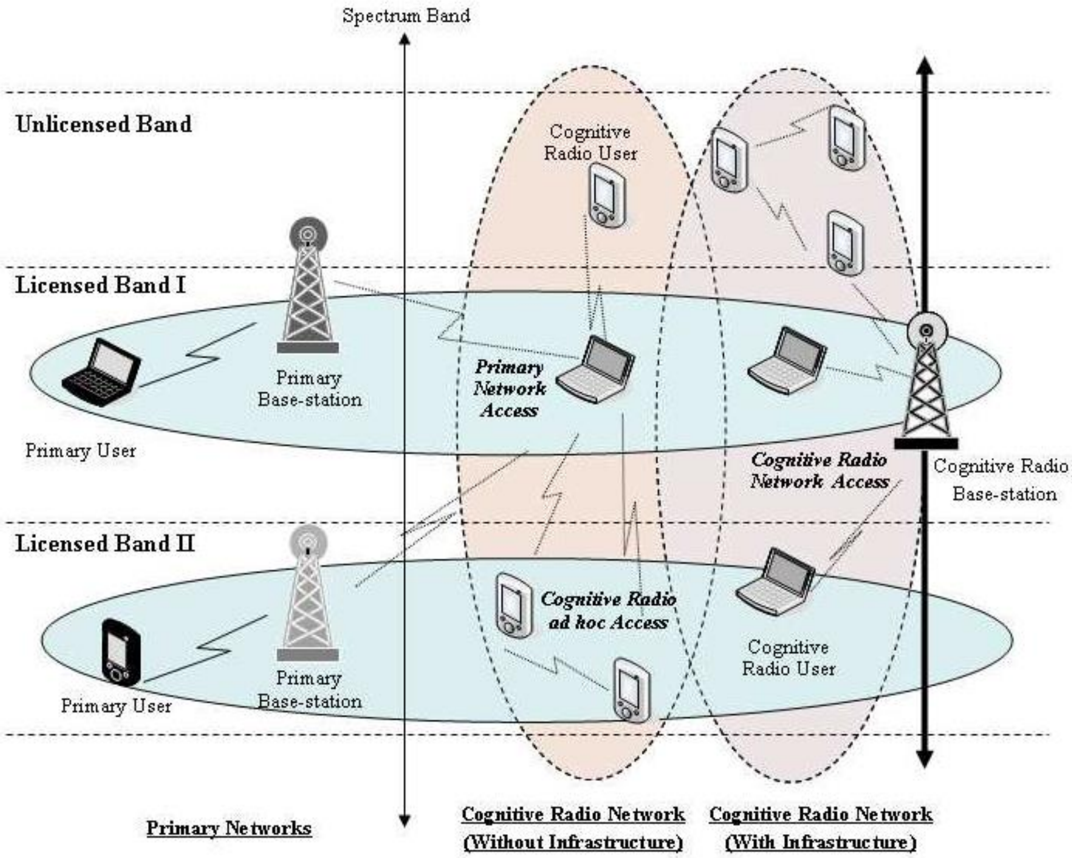


Figure 1.2: Cognitive radio structure

Cellular systems usually use a single hop between the mobile unit and the base station. As cellular systems develop from voice-centric communications to data-centric communications, cell edge throughput is becoming an important issue. This problem is more prominent in systems with higher carrier frequency (more path loss) and large bandwidth (large noise power).A promising solution to the coverage and throughput problem is The use of relays. Various relay technologies are being intensively developed Investigation, including fixed relays (electric infrastructure equipment) Not connected to the network backbone), mobile relay (other users) Opportunistically agree to relay each other' s data packets) and mobile fixed Relays (fixed relays installed on buses or trains and moved accordingly). In recent years, under the guise of relay networks or cooperatives For multi-hop cellular networks, extensive research has been conducted. Research on routing in dynamic spectrum access system is crucial Multi-hop CRN supporting multimedia applications should be considered Recognize the unique nature of the environment. Data forwarding Form a multi-hop cognitive radio network through a few hops The receiver is not within the transmission range of the sender. Generally two The user is connected in a multi-hop cognitive network, which depends on Universally available frequency.

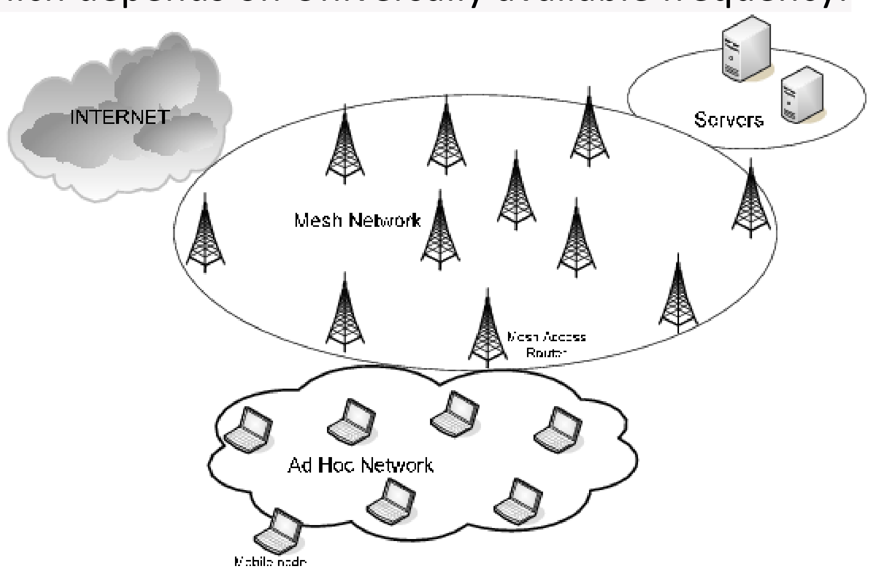


Figure 1.3: Multi-hop mesh network.

In a multi-hop cognitive radio network, each node has one or more radios. For communication, each radio can be changed to the required Frequency (channel). Therefore, the ultimate challenge of multi-radio, multi-channel cognitive radio The network will allocate channels in a way that results in the optimal routing algorithm.

1.4 Motivation of the Work

The working principle of cognitive radio technology is dynamic spectrum access, in which secondary users exploit spectrum loopholes. In a multi-channel network, the available communication channels vary with the main user's traffic. A route is possible only when there is at least one for each pair of users Public channels available in their respective locations. Due to the dynamic nature of the physical layer, sometimes routing may fail. Therefore, a new set of available channels needs to be used to discover a new route. Graph theory method is very effective in traditional network modeling, but it fails to effectively model the cognitive radio network environment.

This motivates us to introduce a multi-edge plan model, where The routing protocol is based on a Depth-First Search algorithm. This model Embed the dynamic behavior of MHCRN into its edge weight. Such a model Taking into account various radio channels, it is simpler and more accurate Determine edge weights and find the shortest relevant parameters path.

1.5 Importance of Appropriate Channel Selection for Routing

The inefficient and sporadic utilization of the spectrum, coupled with the increased utilization of the radio frequency environment, has reduced the quality of service of various wireless networks and applications (such as cellular networks). Therefore, spectrum switching is very important in cognitive wireless networks to ensure that secondary user communications have sufficient service quality and performance. Spectrum mobility is also called spectrum handover, and can be defined as the following process: when channel conditions decrease, for example, when the primary user (PU) reaches the same frequency or the SU is interfering with the PU. The selection criteria for selecting the best alternate channel are the probability of channel availability, estimated channel time availability, signal-to-noise ratio and interference ratio, and bandwidth.

The fixed spectrum allocation has led to inefficient use of licensed spectrum, so the cognitive radio network (CRN) emerged As a promising solution. Cognitive radio networks use spectrum loopholes opportunistically, that is, spectrum that is not used by the main user (or PU or licensed user), and dynamically allocate spectrum. For every network, including cognitive radio networks, routing is very important.

Routing is the basis of communication that transmits data from one machine to another. The other is multi-hop mode. Effective communication requires a good routing protocol, and a good routing protocol is based on the channel Choose a strategy. An effective routing protocol requires a good channel selection strategy so that the routing should be stable and exist for a longer time.

1.6 Contribution of Work

The main goal is to find an appropriate path leading to the target node, which has more than one hop distance in the CRN environment. To do this, we need to create a network model consisting of a certain node, and declare the two nodes as a source node and a target node. The main goals and possible results of this work may be mentioned below: Use the following commands to build a logical loopless topology for the cognitive radio network DFS-based protocol. Choose the best message channel by minimizing interference. Determine the percentage of successful routing for a single radio and multiple radios Switch or not switch to use wireless network DSDV with switching or DSDV calculation without switching accuracy.

1.7 Organization of the Thesis

The report is organized as occurs. In chapter two, an overview of terminology related to the project, including a short discussion Regarding previous works that have been implemented but have their limitations. Chapter three introduces the working process of our proposed system. In Chapter four, we have detailed the implementation of the project. Chapter five focuses on the experimental results of the system. To evaluate the system, we used subjective and quantitative Measures. The paper concludes with the research results Work in Chapter six.

Chapter 2

Literature Review

2.1 Literature Review

The main motive of implementing Cognitive Radio Network (CRN) in the modern telecommunications industry is to reduce costs through proper use of spectrum. When integrating CRN into a wireless network, the challenge is to manage interference. However, through proper use of channel allocation, this interference can be minimizing or even possible to erase from root.

In order to sort out interference, it helps to use multiple channels and multiple radios in each radio, and is called MRMC (Multi-Radio Multi-Channel) CRN. In addition, multi-radio nodes can simultaneously transmit and receive data packets by using two different radios, thereby increasing throughput.

Here this chapter we have discussed MRMC CRN, channel assignment and switching in wireless network communication.

2.2 Cognitive Radio Overview

Extensive research on 5G networks and its revolutionary scriptures Success marks the importance of current wireless networks, the days to come. The demand for wireless spectrum is growing rapidly But be aware that its number is limited, so it must Spectrum usage flexible.Federal Communications Commission (FCC) Cause CRN to use unused frequency bands for unlicensed users. At present Spectrum framework, the allocation of spectrum bands to license holders, In large demographic areas, also known as primary users (PU) Long-term basis. However, part of the spectrum band is used PU. Inefficient use of spectrum bands can be caused by Unlicensed users are called "auxiliary users (SU)" for a temporary period. This one the technique of allocating spectrum to SU is called dynamic spectrum Allocation (DSA). Authorized users have priority in the spectrum band, while SU needs Continuously monitor the use of licensed spectrum. In this way, SU the interference temperature should not be violated. SU should pay attention the reappearance of PU. Technology for sensing the presence of PU It is called spectrum sensing. There are multiple sensing technologies, such as Energy detection, cyclostationary feature detection, matched filter, central Cooperative perception and distributed cooperative perception. In the spectrum Sensing SU continuously senses/checks the transmission channel Whether there is a main signal in the channel. After sensing the spectrum CR allocates spectrum bands to SU, but uses spectrum bands SU must reconfirm.

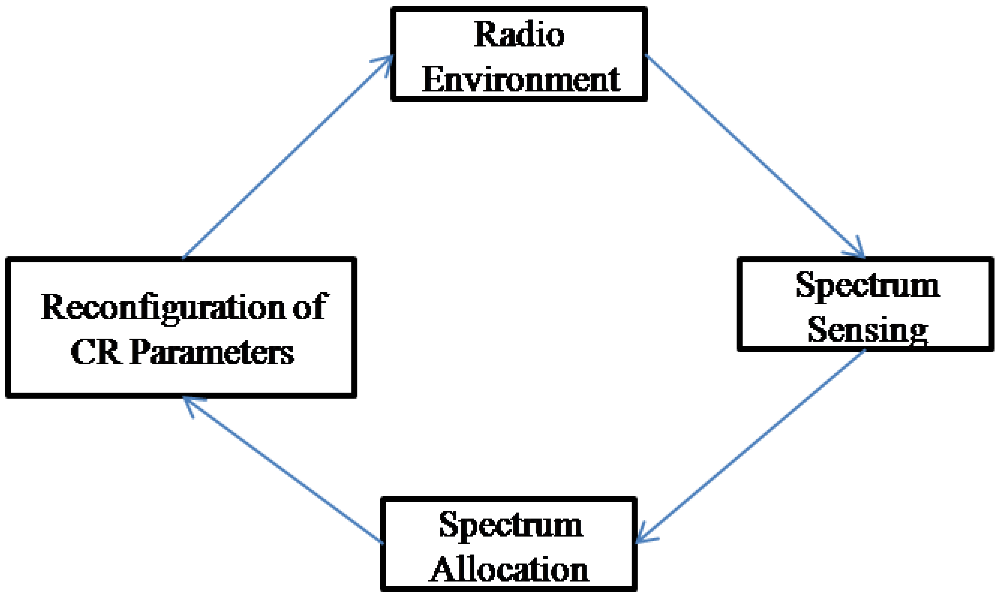


Figure 2.1: Cognitive Radio Cycle

2.2.1 Characteristics of Cognitive Radio

CR network has few characteristics. For operational state languages,Using operating state language for share information of CR network. As mentioned above, CR should inform other nodes in its network of its status and observation results. The language CR uses for this intention is called the operating state language. The information sent by CR may be a list of all transmitters it has recently detected. For operating environment sensing,Multiple dimensions may include open CR run cooperative or non-cooperative transmitters and open environments that can be adapted to local changes and rapid changes in traffic conditions. In order to perform its tasks correctly, CR must be changed according to the changing environment, and it should ability to notify other devices in the network of configuration changes. For distributed resource management, Radio spectrum is a distributed resource. So use one spectrum band in one location make it unavailable elsewhere. So the spectrum setting resources must be done in a balanced manner. Various algorithms have Developed to handle distribution and manage distributed Spectrum and resources based on traffic load.

2.2.2 Functions of Cognitive Radio

The functions which are performed by Cognitive Radio Network given below:

For spectrum management, it is necessary to capture the best meet the user's communication spectrum requirements. CR should determine the best spectrum to meet the quality of service requirements on all spectrums. The management function is divided into spectrum analysis and spectrum detection. For spectrum sharing, It is essential to provide a fair spectrum scheduling strategy. This is also one of the most important challenges in the use of open spectrum. In the existing system, it corresponds to existing MAC problem. For spectrum mobility, This is the process of CR users exchanging passwords Operating frequency. Their goal is to use the spectrum in a dynamic manner by allowing the radio terminal to work at its best. frequency band. The transition to a better spectrum must be seamless. For spectrum sensing,In order to avoid interference to the system, knowledge of the spectrum environment must be understood. PU detection technology,this is the most effective method. Spectrum sensing technology is basically divided into three categories, namely transmitter Detection, cooperative detection and interference-based detection.

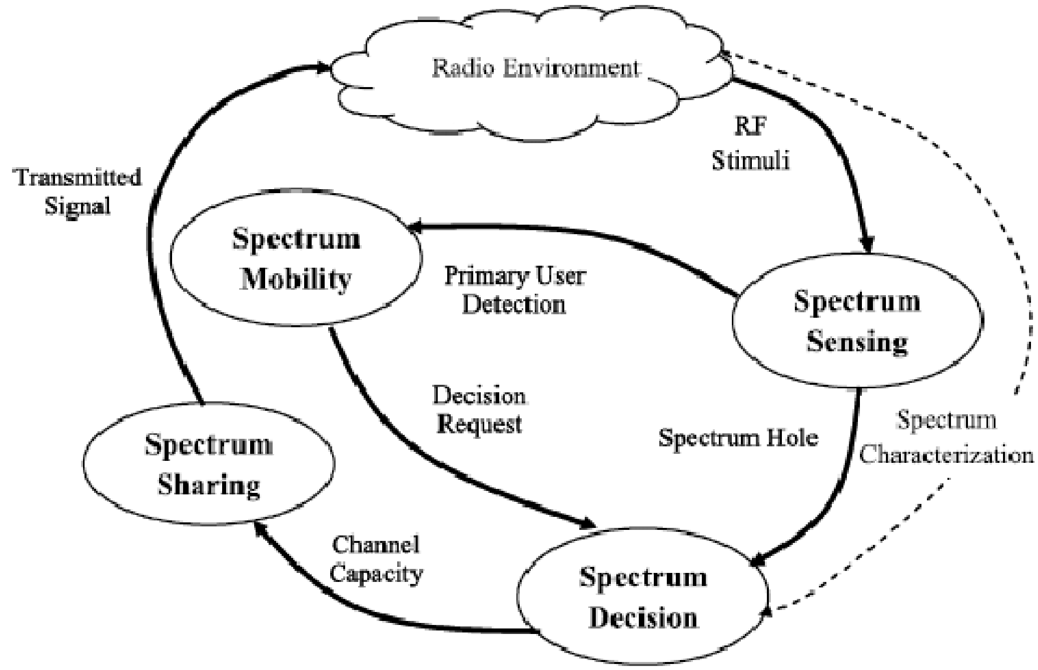


Figure 2.2: Functions of Cognitive Radio Network

2.2.3 Cognitive Radio Concept Architecture

Cognitive radio consists has two main subsystems. Which are cognitive unit and included. The cognitive unit that makes decisions based on input but other is the operating unit Software that can run in multiple modes. To identify other users in the system, a separate spectrum detection subsystem is also included. These subsystems are not individual devices, but components are distributed throughout the system. Sothat,we can see cognitive radio is not a single system, it is cognitive radio system orCognitive radio network.Cognitive Unit (CU) is divided into two parts. The first engine labeled "cognitive engine" effort to find solutions or optimize performance goals based on the current internal state of a given radio and input from the operating environment. The second engine is "policy engine" and used to ensure that "Cognitive engine" is based on regulatory rules and other policies The outside of the radio.

2.2.4 Communication Channels and Channel Interference

Interference is one of the main factors affecting wireless performance the internet. Wireless networks are doing their best to get rid of long-term damage and interference. In addition, this issue is very important for CR networks, because a promise of CR network is not to have any harmful effects on PU. Figure 2.3 shows how harmful interference is used in CR network. It can be seen from the figure that if there is no interference signal, the bit error rate (BEP) will drop sharply. sothat, the system must be interference-free to reduce BEP and improve PU performance.

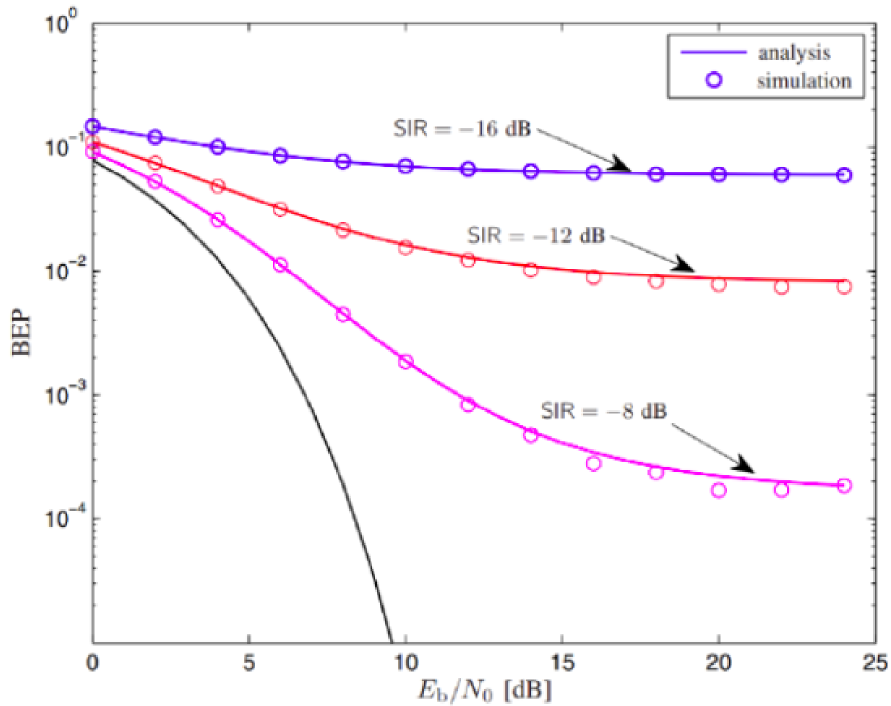


Figure 2.3: Bit error probability (BEP) of a BPSK system in the presence of cognitive network interference

The frequency band on which the signal is sent is called the communication channel. Channels are identified by their frequency, bandwidth and alias. Aliases represent channels as integers, for example Channel 4. In figure 2.4, Explains all channels in the IEEE 802.11b standard and 27 They correspond to the channel number, center frequency and bandwidth.

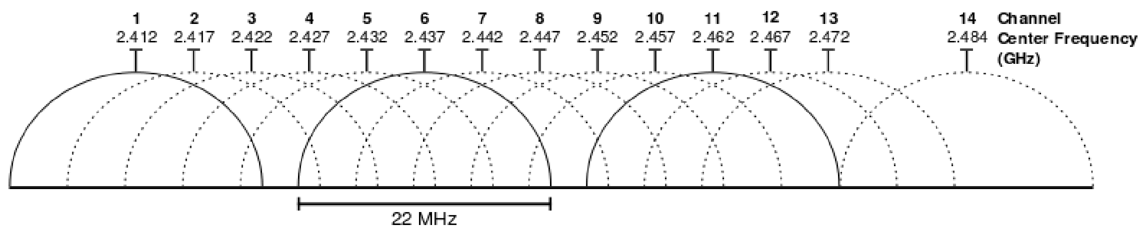


Figure 2.4: IEEE 802.11b standard

For example, the center frequency of channel 6 is 2.437 GHz, Its frequency bandwidth is 22 MHz IEEE 802.11b operates on 2.4 GHz Spectrum, 11 of the 14 channels are available. Two channels If their frequency bands cross each other, they are called overlapping channels Others are called non-overlapping or orthogonal. E.g, In Figure 2.4, channels 1 and 3 are overlapping channels, while channel 1 And 6 are orthogonal channels. In CRN communications, interference refers to any change or It destroys the signal when it travels along the channel between the transmitter and receiver receiver. Beyond the transmission range, the receiver cannot receive The data from the transmitter is correct because the signal from is not strong enough. However, the signal from the transmitter is still strong enough Interfering with the receiver's reception from another transmitter. As shown in Figure 2.5, C is not within the transmission range of A, but Within B's transmission range. However, the signal from A may interfere C receives from B. Therefore, the interference range of the node is New transmission may interfere with the range of data packets Reception. Here, the interference range is always greater than and covered Transmission range.

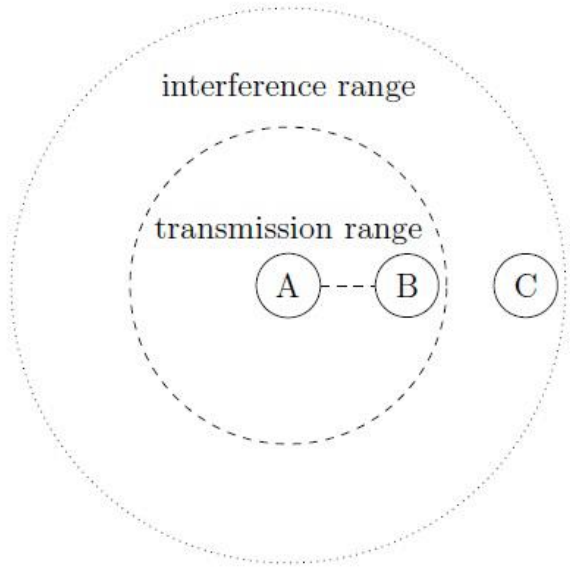


Figure 2.6: Interference range and transmission range

Two types of interference may occur in wireless communication: co-channel interference and adjacent channel interference. When two radios are in the interference range, co-channel interference occurs Send data to the same location on the same channel at the same time receiver. Adjacent channel interference occurs when two radios are placed Simultaneously send signals within the interference range of each other On two overlapping channels. Channels with more overlap will generate more Adjacent channel noise. For example, channels 1 and 2 introduce more The adjacent channel interference ratio is channel 1 and 4.

2.3 Background and Present State of the Problem

In the traditional sense, radio is the passage of electromagnetic signals Atmosphere or free space. In wireless network literature, broadcasting Usually used to describe a wireless network interface card (NIC) Connect to the antenna of the wireless device. In this article, the term "Radio" and "interface" can be used interchangeably.

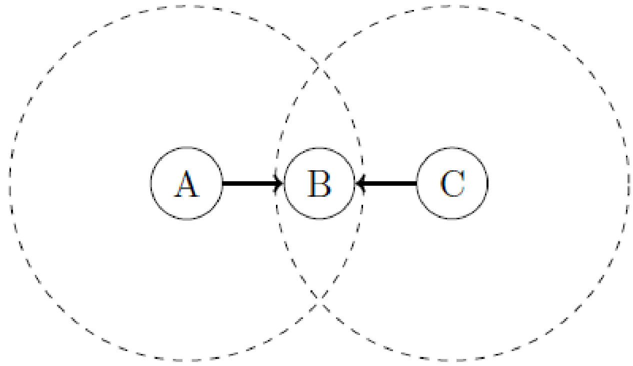


Figure 2.6: Transmission range of nodes A and C is represented by the dotted circles

Channel indicates the frequency band on which the signal is carried from source to destination. At any time, the radio can Send or receive signals on a specific channel. Transmission range The range of the radio is designed to be the range where the node can receive the radio data transmission is correct (assuming no interference). Transmission the range of nodes is usually surrounded by Node (see Figure 2.6).As shown in the figure, if node B is located the transmission range of node A, then node B can successfully receive data assuming that both nodes are in a non-interference environment, it comes from A.When the sender node sends data to the receiver,The transmission range of the sender may overhear its transmission, even if they are not the intended recipients of these transmissions.

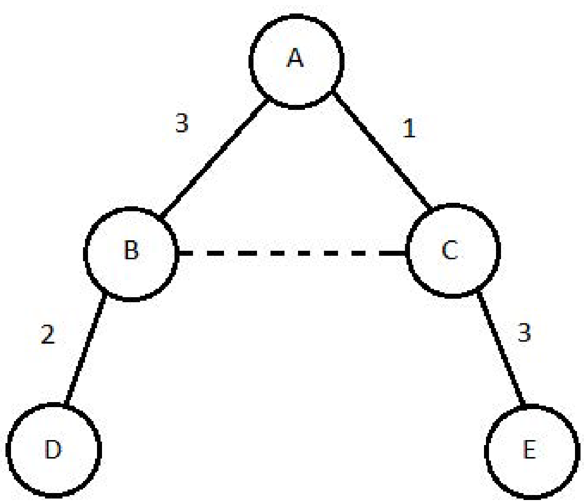


Figure 2.7: Overhearing eect

In Figure 2.7,it is example of the overhearing effect is illustrated. In thisexample, node A is the multi cast source and nodes B, C, D and E are forwarders. There is a wireless link between node B and C, but it is not part of the routing tree; therefore, B and C are not connected directly,however, they are in the transmission range of each other. Since B is receiving multi cast ow on channel 3 and C is transmitting on the same channel, node B is able to overhear the data that node C is transmitting to node E. In this case, if node B does not receive a packet from its parent, node A, (due to e.g., a collision or interference), there is another chance for B to receive (overhear) the packet from node C. As a result, the broadcast nature of wireless communication provides the overhearing effect which has the potential to improve packet delivery ratios of nodes.In wireless communication, concurrent data transmission on the same data The channels from nodes A and C to node B cause packet collisions at B, and Cause data loss. In order to minimize this conflict, professional awareness The multiple access (CSMA/CA) algorithm to avoid conflicts [10] is used. CSMA/CA is a media access control (MAC) protocol, It was approved for use in the IEEE 802.11 standard in June 1997.

2.3.1 Routing in Graph Representation

Some widely used routing protocols, such as AODV, DSDV, DSR, etc. Implemented in a multi-hop network. In these protocols, the network is Graphically. In the graph-based representation, the nodes are users and The edge is the connection between users. Distance from one Represent other nodes by assigning weights to edges. Communication protocol The graph-based scheme mentioned above is used for communication. Therefore, the graph theory model is Modeling traditional networks can be used to determine the best route effective.

Traditional wireless network it assumes two case,such as:

i)Fixed spectrum allocation

ii)Fixed channels are always available

The result of these assumptions is that the graphical model is sufficient to represent Traditional wireless network. But in a multi-hop cognitive radio network (MHCRN), the channel is Not always available for SU. It only depends on the PU flow. therefore, The physical layer of MHCRN is different in terms of power, constraints, and interference And bandwidth. To discover the route to start the network, first check Idle channels of each node, and sort out public channels For each pair of nodes. If multiple channels are found, then According to certain criteria (e.g. interference level, Bandwidth etc.

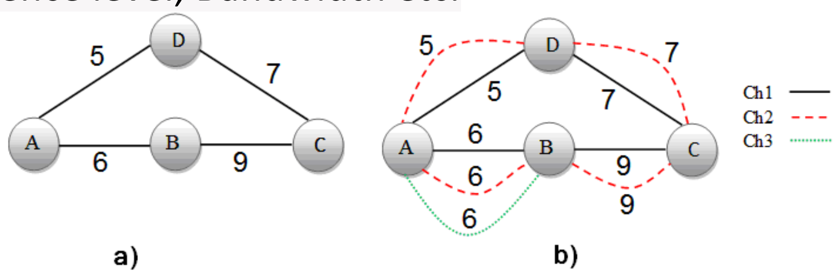


Figure 2.8: Graphical representation in traditional network(a) and Cognitive Network(b)

Different from traditional graphical models in the multilateral graphical model (MEGM) As shown in Figure 2.8, it has multiple edges between two nodes. This extra MHCRN account characteristics of available channels, possible routes To reach the destination through various channels, and the collection of neighbors in each channel. The weight of MEGM represents channel characteristics. In stratification Each layer of the graph model represents a different channel. Edge complexity The value of MEGM is much lower than the hierarchical graph model. cut back The complexity of the graph model was developed by Yogesh R Kondareddy and Prathima Agrawal. In this article, the model is formulated in a custom simulator in C++. of The simulator uses the DSDV routing protocol.

2.3.2 Spectrum-Aware Dynamic Channel Assignment

The distributed spectrum sensing dynamic channel allocation scheme is Proposed by Yasir Saleem and Adnan Bashir [11] for MHCRN. In this scheme The CR network will first check the activity of the PU. If there is no activity PU then searches for maximum connectivity and minimum interference Between the two nodes of all channels. After considering the channels allocated to other nodes, calculate the minimum interference between CR nodes, The best channel will be allocated on the multi-radio interface. The goal This work is to minimize the interference to the PU mode and maximize Connectivity. In this article, they reduce the interference to PU nodes and increase The transmission rate of the data packet. But they didn’t consider the main station Occupancy rate while calculating channel quality.

2.3.3 Channel Assignment (CA)

The task of the CA scheme is to determine which channel two adjacent nodes are on Should share and exchange. The channel allocation algorithm is An important factor in improving network performance. When deciding Regarding channel allocation, different CA algorithms consider different goals. Some examples of goals are: minimize interference Maximize overall network throughput , or balance traffic load This can be done in MRMC WMN. Represented by using the concept of graphs.To do this, we assume that:

i)G = (V, E) is a graph in which vertices represent mesh nodes and edges represent the direct communication links.

ii)There exist a communication link between a pair of nodes (u; v) if the transmission range of radios given that all radios use the same power to transmit data.

iii)Two nodes are neighbors if there is a link between them.

2.3.4 A Minimum Spanning Tree based Routing Protocol

In the minimum spanning tree (MST), a topology for routing is formed. Here, the route from the source to the destination is found through the MST-based protocol. they The redundant neighbors of each node are eliminated. Remaining nodes It is likely to be the next hop of transmission. try to deduct Cost, considering the weight of the edge. Agreement is Even if the network performance is improved, it will not decrease User number. MST-based routing protocols have a higher success rate Better than DSDV routing protocol.

2.4 Chapter Summary

This chapter provides an overview of cognitive radio networks According to definition and terminology. Then there is the concept of CR architecture Discuss channel interference. The routing in the graphical representation is discussed for MHCRN and MEGM. Finally understand the spectrum Review dynamic channel allocation and determine channel allocation issues Develop using the concept of graphs.

Chapter 3

Methodology

In this chapter, we will discuss successful routing algorithms from source to destination. The focus of the discussion is graph algorithms We have used it. Dijkstra's shortest path algorithm is discussed by Picture presentation. This chapter mainly focuses on The proposed system and graphical representation of the system.

3.1 Representation of System Model

Cognitive Radio (CR) is a form of wireless communication in which The transceiver can intelligently detect which communication channels Use and not use, and immediately move to idle channels, while Avoid being occupied. In the CR network topology, there are many networks Consider which are located in different locations to provide services user.Channels should be allocated in a certain way so that The positioning does not cause any type of interference. The number of networks is It is not always constant. Therefore, pay a sufficient price when designing the model The proposed model will not decrease its performance as performance increases Number of nodes. In CR, the network consists of a node and some According to some properties of nodes, there are edges between nodes The internet. This is done so that the problem can be represented graphically.

3.2 Multi-Edge Graph Model

In the polygonal graph model, there are multiple nodes in the middle Two nodes. This new feature of the diagram is used for modeling Additional features of multi-hop CRN, such as available channel set, Possible routes to other nodes through various channels and neighbor sets Every channel. This section details the recommended model.

3.2.1 Graphical Representation of CRN

In a multi-hop CRN, routing depends on the available channels, and the available channels Depends on the main user's traffic. This means the best choice Routing, the routing layer should provide relevant Available channels set by the PHY layer on each node. In addition, The total number of available interfaces on each node The best choice for the route.For example, if an intermediate node in the route has multiple Interface, it can choose two different input and output channels Communication, thereby reducing interference and increasing throughput increase. Assuming information about totals The interfaces and available channels on each node are provided for routing Node layer. In the multilateral graph model, representing each user As a node in the graph. Each channel is represented by an edge.

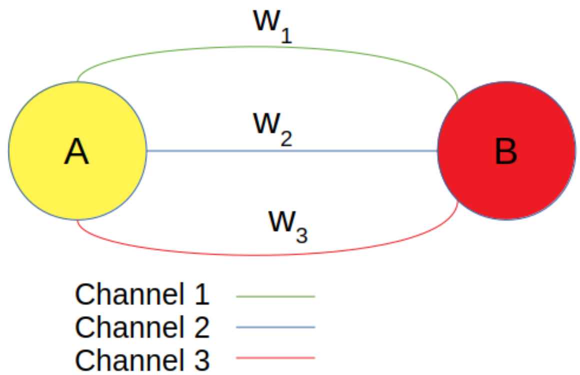


Figure 3.1: Node A and B linked by 3 dierent channels

Let graph G be a multilateral graph and N be a node set And C represents the set of all possible channels. So a pair of nodes can have If more than one edge can use different frequencies (channels) Communication is different from the traditional graph model. In the traditional graph model, each pair of nodes There is only one common channel. For example, if nodes A and B have three communication channels, then As shown in Figure 3.1. A and B can pass Channel 1, Channel 2, and Channel 3. Therefore, nodes A and B are connected By three edges. W1, W2 and W3 represent Each channel can be a simple distance or a complex combination Different factors, such as interference, energy, etc. Consider the distance here As the number of hops from one user to another. Every channel is The channel ID is assigned. For example, assign channel 1 as "1" as the channel ID. In traditional networks, each node in the graph has a fixed channel allocation. Since all nodes in this network can only communicate in one channel, 37 Only one edge can communicate between each pair of nodes each other. The numbers on the edges represent In this case it is the edge of the distance.

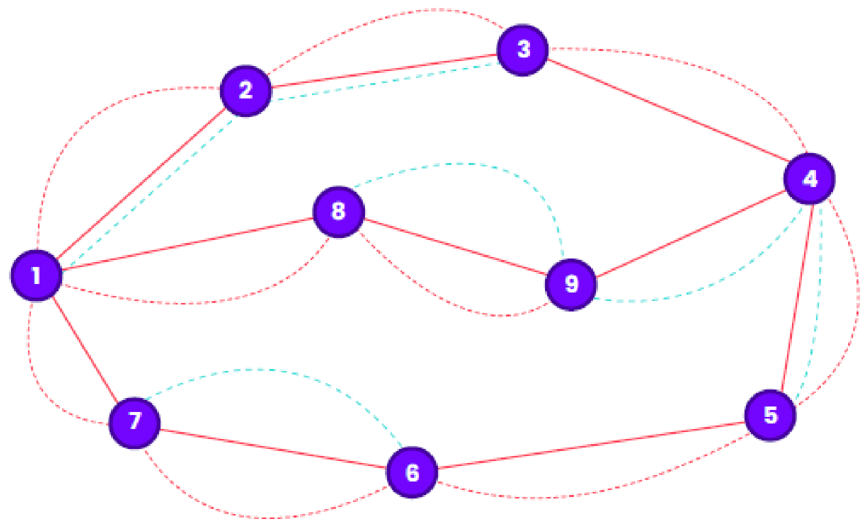


Figure 3.2: Graphical Representation of traditional network

Now, consider a graph with 9 nodes and 3 different channels Figure 3.2. Each node may have different channels available. A kind Cognitive network where each node has a different number of available channels Use it. Here, all edges between each pair of nodes are assigned the same Weight, because weight represents the distance between a pair of nodes It does not change with the channel type.

3.2.2 Topology Formation

The most direct consideration of node redundant neighbors Is that the node will not choose this neighbor as the next hop Transmission under any circumstances. Propose a scalable deletion solution Redundant neighbors of each node, an algorithm aims to retain at least k neighbors, and the selected neighbor is likely to become The next hop of the node in transmission. To keep the best neighbor In terms of minimizing the total link cost, the number of each node It is considered by calculating the link weight.

3.2.3 Weight Assignment and Channel Switching

All graph-based protocols find paths based on the weight of edges The weight of the edge of the graph represents the distance between On the node, any graph-based protocol will choose the shortest path route. If the weight is a function of interference, the route is lowest Select interference. So weight distribution is very important Steps to build a graphical model. In the multilateral graph model, a problem Due to the multilateral nature of the graph, interface constraints appear.

3.2.3.1 Interface constraint

A node can have one or more radios. Each radio (interface) is It can be changed to the required frequency (channel) for communication. In traditional wireless networks, it is feasible to switch interfaces between channels Network, although it requires synchronization between neighboring nodes And introduce overhead. But the channels in the cognitive network are Distributed on a large frequency spectrum, the channels may be separated Big band. In this case, switching may be impractical Channel for packet granularity. A way to avoid choosing routes in it The following is a suggestion as to what kind of handover occurs.

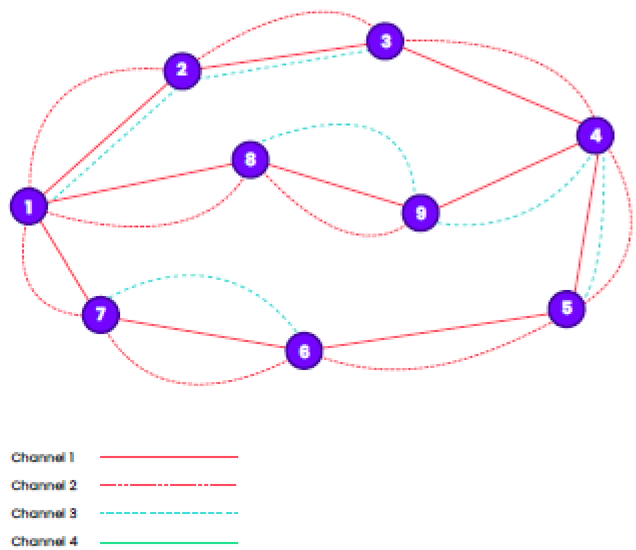


Figure 3.3: Graphical representation of Cognitive Radio network

3.3 Proposed Algorithm

Steps required to discover routes-

Generate graph, G.
Topology formation by DFS.
Selects appropriate channels based on radio.

3.3.1 Algorithm 1: Generate Graph

Algorithm for graph generation is given below-

1. Add a node Ni to the graph G for each user in Multi-Hop CRN.
2. Add an edge between node Ni and node Nj if they are potential neighbors through channel Ci for all Ni , Nj 2 N and Ci 2 C.
3. Assign the weight for the edge connecting nodes Ni and Nj for all Ni , Nj 2 N.
4. Generate G (v, E).

3.3.2 Algorithm 2: Topology Formation using DFS

```
Algorithm for topology formation using DFS is given below-

Input: G(V, E)
Output: Topology for the routing phase, G* (V*, E*)
DFS(G, s) where u is the source node,
   let S be stack
        S.push(u)
        mark u as visited.
        while(S is not empty)
           v = S.top()
           S.pop()
           for all neighbours w of v in Graph G
           if w is not visited
                S.push(w)
                mark w as visited
return G* (V*, E*).
```

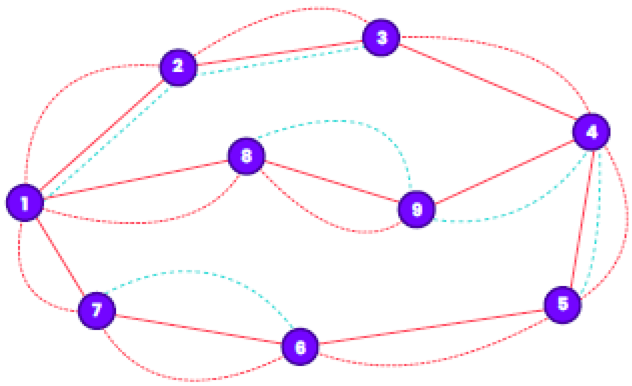
3.3.3 Algorithm 3: Channel Assignment

The Algorithm for Channel Assignment is given below-

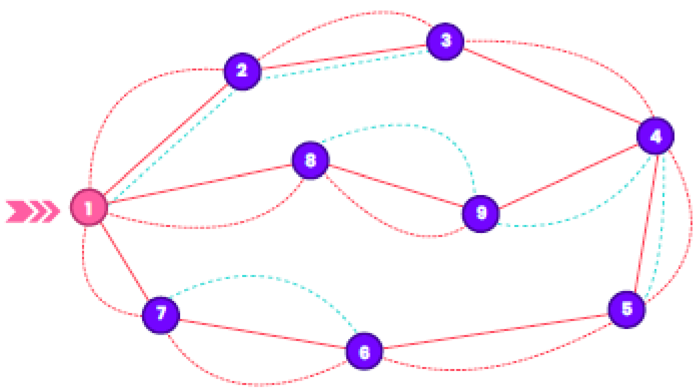
1. Input: G(V, E)
2. Output: Topology for the routing phase, G^L (V^L, E^L)
3. BFS (G, s) where s is the source node.
4. let Q be queue.
5. Q.enqueue(s)
6. mark s as visited.
7. while (Q is not empty)
8. v = Q.dequeue()
9. for all neighbours w of v in Graph G
10. if w is not visited
11. Q.enqueue(w)
12. mark w as visited.
13. Return G^L (V^L, E^L).

3.4 Pictorial Representation of Sample Network

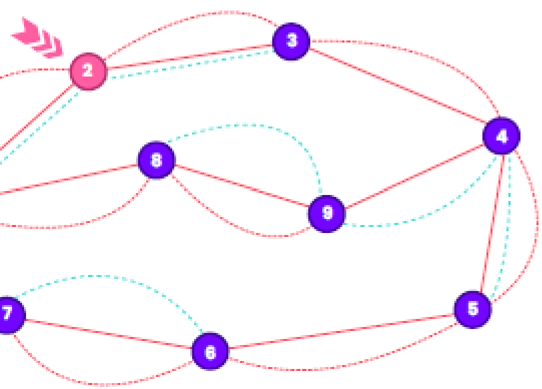
Now in here draw my dfs node routing.



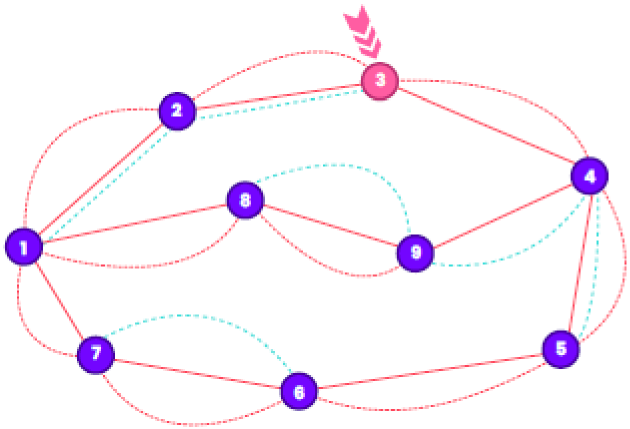
Step-1



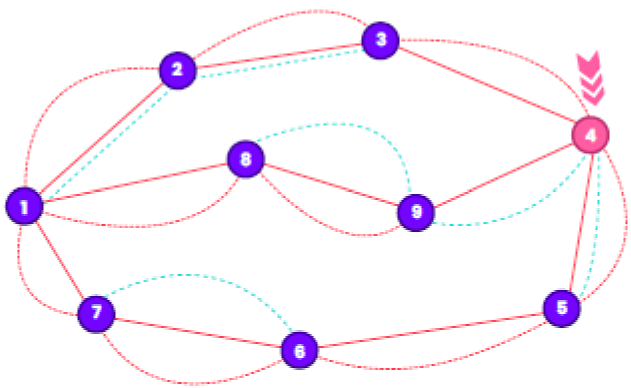
Step-2



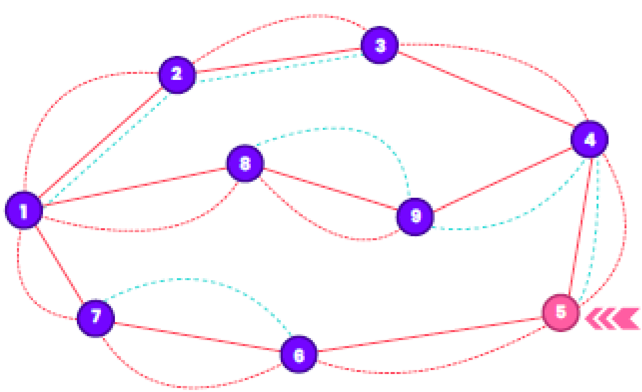
Step-3



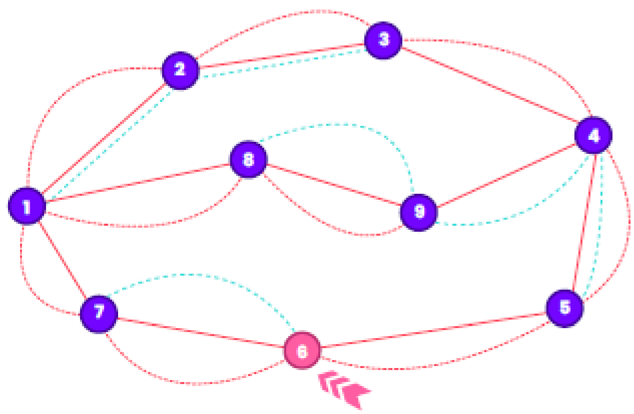
Step-4



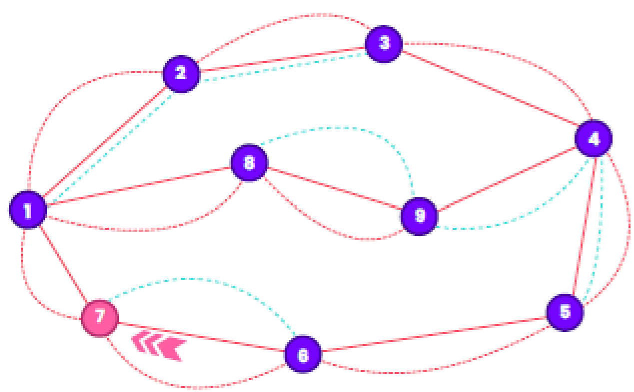
Step-5



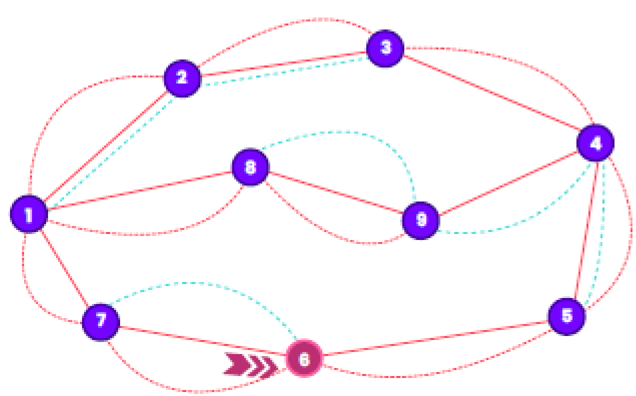
Step-6



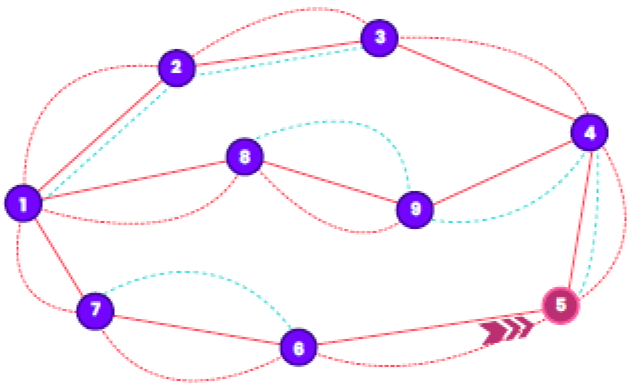
Step-7



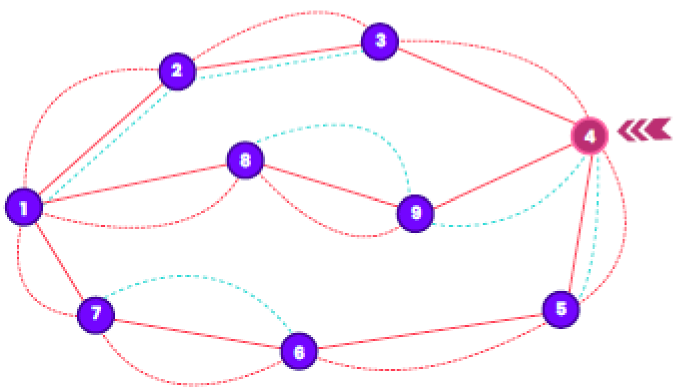
Step-8



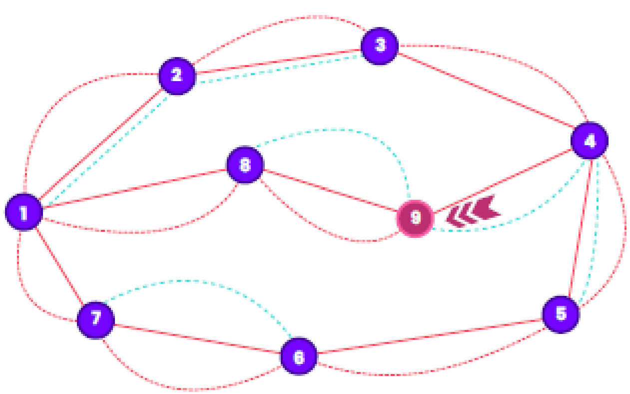
Step-9



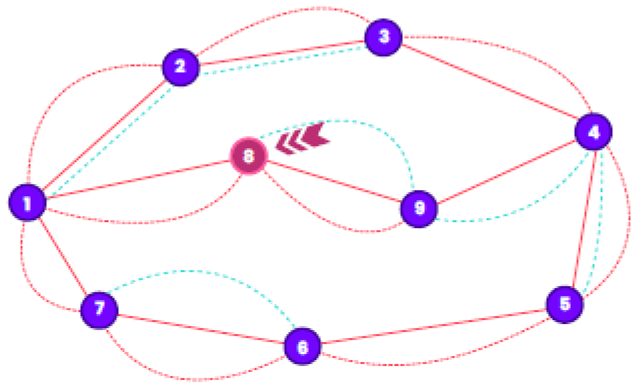
Step-10



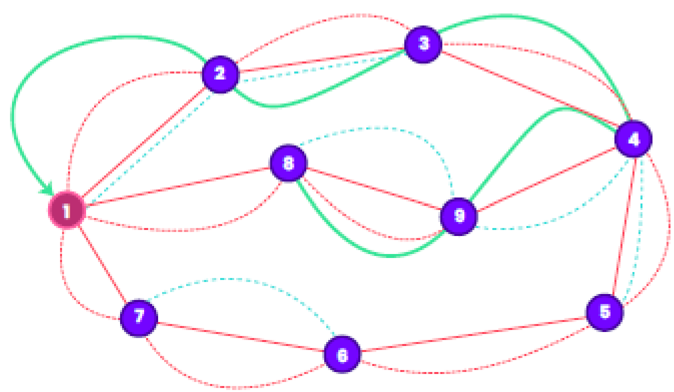
Step-11



Step-12



Step-13



Step-14

3.5 Channel Assignment

After formatting the topology, the routing is discovered by assigning channels according to Algorithm 3. The packet is transmitted to the target in the following way: Channel to switch or not to switch.

3.5.1Single Radio Network

Suppose a data packet is sent from node 1 to node 9. Travelling node are,

1→2
2→3
3→4
4→5
5→6

6→7
After back routing it also follow single way
7→6
6→5
5→4
4→9
9→8

3.5.2 Multi Radio Network
In a multi-radio network, you can switch channels, that is, input and output The channels of the nodes may be different. So the route from node 1 to node 9 is as follows based on the minimum cost from source to destination:
1 →7(Usingchannel1,2,3,4,5,6,7)

After back routing it also follow multipul node

7→8(Usingchannel 7,6,5,4,9,8)

3.6 Chapter Summery
In this chapter, route discovery in multipath will be explained. Edge planner diagram. The routing protocol based on depth first search (DFS) is It is used for path search and assigns channels according to network type. For a single radio network, it is impossible to find routes for all channels. it is There must be a public passage from source to destination. but For multi-radio networks, channels are allocated based on available groups Channels, and there is no need to establish public channels from source to destination destination. This is why the percentage of multi-radio routes successfully Not just a single radio network.

Chapter 4

Implementation Details

We need system requirements to implement. Main meaning The purpose of the proposed system is to reduce the interference between nodes, and Channel allocation. This operation task is executed by software. Since the software runs in the operating system, you need to specify Requirements for running this software. To run this project, there are two types Demand requirements, such as hardware requirements and necessary software tool. For the proposed model, the system requirements are listed below:

1. Hardware Specication:

Personal Computer
Core i5 8th Generation Processor
1.60 GHz x 8 clock speed
Intel UHD Graphics 620 (Kabylake GT2) GPU

2. Software Specication:

Operating System Ubuntu 18.04.2 LTS
Linux Platform
Programming Language used :- C++
IDE : Code::Blocks 16.01

4.1 Functions and Data Types

4.1.1Stack

The stack is a container for objects inserted and deleted according to the last-in first-out (LIFO) principle. In a push-down stack, only two operations are allowed: push the item onto the stack, and then pop the item off the stack.

4.1.2 Vectors

In structured programming languages, vectors are similar to dynamic arrays Ability to automatically change the size of elements Insert or delete, its storage is container. Vector uses dynamically allocated array to store its elements internal.

4.1.3 size()

It returns the number of elements in the vector. This is the quantity The actual object stored in the vector is not necessarily equal to its vector storage capacity.

4.1.4 empty()

The empty() function of the queue returns empty whether its size is zero. If the queue size is zero, it will return a true otherwise it will return false.

4.1.5clear()

Removes all elements from the set container (which are destroyed), leaving the container with a size of 0.

4.2 Existing Algorithm Implementation

The minimum spanning tree (MST) is a subset of connected edges, An edge-weighted undirected graph that connects all vertices together, There are no loops and the total edge weight is minimal. each The node acts as a router that maintains the routing table and performs regular routing Routing updates are exchanged, even if routing is not required. A sequence The number is associated with each route or path to the destination to prevent Routing loop.To explain the implementation, consider the following network scenario. In the topology, there are 4 nodes and 3 channels. Node B and C It is a neighbor of node A. Node D is a neighbor of nodes B and C.

4.3 Proposed Algorithm Implementation

The nodes in the network themselves have some information. These data Is the neighbor list (neighboring nodes), channel list (connected channels) And neighbors), the distance between the pair of nodes, and It has a radio. Declares a structured data structure to hold all of these information.Assume that a node has at most three neighbors. If the graph There are only two nodes, but only one node. Number of neighbors It is bound to reduce overhead. On the other hand, actually The node cannot connect to all other nodes of the network. Set as Used to prevent double deception by neighbors. After defining neighbors After setting, the channels will be allocated among them. The road is represented as channel.The function of input source, target and channel ID is Declared as a routing protocol.

4.3.5 Simulation Setup

Simulation is done for 100 times based on 2 criteria. One is by keeping the number of channel fixed and another is by keeping the number of nodes fixed.

Chapter 5

Experimental Results and

Evaluation

In this chapter, use depth first to test the polygonal graph model Search-based routing protocol. Two sets of simulations Here and the simulation results are also compared with the theoretical results And the final complexity of the proposed system is determined.

5.1 Experimental Results of Existing Algorithm

In a multi-radio network, incoming and outgoing are not necessary The channels are the same. Here, channels are allocated according to synchronization items. If the channel is already occupied, set this item to 1, otherwise set to 0. Check this term, the channel has been switched. In this case, the route to success is About 95%.

5.2 Simulation Study of Proposed Model

A custom simulator written in C is used to simulate a polygonal graph. Consider a grid topology with a different number of nodes. The channels available on each node are randomly selected. maximum The number of channels (channel set) on each node varies from 1 to all. of Check the connectivity from the first node to the last node in the graph Draw the percentage of successful routes. Every point The graph is the average of 100 simulations.

5.3 Experimental Result of Existing Model

In this section, the experimental results of the percentage of successful routes Both single radio and multi-radio networks for the multi-edge graph model are shown. In order to discover the route, the routing protocol based on MST is used in this work.the

variation of percentage of successful routes with the number of nodes in the networks in which switching the interface between
the channels is allowed and not allowed respectively. It is observed that there is a success rate of nearly 100% with switching. In without switching case, success rate decrease with the increase of node number.
Comparison of Existing and Proposed Mode In graphical representation is seen that the percentage of successful routes for single radio network varies in proposed model and existing model where nodes number are varying A graph model, while modeling the network correctly should be simple. In the following discussion the complexity of the graph model is compared with the DSDV routing protocol based graph model proposed. In our proposed algorithm the time complexity is O(V+E). Here, V is the number of nodes, E is the number of edges. But in existing algorithm the time complexity is O (n (E log n) C). Here, n is the number of nodes, E is the number of edges and C is the number of channels. BFS provides scalable solution for removing redundant neighbors of each node. Its performance does not degrades with the increase of Nodes. BFS can nd the shortest path but not the MST, in case of BFS the time taken is more but the success rate is more than MST.

5.5 Chapter Summary

In the real world, the network is composed of heterogeneous nodes. some A node may have only one interface, and some nodes may have multiple interfaces. When nodes have multiple interfaces, they can use one interface In this case, for each incoming and outgoing channel, The success rate is similar to a network that can be exchanged. and so, The actual success rate is somewhere between the results shown in the previous sections.

Chapter 6

Conclusion and Future Recommendation

6.1 Conclusion

In this article, the new polygonal graph modeled for multi-hop CRN is Present. The proposed graph model has two-way edges and planes The structure is similar to that in the classical graph model. Use graphical modeling like this Multi-hop CRN supports direct use of traditional routing protocols Used for route discovery. Use a custom simulator to validate the model Built-in C++. Routing protocol based on breadth first search is used simulation. Finally, the time complexity of the model is It demonstrates a comparison with the existing MST-based routing model.

6.2 Future Improvements

For our future work, many people recommended various features to us. However, if we meet the following conditions, our project will be a complete project. Future suggestions are: We will try to integrate machine learning algorithms to reduce Time complexity and success rate of finding routes In this model, only the number of edges is minimized. So further The improvement is to optimize the total cost by keeping the number of edges More, The way the algorithm is developed should ensure the security of the main database The user will not be interrupted.

Bibliography

[1] Mihail L Sichitiu. Wireless mesh networks: opportunities and challenges. In Proceedings of World Wireless Congress, volume 2, page 21, 2005.

[2] Ian F Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. Computer networks, 47(4):445{487, 2005.

[3] Gyanendra Prasad Joshi, Seung Yeob Nam, and Sung Won Kim. Cognitive radio wireless sensor networks: applications, challenges and research trends. Sensors, 13(9):11196{11228, 2013.

[4] Qing Zhao and Ananthram Swami. A survey of dynamic spectrum access: Signal processing and networking perspectives. Technical report, CALIFORNIA UNIV DAVIS DEPT OF ELECTRICAL AND COMPUTER ENGINEERING, 2007.

[5] Ejaz Ahmed, Muhammad Shiraz, and Abdullah Gani. Spectrum-aware distributed channel assignment for cognitive radio wireless mesh networks. Malaysian Journal of Computer Science, 26(3):232{250, 2013.

[6] Nafees Mansoor, AKM Muzahidul Islam, Mahdi Zareei, Sabariah Baharun, Toshio Wakabayashi, and Shozo Komaki. Cognitive radio ad-hoc network architectures: a survey. Wireless Personal Communications, 81(3):1117{1142, 2015.

[7] Yogesh R Kondareddy and Prathima Agrawal. A graph based routing algorithm for multi-hop cognitive radio networks. In Proceedings of the 4th Annual International Conference on Wireless Internet, page 63. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2008.

[8] Ian F Akyildiz, Won-Yeol Lee, Mehmet C Vuran, and Shantidev Mohanty. A survey on spectrum management in cognitive radio networks. IEEE Commun. Mag. 46(4), pages 40{48, 2008.

[9] Cesar Hernandez, Camila Salgado, H L opez, and Enrique Rodr guez-Colina. Multivariable algorithm for dynamic channel selection in cognitive radio networks. EURASIP Journal on Wireless Communications and Networking, 2015(1):216, 2015.

[10] Alberto Rabbachin, Tony QS Quek, and Moe Z Win. Statistical modeling of cognitive network interference. In 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, pages 1{6. IEEE, 2010.

[11] Yasir Saleem, Adnan Bashir, Ejaz Ahmed, Junaid Qadir, and Adeel Baig. Spectrum-aware dynamic channel assignment in cognitive radio networks. In 2012 International Conference on Emerging Technologies, pages 1{6. IEEE, 2012.

[12] Anand Prabhu Subramanian, Himanshu Gupta, Samir R Das, and Jing Cao. Minimum interference channel assignment in multiradio wireless mesh networks. IEEE transactions on mobile computing, 7(12):1459{1473, 2008.

[13] A Hamed Mohsenian Rad and Vincent WS Wong. Joint optimal channel assignment and congestion control for multi-channel wireless mesh networks. In 2006 IEEE International Conference on Communications, volume 5, pages 1984{1989. IEEE, 2006.

[14] Vinay Kapse and Urmila Shrawankar. Interference-aware channel assignment for maximizing throughput in wmn. arXiv preprint arXiv:1305.2838, 2013.
[15] A. Singh and K. Singh. Load balancing based channel assignment in multicast wireless
mesh network. pages 865{869, March 2014.
[16] S. Sultana and A. Asaduzzaman. A minimum spanning tree based routing protocol for multi-hop and multi-channel cognitive radio. pages 206{211, June 2018.

Appendix

Source Code for BFS based routing

```
#include <bits/stdc++.h>
using namespace std;
const int inf = 1e9+7, SIZE = 105;
int solve(int source, int destination,int channelID,int isSwithingAllowd);
void modifiedDSDV(int channelID, int nodeNumber, int isSwithingAllowed);
void generateGraph(int totalNode,int channel);
void simulateMySystem(void);
void clearAll(void);
void clearRoutingTable(int nodeNumber,int isSwithingAllowed);
void printGraph(int nodeNumber);
struct node{
    vector <int> neighbourList, channelList[SIZE], distance;
    int hasConnectionWith[SIZE][12];
}Network[SIZE];

struct node2
{
        int from, to, channel, cost;
        node2(int from, int to, int channel, int cost)
        {
                this->from = from;
                this->to = to;
                this->channel = channel;
                this->cost = cost;
        }
        bool operator <(const node2 &p)const {return p.cost<cost;};
};
struct road{
        int from,to,channel;
        road(int from,int to,int channel)
```

```
	{
		this->from = from;
		this->to = to;
		this->channel = channel;
	}
};

vector <road> Graph;
unordered_set <int> channelsOfSource;

int totalNode,totalCost,finalCost;
int routingTable[101][101][11];
int isConnected[SIZE];
int hasRoad[SIZE][SIZE];
void generateGraph(int totalNode,int channel)
{
	int n,i,j,k,neighbourNo,c,d,temp,source,destination;
		memset(hasRoad,0,sizeof(hasRoad));
		for(i=1;i<=totalNode;i++)
		{
			if(totalNode>2)
				neighbourNo = 1+rand()%min(3,totalNode-1);
			else
				neighbourNo = 1;
			set<int>neighbourSet;
			for(j=1;j<=totalNode;j++)
			{
				if(hasRoad[i][j]==1)
				{
					neighbourNo--;
					neighbourNo = max(0,neighbourNo);

				}
			}
			while(neighbourSet.size()<neighbourNo)
			{
				n= 1+rand()%totalNode;
				if(hasRoad[i][n]==0&&n!=i)
				{
					neighbourSet.insert(n);
					hasRoad[i][n]=hasRoad[n][i]=1;
				}

			}
			//printf("Neighbour Set generation end\n");
			set<int>::iterator itt;
			for(itt=neighbourSet.begin();itt!=neighbourSet.end();++itt)
			{
```

```
                    d=1+rand()%1007;
                    Network[i].neighbourList.push_back(*itt);
                    Network[*itt].neighbourList.push_back(i);
                    Network[i].distance.push_back(d);
                    Network[*itt].distance.push_back(d);
                    set<int>channelSet;
                    int channelNumber = 1+rand()%channel;
                    while(channelSet.size()!=channelNumber)
                    {
                        channelSet.insert(1+rand()%10);
                    }
                    set<int>::iterator itt2;
                    for(itt2=channelSet.begin();itt2!=channelSet.end();++itt2)
                    {
                        Network[i].channelList[*itt].push_back(*itt2);
                        Network[*itt].channelList[i].push_back(*itt2);
                        Network[i].hasConnectionWith[*itt][*itt2]=1;
                        Network[*itt].hasConnectionWith[i][*itt2]=1;
                    }
                }
            }

}


void simulateMySystem(void)
{
    int i, j, switching, withSwithcing, withoutSwitching, n, dsdvWithoutSwitching,
dsdvWithSwitching;
    int nodeNumber = 10;
    for(int channelNumber=2;channelNumber<=10;channelNumber++)
    {
            withSwithcing=0;
            withoutSwitching=0;
            dsdvWithoutSwitching=0;
            dsdvWithSwitching=0;
            for(int numberOfSimulation =1;
numberOfSimulation<=100;numberOfSimulation++)
            {

                    clearAll();

                    generateGraph(nodeNumber,channelNumber);
                    n = Network[1].neighbourList[0];
                      withSwithcing+=solve(1,nodeNumber,Network[1].channelList[n][0],1);

                      channelsOfSource.clear();
```

```
                for(i=0;i<Network[1].neighbourList.size();i++)
                {
                    n = Network[1].neighbourList[i];
                  for(j=0;j<Network[1].channelList[n].size();j++)
                  {
                        channelsOfSource.insert(Network[1].channelList[n][j]);
                    }
                }
              unordered_set<int>::iterator itt;

                for(itt=channelsOfSource.begin();itt!=channelsOfSource.end();++itt)
              {
                    int isPossible = solve(1,nodeNumber,*itt,0);
                    if(isPossible==1)
                      {
                          withoutSwitching++;
                        break;
                    }
                }
                //---------------- code form MST end ------------------------------

                //---------------- code for modified DSDV begin --------------------
                clearRoutingTable(nodeNumber,1);
                modifiedDSDV(1,nodeNumber,1); // with switching

                if(routingTable[1][nodeNumber][1]!=inf)
                    dsdvWithSwitching++;
                for(int i=1;i<=10;i++)
                {
                    clearRoutingTable(nodeNumber,0);
                    modifiedDSDV(i,nodeNumber,0);
                    if(routingTable[1][nodeNumber][i]!=inf)
                    {

                        dsdvWithoutSwitching++;
                        break;
                    }
                }
                //---------------- code for modified DSDV end ----------------------

            }
            printf("For %d nodes and %d channel: %d %d
| %d %d\n",nodeNumber,channelNumber,withoutSwitching,withSwithcing,dsdvWithoutSwitchin
g,dsdvWithSwitching);
        }
}
```

```
void modifiedDSDV(int cID, int nodeNumber, int isSwithingAllowed)
{
	int to,cost,chnl;
	for(int itteration=1;itteration<=nodeNumber;itteration++)
	{
		for(int i=1;i<=nodeNumber;i++)
		{
			for(int j=0;j<Network[i].neighbourList.size();j++)
			{
				to = Network[i].neighbourList[j];
				cost =Network[i].distance[j];
				for(int k=1;k<=nodeNumber;k++)
				{

					if(isSwithingAllowed==1)
					{
						routingTable[i][k][1]=min(routingTable[i][k][1],
routingTable[i][to][1]+routingTable[to][k][1]);
					}
					else if(Network[i].hasConnectionWith[to][cID]==1)
					{
						routingTable[i][k][cID]=min(routingTable[i][k][cID],
routingTable[i][to][cID]+routingTable[to][k][cID]);
					}
				}
			}
		}
	}
}

void printGraph(int nodeNumber)
{
	for(int i=1;i<=nodeNumber;i++)
	{
		for(int j=0;j<Network[i].neighbourList.size();j++)
		{
			printf("%d -> %d %d c =",i,Network[i].neighbourList[j],Network[i].distance[j]);
			for(int k=0;k<Network[i].channelList[Network[i].neighbourList[j]].size();k++)
				printf(" %d",Network[i].channelList[Network[i].neighbourList[j] ][k]);
			printf("\n");
		}
	}
}

void clearAll(void)
{
	for(int i=1;i<=101;i++)
	{
```

```
            Network[i].neighbourList.clear();
            Network[i].distance.clear();
            memset(Network[i].hasConnectionWith,0,sizeof(Network[i].hasConnectionWith));
            for(int j=0;j<104;j++)
            {
                Network[i].channelList[j].clear();
            }

        }
}

void clearRoutingTable(int nodeNumber,int isSwithingAllowed)
{
        for(int i=1;i<=nodeNumber;i++)
        {
            for(int j=i;j<=nodeNumber;j++)
            {
                for(int c=0;c<=10;c++)
                {
                    if(i==j)
                    {
                        routingTable[i][i][c]=0;
                    }
                    else
                    {
                        routingTable[i][j][c]=inf;
                        routingTable[j][i][c]=inf;
                    }
                }
            }
        }
        if(isSwithingAllowed==1)
        {
            for(int i=1;i<=nodeNumber;i++)
            {
                for(int j=0;j<Network[i].neighbourList.size();j++)
                {
                    int n=Network[i].neighbourList[j];
                    for(int c=0;c<=10;c++)
                    {
                        routingTable[i][n][c]=Network[i].distance[j];
                    }
                }
            }
        }
        else
        {
            for(int i=1;i<=nodeNumber;i++)
```

```
            {
                for(int j=0;j<Network[i].neighbourList.size();j++)
                {
                    int n=Network[i].neighbourList[j];
                    for(int c=0;c<Network[i].channelList[n].size();c++)
                    {
                        int ch =Network[i].channelList[n][c];
                        routingTable[i][n][ch]=Network[i].distance[j];
                        routingTable[n][i][ch]=Network[i].distance[j];
                    }
                }
            }
        }
}


int solve(int source, int destination,int channelID,int isSwithingAllowed)
{
        /*
        Depth First Search With Stack
        */
        int isPossible = 0;
        memset(isConnected, 0, sizeof isConnected);
        isConnected[source] = 1;
        Graph.clear();
        stack <node2> myStack;
        for(int i=0;i<Network[source].neighbourList.size();i++)
        {
            int to = Network[source].neighbourList[i];
            int cost =Network[source].distance[i];
            int chnl = Network[source].channelList[to][0];
            if(isSwithingAllowed==1)
            {
               myStack.push(node2(source,to,chnl,cost));
            }
            else if(Network[source].hasConnectionWith[to][channelID]==1)
            {
                myStack.push(node2(source,to,channelID,cost));
            }
        }
        while(myStack.empty() != true)
        {

            node2 temp = myStack.top();
            myStack.pop();
            if(isConnected[temp.to]==0)
            {
```

```
                        Graph.push_back(road(temp.from,temp.to,temp.channel));
                        if(temp.to==destination)
                                isPossible =1;
                        isConnected[temp.to]=1;
                        totalCost+=temp.cost;
                        for(int i=0;i<Network[temp.to].neighbourList.size();i++)
                        {
                                int to = Network[temp.to].neighbourList[i];
                                int cost =Network[temp.to].distance[i];
                                int chnl = Network[temp.to].channelList[to][0];
                                if(isSwithingAllowed==1)
                                {
                                        myStack.push(node2(temp.to,to,chnl,cost));
                                }
                                else if(Network[temp.to].hasConnectionWith[to][channelID]==1)
                                {
                                        myStack.push(node2(temp.to,to,channelID,cost));
                                }
                        }
                }
        }
        if(isPossible==0)
                totalCost = 9999999;
        return isPossible;
}
int main()
{
        srand(time(NULL));
        simulateMySystem();

        return 0;
}
```