Routing Protocols in Cognitive Radio Networks

Written by: Jeremy Mack

Professor: Dr. Ibnkahla

Date Submitted: March 5th, 2013
# Table of Contents

1.0  Introduction .................................................................................................................. 4

2.0 Routing Problems in Cognitive Radio Networks .......................................................... 5

3.0 Classifications in Cognitive Radio Networks ............................................................... 8

   3.1 Spectrum Knowledge .................................................................................................. 8

      3.1.1 Centralized Network ......................................................................................... 8

      3.1.2 Distributed Network ........................................................................................ 8

   3.2 Primary User Activity ............................................................................................... 9

      3.2.1 Static Cognitive Multi-Hop Networks ................................................................. 9

      3.2.2 Dynamic Cognitive Multi-Hop Networks ............................................................ 9

      3.2.3 Opportunistic Cognitive Multi-hop Networks ..................................................... 10

4.0 Centralized and Basic Distributed Protocols............................................................... 10

   4.1 Centralized Protocols ............................................................................................... 10

      4.1.1 Color Graph Routing ....................................................................................... 10

      4.1.2 Conflict Graph Routing .................................................................................. 11

      4.1.3 Optimization Approach ................................................................................... 12

      4.1.4 Layered Graph Routing ................................................................................... 13

   4.2 Distributed Protocols .............................................................................................. 17

      4.2.1 Control Information ......................................................................................... 17

      4.2.2 Source or Destination Based Routing ............................................................... 18

5.0 Geographical Protocol for Dynamic Networks .......................................................... 18

   5.1 Initial Route Setup .................................................................................................... 19

      5.1.1 Greedy Forwarding .......................................................................................... 20

      5.1.2 PU avoidance .................................................................................................. 21

      5.1.3 Joint Channel-Path optimization ..................................................................... 21

   5.2 Routing Algorithm Mathematical Formulation ....................................................... 21

      5.2.1 Route Setup ..................................................................................................... 21

      5.2.2 Route Enhancement ....................................................................................... 22

      5.2.3 Route Maintenance ....................................................................................... 23

      5.2.4 Simulations ..................................................................................................... 23

      5.2.5 Conclusions and Opinions ............................................................................. 25

March 5, 2013
6.0 Opportunistic Cognitive Multi-hop Protocol

6.1 Protocol Overview

6.2 Performance Metrics

6.2.1 Relay Distance Advancement

6.2.2 Per Hop Delay

6.3 Probabilistic Formulations

6.3.1 Channel Sensing

6.3.2 Relay Selection

6.3.3 Data Transmission

6.4 Joint Channel and Relay Selection Extension

6.4.1 The OCR Metric

6.5 Heuristic Algorithm

6.5.1 Delay Simulation of Heuristic Algorithm

6.6 Simulations

6.6.1 Primary User Activates

6.6.2 Multi-User Diversity

6.6.3 Effectiveness in Routing Metric

6.7 Comments and Conclusions

7.0 Summary of Protocols

8.0 Conclusion

9.0 References

10.0 Appendix

10.1 Layered Graph Matlab Code

10.2 Summary Tables
Abstract - Cognitive radio is an important technology for supporting dynamic spectrum access and is widely regarded as one of the most promising technologies in wireless communications. The majority of the literature in this area focuses on creating efficient spectrum sensing and sharing algorithms at a single node level. Recently, authors have started extending the research and began developing algorithms for multi-hop networks. This is required due to the rise of Ad-Hoc and Mesh networks in real applications. When working with any sort of multi-hop network, routing is a very important aspect. Traditional protocols for multi-hop networks will not work in most cognitive radio networks due to the use of multiple channels and the introduction of primary users. Authors have also shown that the routing protocol should be coupled with the spectrum sensing module to allow for the best results. This report is a survey of current literature in regards to routing protocols and challenges in cognitive multi-hop radio networks. The obstacles that arise with routing packets are described as well as some classifications of protocols and networks. Finally, the report describes and compares a number of protocols and provides suggestions and potential future work.

Index Terms: Cognitive Radio Networks, Routing, Multi-hop transmission, Dynamic spectrum access

1.0 Introduction

Research in cognitive radio networks was effectively launched in 2002 after a report was published by the Federal Communication Commission (FCC) [1]. The results from this document showed that at any time and almost anywhere less than 10% of the available spectrum was utilized efficiently. It also showed that some parts of the spectrum oscillated from 15% to 85% usage with high variance in time. This report showed the research community the need for a new generation of smart programming radios that can adapt to the changing network conditions and utilize the spectrum holes that appear in unused frequency bands.

To address this problem, the notion of dynamic spectrum access (DSA) was proposed. DSA allowed users to use licensed frequency bands opportunistically and in a non-interfering manner. It also introduced a type of economical aspect of networking that allowed the owners of the licensed bands to rent portions of their bandwidth to users for a price.

To implement DSA, software defined radios began development to allow engineers to program their networking devices to make decisions based on the network condition. [2] Software designed radio resulted in thousands of lines of codes that required a lot of debugging and upkeep in real world applications. The solution to this problem was the cognitive radio, which implements a cognitive engine that allows the device to adapt on its own.

There are two types of users that operated in cognitive radios networks, Primary users and Secondary users. Primary users have the priority of the spectrum as they are the owners the spectrum, while secondary users are just renting the bandwidth so they can only access the spectrum when the primary user is not using it. Primary users do not technically need cognitive radios, but secondary user must have
them in order to be able to scan the spectrum to ensure they will not be interfering with any primary users.

Cognitive radio networks have been a popular topic in literature over the past 10 years, but the majority of the authors have focused on the physical and mac layer problem such as spectrum sensing, spectrum decision and sharing. This is due to the single hop network assumption that the majority of the authors make. However it is important to integrate cognitive radio’s into multi-hop networks due to the increase in use of Ad-Hoc and Mesh networks in real life applications. The major difference between single hop and multi-hop networks is routing. Routing is the process of selecting a path in a network in which to send the packets. Since all users will not be in range of each other, intermediate notes must be used as relays to pass the information from source to destination.

Routing in multi-hop cognitive radio networks must consider the creation and maintenance of wireless multi-hop paths among Secondary users. For each hop along the path the user must decide which relay node to use and the spectrum channel they will transmit on. These considerations are similar to routing in traditional multichannel, multi-hop ad hoc networks however with the added complexity of dealing with primary users and switching frequency bands to utilize the spectrum effectively.

The rest of the report is organized as follows. Section 2 describes the routing problems in cognitive ratio network and some classifications of networks are described in Section 3. In Section 4, an overview is provided on some basic centralized and distributed solutions and an in depth look at a layered graph protocol. Section 5 describes a geographical algorithm “SEARCH” which is an extension of a basic distributed protocol. Section 6 provides an extension of the “SEARCH” protocol that allows it to operate on highly dynamic networks. Section 7 will provide a summary of the protocols and finally Section 8 will provide some conclusions on the presented material and future work.

2.0 Routing Problems in Cognitive Radio Networks
Routing protocols in traditional multi-hop networks must overcome a large number of challenges to provide the best results for the network. Protocol designers must keep the following challenges in mind when designing their protocols [3].

**Energy Efficiency:** Many multi-hop networks do not have a central fixed infrastructure to provide unlimited power to the nodes. Thus, the protocol must keep in mind its energy consumption to ensure the algorithm does not deplete the user’s power to rapidly.

**Network Topology Changes:** If a node fails, let’s say from power depletion or from technical difficulties, the routing protocol must be able to detect and adapt to this failure and determine a new route. During this process it must consider the latency introduced and trying to reduce the downtime to a minimum.

**Scalability:** The algorithm should operate just as well when the number of users in the network increases. This is a very important factor that many authors do not consider when developing their protocols.
Mobility: Some nodes in a network may be constantly moving, thus the routing protocol must be able to adapt to these moving nodes, and detect broken routes and provide replacement paths.

In cognitive radio networks the previous four problems must considered as well as the following challenges [4] that are introduced by the introduction of primary users into the networks:

Spectrum Awareness: Due to restrictions created by primary users, the nodes must always be aware of their local spectrum to ensure that they are not affecting a primary user. Due to the multi-channel aspect of cognitive radio networks, each node may have a different view of the spectrum even if they are neighbors. Thus, some form of communication must be implemented to ensure that when two communication communicated they will choose a channel that is appropriate for both of them. To ensure awareness of the entire network these three scenarios are implemented:

a. The information on the spectrum availability is provided to the nodes by a central all-knowing entity that monitors the entire network and the spectrum bands available.
b. The spectrum availability is gathered locally by the nodes and distributed to all nearby nodes that require this information.
c. A combination of a and b.
**Quality Routes:** Route quality is a major consideration in traditional multi-hop networks, but in cognitive radio networks it has to be redefined with additional variables. The topology of the network is highly influenced by the primary user’s behavior, thus the traditional way of measuring the quality of a route (bandwidth, throughput, delay, energy efficiency, fairness) must be coupled with new measures like path stability, spectrum availability/efficiency and primary user presence.

**Route Maintenance:** In cognitive radio networks, primary users come and go quite often so there is a high probability that during a transmission the route will break. Thus, the protocol must be able to quickly detect a break in the route, stop transmission and provide a replacement path. A lot of protocols deal with this challenge by storing a set of backup routes that they are able to quickly switch to if the route fails.

**Complexity:** Unlike traditional multi-hop networks the routing protocol must be coupled with the spectrum sensing, this results in more complexity in the algorithm. Higher complexity could affects some of the other challenges such as power efficiency or cause a long convergence time.

As described in the complexity problem above CRN routing must be integrated into the spectrum cycle, as shown in Figure 1, and highly coupled with the spectrum sensing aspect to ensure that the most efficient route is chosen. To show this I will use an example, consider the small network in Figure 2. In this network there are two available channels, red and blue, the triangles are primary users and the circles are secondary users. If A wants to transmit to E it has two choices, either A->C->E or A->B->D->E. The times it takes for a packet to travel that path can be seen below and it takes $\gamma$ time to change channels.

\[
\gamma = 4 \text{ ms} \\
A \rightarrow C \rightarrow E = 10\text{ms} \\
A \rightarrow B \rightarrow D \rightarrow E = 12 \text{ms}
\]

If the routing is separate from the spectrum cycle, then it will choose A->C->E since it has the lower travel time. However, this does not consider that it must change channels twice to get to E, thus the total time will actually be 18ms. In the case when the routing is coupled with the spectrum cycle, the routing protocol knows the location and channel of the nearby primary users and knows that A->C->E may be shorted in hops/distance but it will cause a larger delay then taking the A->B->D->E path and only having to change channels once.

Due to the large number of challenges, no protocol is able to optimize them all. Thus, trade off occurs and this is why there are over 1000 papers describing different protocols for cognitive multi-hop radio networks. This report will give an overview of a few protocols to show the wide variety of solutions that have been developed to try and solve these challenging problems.
3.0 Classifications in Cognitive Radio Networks

As mentioned in Section 2.0, there are a number of challenges that must be considered when designing a routing protocol for cognitive radio networks. Due to the large number of problems, there cannot exist a general routing protocol that is applicable to all networks, there is always some form of trade off. In literature, many authors develop protocols that make certain assumptions and focus on solving a particular problem. This section will go over some classifications that have arisen from the authors assumptions in literature.

3.1 Spectrum Knowledge

Spectrum knowledge is the amount of knowledge that each secondary user has available when it makes it routing decision. There are two main classifications in this area, centralized and distributed. Figure 3 shows a flow chart of the main types of spectrum knowledge classifications and some examples of the protocols that are based on them.

3.1.1 Centralized Network

A centralized network, also called a full spectrum knowledge network, assumes that each secondary user has full spectrum knowledge of the entire network. This is accomplished by having an all-knowing central devise that's sole purpose is to monitor the available spectrum and relay the information to the users in the network. These protocols usually involve more theoretical mathematics since the centralized devices usually assume unlimited computational power. They are usually designed for very specialized networks their results usually provide upper bounds and benchmarks for other algorithms. The authors usually assume they are working with static networks. This is due to the complexity of the algorithms, so their convergence time will not be fast enough to operate on dynamic networks. As I mentioned before these networks are usually designed for very specialized scenarios, however, I know firsthand that they do exist in real world applications. My current research project is working with a centralized network that is assumed to be static and I am actually using a modified version of one of the centralized protocols described in Section 4.

3.1.2 Distributed Network

Distributed networks, or local spectrum knowledge networks, assume that the spectrum knowledge is determined by each secondary user locally and then forwarded to their neighbors. This requires the routing protocol to be able to determine the appropriate routing paths while also measuring the spectrum and identifying potential bands that the user is able to transmit on.
The measuring of the local spectrum can be accomplished with either a more deterministic approach or dealing with a probabilistic nature. In a deterministic approach, the nodes measure the spectrum and make their decisions on the current spectrum band situation. In a probabilistic approach, the secondary users periodically measure the bands and generate statistics. Using these statistics, the users can make better decisions based on the past nature of a particular band. For example, if a particular band has shown that the primary user’s on/off time is sporadic, then it is probably not the best band to choose for a long transmission.

Distributed approaches are usually harder to implement and result in worse results than a centralized solution. The worse results are due to the increased overhead from distributing the spectrum information. However, they are more applicable to general real-world networks since they work with dynamic networks and thus will be the primary focus of my report.

3.2 Primary User Activity

Primary users have a large impact on the routing in cognitive radio networks, thus the authors of “Multi-hop Cognitive Radio Networks: To Route or Not to Route” [5] classified routing protocols based on their primary user activity into three categories: Static, Dynamic and Opportunistic (Highly Dynamic). Figure 4 shows how the networks are classified in reference to the average primary user idle time.

3.2.1 Static Cognitive Multi-Hop Networks

In this classification, the primary users’ bands are available for a duration that, on average, exceed the communication time of the secondary users. The secondary users usually make the assumption that an available frequency band is a permanent resource. The only difference between a traditional multi-hop network and a static cognitive network is the use of different spectrum bands, and the physical capability of transmitting simultaneously over several frequency bands.

In most cases, a traditional multi-hop protocol can be slightly altered and used. Typical practical examples of these types of networks are projects that operate with satellite or the analog television bands. Section 4.1 discusses a protocol that is designed for a static network.

3.2.2 Dynamic Cognitive Multi-Hop Networks

In dynamic networks, the primary users are assumed to be idle for minutes at a time. In these networks, you cannot assume the frequency band as a permanent resource, so you cannot use traditional multi-hop protocols. The protocol must incorporate route stability, exchanging control information and channel synchronization into its algorithm.

Path stability can be improved by incorporating the routing protocol with the spectrum cycle as described in Section 2. This can be further extended by using past channel statistics and favoring stable spectrum bands (less dynamic) over unstable ones.
The computation of the algorithms must be quick in this type of network to allow the protocol to adapt to the dynamic changes, thus creating a restraint on the complexity of the algorithm. Usually, a route is created for the whole flow and routing tables are generated to let the secondary users know which user they have to transmit to next. The disadvantage of routing tables is they must be periodically updated, which generates a large amount of overhead in the network. However, due to the non-highly dynamic nature of these types of networks the overhead is reasonable. Section 5 discusses a protocol that is designed for a dynamic network.

3.2.3 Opportunistic Cognitive Multi-hop Networks
In opportunistic networks, the time in which the spectrum is available is less than the average communication duration. Thus an end to end path will not work since each packet may experience different network properties due to the highly dynamic nature. Thus, it is usually better to use an opportunistic solution, in which every packet is sent and forwarded over opportunistically available channels. This means, that each packet makes its own decision on which channel and relay to choose for each hop. Having the algorithm deal with each packet, or a group of packets can reduce the complexity of establishing end-to-end routes and increase the efficiency of the proposed solutions.

The choice of the channel to forward your data is very important in these types of networks since a packet may operate on more than one channel during its trip from source to destination. Channel history can be a good measure to determine which channel to choose at each hop. For example, if a band has been unreliable for a long period of time it should be avoided even though the primary user is currently not there.

Although opportunistic protocols are the most flexible, they have the largest number of challenges, especially in terms of reducing complexity to allow the algorithm to operate fast enough. Section 6 discuses a protocol that is designed for an opportunistic network.

4.0 Centralized and Basic Distributed Protocols

4.1 Centralized Protocols
Centralized protocols assume that each secondary user has spectrum knowledge of the entire network. This knowledge is usually determined by a central device and usually the routing algorithm is performed on the central device due to its computation power. These methods are usually more mathematical and usually are less practical. I will first give a basic overview of three basic centralized solutions and then provide a more detailed description of a more complex protocol, the layer graph [6].

4.1.1 Color Graph Routing
The coloring graph method [7] uses graph theory to model the network and all the channels available. This method can be split up into two parts:

- Creation of the graph
- Route calculation
When creating the graph the secondary users are represented by the vertices of the graph and if two users want to communicate they can choose one of M colored edges, where M is the number of available channels. The color of the edge corresponds to the carrier frequency that will be used. The network can be represented using Equation 1.

\[ G_c = (N_c, E_c) \] (1)

Where \( N_c \) is the vertex set and \( E_c \) is the edge set. Each of the vertices can be connected by a number of different edges as seen in Figure 5. After the graph has been created it begins to generate routes. The goal is to generate all the routes needed in the network while trying to minimize the number of channels used in the network while also ensuring the interference is at a minimum.

![Figure 5: Color graph protocol](image)

This algorithm is very basic and will only work on static networks with primary users that have low mobility. The author did not consider scalability when designing the protocol, since as the number of nodes increase, the number of route also increase drastically. Since the author searches for the best route exhaustively, the convergence time has a high probability of taking so long the graph will change by the time it completes.

It also does not have any route maintenance, thus if a route breaks or the network changes, it will have to generate another graph and run the routing portion of the algorithm again. This causes a large amount of overhead when the network conditions change. This algorithm could be extended easily by adding weights to each edge, thus allowing the graph to more closely model the network.

### 4.1.2 Conflict Graph Routing

The conflict graph method [8] is similar to the color graph protocol however it decouples choosing channels and routing into two separate parts. The graph aspect is used for selecting the channel and the routing is done with an unspecified basic routing algorithm. The protocol considers all the available routes between a source and a destination, and for each route it considers all the available channels for each relay. The best combination is chosen for the route.

Unlike a color graph an edge is drawn between two vertices if the two users cannot be active at the same time. The graph can be represented by Equation 2:

\[ G_F = (N_F, E_F) \] (2)
Where $N_F$ represents the set of vertices and $E_F$ represent the set of edges between the vertices as seen in Figure 6. The graph is used to determine the channels that will provide no conflict between nearby secondary users. Once the channels are determined, then the routing is done with any basic routing protocol. This is NP-hard, which means that it either requires a large amount of computational power or it will take a long time to converge.

![Figure 6: Conflict graph protocol](image)

This algorithm is similar to the color graph but I believe it will have a shorter computation time than the previous method since there will be fewer edges on the map. However, since the author is still searching the route exhaustively the convergence time may still be a problem. The second problem I see is that the algorithm uses a basic routing protocol, so it will not account for primary users or include route maintenance. Thus, it really is not made for cognitive radio networks as the author described it as. I believed both these protocol will need a lot of work to be implemented into a real work network.

### 4.1.3 Optimization Approach

Another centralized solution is to take a more mathematic view of the problem, other than modeling it as a graph. Hou et Al modeled the routing problem as a mixed integer optimization problem in [9]. The objective of the protocol is to maximize the spectrum reuse factor throughout the network. A number of restrictions are imposed on the problem to allow it to model a cognitive radio network:

1) **Link Capacity**: This restriction forces the total traffic flow in each link to not exceed the capacity of the link. Shannon law is used to define the link capacity, as shown in Equation 3 for the link $(i,j)$

$$c_{ij}^L = W^L \log_2 \left(1 + \frac{g_{ij}Q}{\eta}\right)$$

Where $W^L$ is the bandwidth of sub band $L$, $g_{ij}$ is the propagation gain of link $(i,j)$, $Q$ is the power spectral density in transmission and $\eta$ is Guassian white noise.
2) **Interference**: The interference is modeled using an interference range $R_T$, defined in equation 4:

$$R_T = \left(\frac{Q_T}{Q_T^1}\right)^{\frac{1}{\eta}}$$  \hspace{1cm} (4)

where $Q_T$ is the threshold power spectral density guaranteeing correct reception. Thus, if two users fall in the range of each other’s $R_T$, they cannot transmit on the same channel.

3) **Routing**: Routing is managed by using flow balancing constraints for each node that restricts the incoming flow to be equal to the outgoing flow. This allows the creation of split routing paths, which allows for added robustness to the network but this technique is harder to implement in a real network.

The protocol determines its routes with the following algorithm:

1. Set up and solve a relaxed linear programming version of the original problem to achieve a lower bound
2. Sort the assignment variables in descending order
3. Set the largest variables in the list to 1, and set the remaining variables to 0
4. Solve the new linear programming problem until all variables are 1

The author presented a technique to speed up the algorithm by fixing groups of variables instead of just one at a time to improve the scalability. Even with this technique it would still take a long time to generate routes. It may be beneficial to model the network as a binary mixed integer optimization problem to decrease convergence time and complexity.

**4.1.4 Layered Graph Routing**

This layered graph protocol [6] is an extension of a basic weight graph that is used in networks that operate on a single channel. The goal of the protocol is to maximize the network capacity while minimizing the interference from neighboring nodes. To accomplish this the protocol generates a layered graph, and then computes routing paths to maximize the network connectivity and provide diverse channel selection to prevent interference from two adjacent hops in a path.

4.1.4.1 Creation of Layered Graph

Each layer of the graph represents a corresponding frequency band that can be used by the nodes. Let $M$ denote the number of frequency bands (layer) available in the network and $N$ be the number of nodes in the network. Figure 7 shows a simple networks with an $M=2$ and $N=4$. In this example, each node in the network is represented by two sub nodes on different layers (i.e. Node 3 has sub node $3_1$ and $3_2$). A sub node can be either active or inactive. If the node is operating on frequency 1, then the sub node on frequency 1 is an active sub node, and all the other sub nodes would be inactive sub nodes.
An edge in the graph is represented by a line connected two sub nodes. There are three classifications of edges in the network

1) **Horizontal Edges**: Horizontal edges connect sub nodes that are on the same layer. A horizontal edge represents two nodes that can transmit to each other on a particular frequency.

2) **Vertical Edges**: Vertical edges connect nodes to their own sub nodes on different layers. A vertical edge is created between a node on layer “Y” and its sub nodes on layer “X” if the nodes can operate on either channel “Y” or “X”.

3) **Internal Edges**: Internal edges connect a sub node to its auxiliary node. Auxiliary nodes are used to allow nodes to be connected to layer/channel that are not directly above or below them.

### 4.1.4.2 Routing and Interface Assignment

The basic idea behind the routing protocol is to traverse the layered graph finding the route with the smallest total sum of the edge weights. Thus, the edge weights are very important in ensuring the protocol is optimum.

The vertical costs are suggested to be negative to try and have as many channel changes as possible to reduce interference in the network. The horizontal edges are positive and are relevant to the “effective” capacity of the edges. So their weight depends on the number of nodes using this channel and the traffic volumes currently being transmitted through the node.

The routing algorithm operates as follows:

1) The graph is generated using the current network situation.
2) Choose a path to reduce the cost of trip
3) Determine which channel to operate each connection between nodes to reduce interference

The author does not describe any sort of route maintenance other than generating a whole new graph. This results in a potential waste of resources since there may only be a few changes in the network, but a whole new graph has to be created. Therefore, a possible extension to this algorithm would be to have the nodes notify the central server when their situation has changed so it can update a smaller section of the graph.

### 4.1.4.3 Simulations

Simulations were completed using Matlab with two networks, one with 15 nodes and another with 30 nodes. The nodes were placed on a Cartesian grid and their X and Y coordinates were uniformly distributed between [0,100]. It assumes there are six available frequency bands each with a bandwidth of 10 Mbps. The number of available channels for each node is randomly selected between 1 and 3. The randomness in the number of channels for each node is supposed to represent primary users acting in a particular area. However this does not make sense to me since you could have two users very close to each other, and one is experiencing a primary user while the other isn’t. This cannot occur in a real network, so the author should consider a better way of choosing the available channels. CSMA/CD is
used on the mac layer and the node waits between 0 to 10 full packets transmission times between transmissions. The cost of edges is shown in Table 1.

Table 1: Edge Weights [6]

<table>
<thead>
<tr>
<th>Type of Edge</th>
<th>Weight</th>
<th>Extra Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>10</td>
<td>Only at start of network</td>
</tr>
<tr>
<td>Vertical</td>
<td>-10</td>
<td></td>
</tr>
<tr>
<td>Internal</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

Whenever a path is completed, the horizontal edge weights in that path increase by one. The internal edge is set reasonably large to prevent a path from leaving its layer and going back to the same layer.

The throughput of the 15 node network is shown in Figure 8, with a 95% confidence interval computed from 10 repeated experiments using different seeds. The confidence interval is large because the node position and radio transmission radius were randomly generated at the start of each iteration. The four graphs represent different transmission ranges and the protocol is compared to a sequential interference assignment (SeqAssign) method. SeqAssign assigns the nodes to the channels in descending order of the number of neighbors that the channel can reach.

Figure 8: Throughput vs. Load, altering the transmission radius, 15 Users [6]
The throughput results show that the proposed algorithm always works better than the SeqAssign method, due to its optimization in the path computation and interference management. As the load increases the throughput gap increased between the two algorithms, this is because the proposed algorithm selects different channels on adjacent hops to effectively avoid interference, thus results in high channel utilization. SeqAssign, does not have any mechanism to select different channels on adjacent hops, thus its throughput is very poor at high load.

Figure 9 shows one of the 30 node network simulations with a transmission range of [0.1,0.3]. From this graph I noticed how poor the scalability of the algorithm is, which will be discussed in the next section.

4.1.4.4 Comments
I believe the layered graph protocol has a number of problems with it, but its largest flaw is its poor scalability. Simulations showed that an increase from 15 to 30 users greatly decreased its throughput (∼42%) , especially with a high load. Thus, if this was extended to let’s say 50 nodes, it may provide unreasonable results.

The reason for this poor scalability is due to the exhaustive search when determining the appropriate route. This will work in a small network, however with each added channel or secondary user the computation time to determine the weight for all the routes will increase drastically. To prove this I developed Equation 5 to determine the number of routes that the algorithm must check:

\[
R(M) = \prod_{i=1}^{i=M} A_i \quad for \ A_i = 1,2,3 ... N \quad (5)
\]

Where \( M \) is the number of frequencies available for the entire network, \( N \) is the number of nodes and \( A_i \) represents the number of nodes that can transmit on frequency “i”。 I wrote a mat lab script, that can be found in the Appendix, which shows the variability of the number of routes as the number of frequencies increases from 1 to 6. For each frequency band a random number of nodes were chosen to be available, similar to the author’s simulations.

Figure 10 show the number of route vs. the number of channels. Each line in the graph represents a separate network with different primary users on each channel. You can see how varied the number of routes can be, and how large it potential can become when you increase the number of available channels to six. The author stated his algorithm is made for static networks however, as described in section 3.2.1, one application of static networks in the television white space. This band of frequencies has 70 channels that the node can transmit on [10], which would cause this algorithm to become inoperable.
To improve this algorithm, the author needs to look at methods of decreasing the routes that the algorithm needs to check. An example of how this can be done can be shown in the heuristic algorithm in the protocol described in Section 6. In that protocol the author proves that a portion of the routes will always be worse off than other route, thus they do not have to be checked. If something like this could be implemented, it may make this protocol a more viable option in centralized real world applications.

4.2 Distributed Protocols
Distributed protocols, unlike centralized protocols, do not make the assumption that all the secondary users know all the information about the spectrum. Thus, each secondary user must measure the spectrum around it looking for idle channel or active primary users and broadcast it across the network. Unlike the centralized section I will not be describing any basic algorithms because I will be covering more advanced algorithms in the following sections which will provide a better representation of what is currently being discussed in literature. I will however, discuss the problems that arise when working with distributed networks and choices the authors have to make when designing their routing protocol.

4.2.1 Control Information
The major difference between distributed and centralized networks is that users can only measure their own local spectrum bands. If users make their routing decisions based only on their sensed spectrum information, their choices may result in worse results for the network as a whole. Thus they must gather global information to make better decisions, which are usually done by with the control channel. When users broadcast on the control channel they need to ensure they do not interrupt the primary users. So protocols must find intelligent techniques to convey the control information without affecting the primary nodes traffic, while trying to reduce the overhead of the network.

Figure 10: Number of routes per Channel for 15 Access Points and 30 Access Point
Two main approaches have been used in literature [5]. The first is based on a synchronization window which uses a fixed time slot before every transmission, and all nodes are tuned to specific frequencies and exchange all control messages. This method requires a centralized clocking between the nodes so they can exchanging synchronization information in the specific time slot, which is very difficult and usually impractical to implement in real networks.

The second approach uses the common control channel for exchanging control information. Since nodes can transmit on this channel whenever they please, given that no one is currently transmitting on it, it removes the synchronization problems that were found in the first method. The trade off in this approach is you have to ensure that the control channel still has enough time slots to allow normal control activities, association etc., to occur.

4.2.2 Source or Destination Based Routing
Routing protocols must choose whether to use destination based or source based routing approaches. Destination based routing requires periodic routing tables and potential broadcasting route requests and route replies. This type of routing is used in many tradition ad hoc networks. In cognitive radio network there are usually multiple channels, thus the messages will have to be broadcasted across all channels, which will result in a lot of overhead. An alternative is transmitting in the control channel; however this may results in a large influx of message on the control channel, thus causing it to lose its purpose of informing nodes about the network status.

For source based routing protocols every node, before starting communication, locally computes the path to the destination using control messages sent by the users. The advantage of source based routing is that no routing tables are used, thus deceasing the overhead from updating them when the network changes. The node is forwarded based on the information found in the header. However, there is a chance the control information that the source received will be wrong, thus the packet will not reach destination and the process will have to start over.

5.0 Geographical Protocol for Dynamic Networks
This section will describe a more complex distributed protocol that is designed for dynamic networks and incorporates locational service. A geographical protocol is an extension of a distributed protocol that uses the location of the secondary users via GPS installed on each device to provide a more optimal solution. The protocol that I will be discussing in this area is called “SpEctrum Aware Routing for Cognitive ad-Hoc networks” (SEARCH) [11]. SEARCH is a completely distributed routing solution that accounts for Primary users, the mobility of cognitive users and jointly considers the path and channel choices to decrease path latency. It was developed to operate on networks with primary users that have a dynamic nature. The main contribution of the algorithm to the research community is evaluating when the coverage region of a primary user should be circumvented, or when the user should just change their channel.

The majority of geographic routing protocols for traditional multi-hop networks formulate their algorithm with a greedy nature. A greedy nature means that the algorithm does not consider fairness, it
just tries to make the network as optimum as possible. In literature, a lot of geographic protocols use a metric based on distance traveled by the packet. In this case the greedy nature is represented by all users always choosing their next hop based on the node that will provide the largest advance in distance towards the destination.

With the addition of primary users in CRNs the greedy algorithm must be altered to ensure that the secondary users do not interfere with the primary users. The algorithm provides the node with two choices if a primary user is found:

1) Switch channels
2) Go around the primary user

If the node chooses to go around the primary user, it will usually result in more hops, thus increasing the latency seen by the packet. Switching channels may also create latency if the new channel is very busy. There is also potential that switching to a nearby channel on the spectrum may not solve the problem since the primary user may have power leaking into another channel.

The author makes the following assumption for the network:

- Primary users are static
- CCC is used to communicate among nodes
- The sender knows the location of its destination
- Each nodes knows its own location either through GPS or triangulation techniques
- There are M channels available with a known bandwidth

The algorithm can be split up into two parts, the initial route set up and the route enhancement/repair.

5.1 Initial Route Setup
A route request (RREQ) is transmitted by the source on each channel that is not affected by a nearby primary user. This RREQ will be forwarded by intermediate users till it reaches the destination. Throughout the trip from the source to the destination the intermediate hops attached their

i) ID
ii) Current Location
iii) Time Stamp
iv) A flag represent which mode it is operating in.

Each node checks their focus region for any hops that comply with the greedy forwarding requirements. If none exist, it means there may be a primary user nearby, so the node flips its operating flag, changing its mode to primary user avoidance. It will go around the primary user and then switch back to greedy forwarding mode. Once all the routes have been received by the destination it chooses the best route with the joint channel-path optimization.

March 5, 2013
5.1.1 Greedy Forwarding

The algorithm by default operates in greedy forwarding mode. At the beginning of the protocol, beacon messages are used to inform the nodes which users are in their transmission range. Using this information the protocol can determine which of the candidate forwarders should be chosen as the next hop to minimize the distance to the destination. To be an eligible relay the node must satisfy the following three conditions:

1) The next hop must be on the same channel

2) The next hop must not interfere with a primary user

3) The user must be in the current node focus region

The first is logical since the channels are orthogonal so if a user tries to transmit to a user on a different channel they will not receive any packets. The second is also logical because we know from the rules of many cognitive radio networks that you cannot interfere with primary users. The third condition is what allows the algorithm to determine the general locations of primary users. The focus region is generated by having a line, with a magnitude of the transmission range, from the source to the destination, and then drawing a pie shape with the angle of \((\theta)\) max above and below the line. A focus region example is shown in Figure 11. In the example, the source node only has one option to transmit its packets too, B. Even if C is closer, it is not in its focus region, thus does not meet condition 3, so it is not a possible relay nodes.

The best way to provide a more detailed explanation is by an example. Figure 12 shows a source node and a destination node, it also shows 4 possible relay nodes A,B,C,E. If the primary user is turn off, then the greedy algorithm will choose the path, Source-> A->E->Destination. This is the shortest route and will provide the least delay. However, if the primary user turns on, node A has no nodes in its focus region since condition 2 is not satisfied for node E. In this case it goes into PU avoidance mode and go around the primary user. I will continue this example after I cover a few more topics.
If there are no nodes in a focus region the node is known as decision point, these decision points are used to give the destination a general idea of locations of primary users and occupied channels.

5.1.2 PU avoidance
When a node determines it’s a decision node it sets the PU avoidance flag in the RREQ before sending it to the next hop. It relaxes the constraints to just 1 and 2 and chooses the next relay hop. At the next hop, if there is a node in its focus region, it goes back to greedy operation and continues to the destination. In example 2, node A will choose B if the primary user if operating. B will check its focus region and see C, thus it will go back to the greedy operation mode and continue to D.

5.1.3 Joint Channel-Path optimization
If the licensed band is free of primary users, then there will be no regions that had to be avoided, and the RREQ’s send on different channels should have similar paths.

If there are primary users in the network then the destination will received RREQ’s with decision nodes that represent where the primary users are. The destination will choose the route with the least latency. To improve this route even more the destination will look at the RREQ’s on different channels and determine if it’s faster to switch channels or to go around that primary user. After it decides upon the optimal channels sends a route reply RREP back to the source along the optimal route.

5.2 Routing Algorithm Mathematical Formulation
The previous section was included to help the reader understand the basics of the algorithm and this section will provide a more rigorous representation.

5.2.1 Route Setup
An anchor point represents the nodes that are included during the forwarding process of the RREQ packets. All the anchor points for a route are shown with the following equation, where k defines the channel of L_k anchor points.

\[ P_k = \{A^1_k, A^2_k \ldots A^{L_k}_k\} \quad (6) \]

The optimum greedy path P_g is the union of the anchor points and the channel switching decisions shown with \( M^{k,i}_j \). The total number of channels is M and the transmission time of a packet on channel i is \( T_i^k \).

Step 1: Initial Path Selection

The source sends RREQ that traverses the network and reaches the destination. The destination extracts the path information \( P_k \) and their respective time stamps \( t^k_j \). Latency from an intimidate node m to the destination node \( L_k \) is calculated by Equation 7:

\[ L^m_k = t^L_k - t^m_k \quad (7) \]

At this step the total latency is calculated for each path and the path with the least amount of latency is chosen among channels k, as shown in Equation 8.
\[ i = \arg \min_k \{ L_k, \forall k \in C \} \]  \hspace{1cm} (8)

At this step we want to check all the decision points along the chosen route to see if we should go around the primary users or change channels. \( P_G \) is initialized on the starting node on the channel \( i \) as \( P_G = A^1_i \)

**Step 2: Greedy Path Formation**

The least latency path may be further improved by switching to a different channel at the decision points. Beginning at the node chosen in step one, the algorithm continues along the path adding to \( P_G \) each node that is not a decision node. If a decision node is found it goes to step 3, if the destination is found then it goes to step 5.

**Step 3: Optimization at a Decision Point**

When a decision point is found, the algorithm checks to see if any other channels will provide a better solution. It determines the time taken to switch to the channel \( t_s \), and the time to transmit a packet to the closest node in that channel \( T_R^k \) on channel \( k \), as

\[ \delta_k = t_s + T_R^k \]  \hspace{1cm} (9)

So the node should only be switched to channel \( k \), which has a route latency of \( L_k \) to the destination if Equation 10 is true

\[ \min \{ \delta_k + L_k \} < L_{Current \ Channel} \]  \hspace{1cm} (10)

**Step 4: Route Expansion**

The greedy path solution is updated with the new paths and channels. It then returns to Step 2 to look for any more decision nodes.

**Step 5: Route Conformation**

When the destination is reached, the route is sent back to the source to sends its packets across the optimum route.

**5.2.2 Route Enhancement**

The route enhancement stage comes after the route has been set up and it tries to incorporate all the work for setting up a route to help set up other routes in the network. For example if a node nearby to the source wants to send to the same destination, it will only have to make a route to the source and then follow the route already optimized before.

This stage also wants to ensure that the most optimum route is still the one chosen. So it sends periodic packets from the source that takes other routes to the destination. If these packets reach the destination and their latency is shorter than the original path, it updates the route to the new path.
5.2.3 Route Maintenance

The algorithm spends a lot of resources making routes, thus it tries its best to maintain them. It considers two major problems when trying to maintain a route:

1) **Primary User Awareness:** A primary user may appear in the network causing a node in a particular route to be required to turn off thus causing the route to break.

To deal with this problem, each node, after the route has been set up, periodically checks to see if the next hop is in its focus region. If the next node in the route is no longer in its focus it marks itself as a decision point and stops routing data and send a route request to the destination. This results in a new route creation, the same as described in Section 5.2.1 however the decision point is the new source.

2) **CR User mobility:** Node mobility may also cause disconnections in the network, so this must also be considered. The nodes themselves are actually not included in the routing tables; it’s the archer points which represent regions that are primary user free. Each node updates its one hop neighbors via beacons, as long as the node is within a threshold distance it is kept in the table, if a node is not in a threshold distance then it is removed and another node that is in the same anchor area is added.

5.2.4 Simulations

The simulation tool used to test this algorithm was a network simulation tool called ns-2. The primary users activity was modeled using the on off application. 400 Nodes are used in a 1000 x 1000m area with 2 to 10 primary users. The coverage of a primary user in its occupied channel is 300 m and the transmission range of a cognitive radio user is 120m. The node and the primary user’s location are randomly chosen and averages of 50 trials were used to generate results.

The algorithm is compared with the Greedy Perimeter Stateless Routing protocol (GPSR) that has been extended for multichannel environment. This algorithm is oblivious to primary users and just tries to get the best path available. Two types of SEARCH algorithm are compared. The least latency does not include the route enhancements or the channel optimization, while the other does.

![Figure 13: End to end delay vs. number of primary users, 5 Channels [11]](image-url)
Figure 13 shows the end to end delay as the number of primary users increase from 10 to 20, with 5 channels. We can see that with an increase in the primary users it only increased the delay by a small amount with the optimized SEARCH algorithm. The GPRS algorithm had a large increase in the delay because when a primary user breaks its path it does not try and fix the route it just has to make an entire new route. The least latency SEARCH algorithm is worse than the optimized SEARCH when there is a larger number of primary users because it only considers going around PU’s, thus there is a lot of delay from the increased number of hops. Also the route enhancement in the optimized SEARCH will ensure that the route is always the best route.

Figure 14: End to end delay vs. number of primary users, 10 Channels. [11]

Figure 15: End to End delay vs. connection load. [11]
Figure 14 is similar to the previous plot however there are now 10 channels to operate on. We can see that the delay for the optimized SEARCH algorithm still operates with very low delay due to the channel optimization. However, since there is more channels there will probably be more hops in the SEARCH algorithms. The reason for the increased delay in GPRS as I had mentioned before is from the retransmission delay of trying to find another route when the route breaks because of a primary user.

Figure 15 considers the case with 10 primary users and 5 channels, with varying load. We see that the optimized SEARCH algorithm had the the smallest delay. The least latency search as well as GPRS suffer from self-contention packets as only one channel is used for data forwarding and no channel switching occurs.

5.2.5 Conclusions and Opinions
The SEARCH protocol has a lot of potential and performed well simulations. The only downfall is that this algorithm cannot operate on network with highly dynamic channel conditions. The reason is cannot is because they use routing tables that will not be kept up to date with a highly dynamic network. If the protocol tried to keep the tables up to date, it would result in a large amount of overhead. One improvement could be to incorporate past primary user statistics when choosing which channel to operate on. This will allow secondary users to avoid operating in channels where the primary users are even more dynamic than usual.

However, this algorithm will work with any dynamic or static network, so overall I think the paper was well written and simulations showed that with the route maintenance and enhancement it will scale well to large networks.

6.0 Opportunistic Cognitive Multi-hop Protocol
The last protocol considered in this report is another extension of the distributed protocol that is designed for networks with primary users that have a highly dynamic nature, the paper is called “Spectrum-Aware Opportunistic Routing in Multi-Hope cognitive Radio Networks” [12]. In the protocol discussed in Section 5, SEARCH, end-to-end routing tables were used. However, as I described routing tables only work on networks that have static or dynamic channel conditions. For networks that have highly dynamic channel conditions, once the route has been update in the routing table there is a high probability the network has changed and thus the table now has incorrect information. In Khalif et al paper “Multi-hop Cognitive Radio Networks: To Route or Not to Route” [5] he shows that the involved computations and communication overhead for re-building routing tables for all flows is non-trivial, especially when the channel status changes frequency.

The proposed algorithm makes routing decisions on a per packet (or per group of packets) basis and uses channel usage statistics in the discovery of the spectrum access opportunities to improve the transmission performance of all the secondary users. The forwarding links are selected based on local spectrum access opportunities. A routing metric is also proposed that tries to increase performance while maintaining reasonable complexity.
6.1 Protocol Overview
This protocol considers a multi-hop cognitive radio network with multiple primary and secondary users that share a set of orthogonal channels. All secondary users communicate on a common control channel (CCC) and it is assumed that the devices have 2 radios, one for transmission and one for control messages. The secondary users use the on/off time of the primary user to model channel usage statistics that are used to help make future decisions during the routing process.

Whenever a user wants to forward a packet the following two steps occur:

1) **Channel Sensing**: In the channel sensing step the secondary user searches for a temporarily unoccupied channel in collaboration with its neighbors. It does this by measuring the energy on the channels.

After selecting a channel, the sender broadcasts a short message in the CCC to inform its neighbors of its selected data channel and the location of the sender and destination. All the other secondary users in the network that receive this message set that data channel as “non-accessible”. This is to prevent co-channel interference from current transmissions. Using the locational information from the CCC message the secondary users determine whether they are an eligible relay candidate (i.e. if the relay node is closer to the destination node then the current sender)

The user will then initiate a handshake with the relay candidates.

2) **Relay Selection**: After a channel has been chosen, the sender selects the next hop from the potential relay candidates. This is done in two steps; the sender first broadcasts a routing request (RREQ) message to the relay candidates.

When a relay candidate receives the RREQ and is eligible to be a relay, it creates a back off timer until it sends a routing response RRSP message back to the sender. This back off timer is based the candidate’s throughput, delay and relay distance achieved. The better the specifications, the smaller the back off timer. All the eligible candidate listen to the channel to see if another user reply first, if so they stop their back off timer and wait for the next RREQ.

The sender waits till it receives its first RRSP messages and chooses that node as its relay. Since the back off timers are based on the specification the first RRSP should have the best specification. When it receives the RREQ is computes a handshake with the eligible candidate and begins to transmit data on the selected channel. If the sender does not hear any RRSP messages, this means there are no eligible candidates for relaying and it will return to the channel sensing stage and try again.

The author did not discuss any protection measure to make sure that a secondary user will not be stuck in an endless loop looking for users. For example, if there was no other relays in the area it may just continue to try and find a relay forever. Thus a small extension of this would be to have a countdown timer that till only let the algorithm run, let’s say 5 times, and then it would fail and return to the previous hop chooses a different hop to send the packets too.
6.2 Performance Metrics
The author used the relay distance advancements and per-hop transmission delay to measure how the algorithm was performing.

6.2.1 Relay Distance Advancement
The distance achieved by a relay is determined by the difference of the distance of the sender to the destination and the new relay node to the destination, as shown in Equation 11.

\[ \text{Dist}(S, R) = d(S, D) - s(R, D) \]  \hspace{1cm} (11)

For \( S \) represents the sender location, \( D \) the destination location, \( R \) the forward node location.

6.2.2 Per Hop Delay
The per hop transmission delay is comprised of three parts:

1) Sensing Delay: The sensing delay includes the transmission time of the sensing invitation (SNSINV) and the energy detection time.

2) Relay Selection Delay: The \( i \)-th relay candidate \( R_i \) sends an RRSP message that is received first only when the first \( i - 1 \) higher-priority candidates are not eligible or have worse channel statistics. Thus, the delay can be represented in Equation 12 which was proposed in “Efficient and Reliable Broadcast in Inter-Vehicle Communication Networks: A cross level approach” [13]

\[ T_{RS}(i) = T_{RREQ} + (i + 1)\mu + T_{RRSP} + 2 \text{SIFS} \]  \hspace{1cm} (12)

TRREQ and TRRSP are the transmission times of the two types of messages and \( \mu \) is the duration of one mini-slot in the back off period.

3) Packet Transmission Delay: Once the relay has been selected the packet transmission delay is determined using Equation 13. This includes the delay of the packet transmission TDATA and the ACK transmission times.

\[ T_{DTX} = T_{DATA} + T_{ACK} + 2 \text{SIFS} \]  \hspace{1cm} (13)

The total delay can be represented by Equation 14.

\[ T_{relay} = T_{SNS} + T_{RS} + T_{DTX} \]  \hspace{1cm} (14)

6.3 Probabilistic Formulations
6.3.1 Channel Sensing
The author provided a great derivation of the probability that a secondary device will find an idle channel to transmit on and how the past channel statistics are used to help improve the channel selection.
Let $I_{c_j}^R$ denote the event that $c_j$ is sensed to be idle by a secondary user “R”. A channel is determined to be idle if it was sensed idle at $t_1$ and remains idle till $t_2$. Given the channel status observed at earlier times the channel status can be estimated. Given that the on/off durations follow an exponential distribution with mean $\frac{1}{E[T_{c_j}^O]}$ and $\frac{1}{E[T_{c_j}^I]}$, the following probability can be calculated using Equation 15:

$$ P_{OFF,R}(t_0, t_1) = \begin{cases} \rho_{c_j} + \left(1 - \rho_{c_j}\right)e^{-\Delta_{c_j}(t_1-t_0)}, & \text{if } c_j \text{ is OFF at } t_0 \\ \rho_{c_j}e^{-\Delta_{c_j}(t_1-t_0)}, & \text{if } c_j \text{ is ON at } t_0 \end{cases} $$

(15)

where

$$ \rho_{c_j} = \frac{E[T_{c_j}^O]}{E[T_{c_j}^O] + E[T_{c_j}^I]} $$

$$ \Delta_{c_j} = \frac{1}{E[T_{c_j}^O]} + \frac{1}{E[T_{c_j}^I]} $$

$\rho_{c_j}$ represents the chance for an idle state in $c_j$. To calculate the likelihood of a channel staying idle during a sensing period we can use renewal theory, which states that the residual time of a state in an alternating process truncates since the time origin can be expressed by the equilibrium distribution of the state duration. Thus, the following probability can be found, as shown in Equation 16, which describes the probability that a channel $R$ state is in the idle state during the sensing period $[t_1,t_2]$:

$$ P_{R}^{c_j}(t_1, t_2) = \int_{t_2-t_1}^{\infty} \frac{\tau_{OFF}(u)}{E[T_{c_j}^I]} \, du $$

(16)

In the previous equation $\frac{\tau_{OFF}(u)}{E[T_{c_j}^I]}$ represents the probability density function of the residual time of an idle channel since the time originally where it was sensed idle. Thus we can say that the probability that $R$ detects a channel $c_j$ that is free is represented in Equation 17.

$$ Pr\{I_{c_j}^R\} = P_{OFF,R}^{c_j}(t_0, t_1) \cdot P_{R}^{c_j}(t_1, t_2) $$

(17)

6.3.2 Relay Selection

After an idle channel has been found the sender needs to select which node to forward its data too. In this algorithm the candidate with the highest relay priority will be chosen, however, a primary user may interrupt this process and cause a failure in the relay selection. This case is quite rare, since it will only occur if a primary user appears in the relay selection period, which is usually less than 1 millisecond.

This probability can be represented using Equation 18:

$$ P_{RS \, fail}^{c_j} = Pr\{I_{S}^{c_j}\} \cdot Pr\{\bigcap_{R_i \in \text{Re}} \overline{I_{S}^{c_j}} \} $$

(18)

Where $Pr\{I_{S}^{c_j}\}$ is the probability that the sender initiates the relay selection when it detects an idle channel, this is defined in the channel selection section. $Pr\{\bigcap_{R_i \in \text{Re}} \overline{I_{S}^{c_j}} \}$ is the probability that all
secondary users sense the channel busy in the previous sensing, which is equivalent to the event that no relay candidate replies in the relay selection step.

An assumption made by the author is that each secondary user only deals with one primary user on a frequency band. This is generally a valid assumption since most networks try to only have one primary user on each frequency to avoid co-channel interference.

### 6.3.3 Data Transmission

Once a relay has been selected the data transmission on the link is successful if no active primary users appear during the transmission period. This can be expressed by Equation 20.

\[
P_{\text{relay}, R_i}^{c_j} = P_{i}^{c_j} \cdot P_{SR_i}^{c_j}(t_3, t_4)\quad (19)
\]

\[
P_{\text{relay}, R_i}^{c_j} = P_{i}^{c_j} \cdot P_{S}^{c_j}(t_3, t_4) \cdot P_{R_i}^{c_j}(t_3, t_4)(1 - x_{SR_i}^{c_j})\quad (20)
\]

Where \( l_{SR_i} \) represents the data transmission link, \( x_{SR_i}^{c_j} = 1 \) if the two secondary users are affected by the same Primary user and 0 otherwise, \( P_{i}^{c_j} \) is the probability that the relay will be selected as the next hop.

### 6.4 Joint Channel and Relay Selection Extension

To improve the performance of the algorithm the authors jointly consider the selection of the channel and relay node. This is beneficial to the algorithm since the relay performance can have aspects that result from the channel and the relay node, so they both must be considered. To do this a new metric is introduced to capture the effects of the joint channel and relay selection and apply it to a heuristic algorithm to select the best relay and channel.

#### 6.4.1 The OCR Metric

The new metric is called the Cognitive Transport Throughput (CCT) and characterizes the one hop relay performance of the algorithm. CCT is the expected bit advancement per second for one hop of a packet with a payload \( L \) on channel \( c_j \). It can be represented in Equation 22.

\[
\text{CCT}(c_j, R_D^{c_j}) = E \left[ L \cdot \frac{A_{Dj}}{T_{\text{relay}}} \right] \quad (21)
\]

\[
\text{CCT}(c_j, R_D^{c_j}) = \sum_{R_i \in R_D^{c_j}} P_{\text{relay}, R_i}^{c_j} \frac{L \cdot A_{D(S,R_i)}}{T_{\text{relay}(R_i)}} \quad (22)
\]

This metric just considers the distance traveled, which may reduce delay, but if all nodes abide by this then it may cause a large congestion on just a few nodes since they are the closest to the destination. So I think they should add some sort of congestion measure in the metric.
6.5 Heuristic Algorithm

Many algorithms that have to choose between a number of channels/hops which is usually done with an exhaustive search for all the possible combinations. This restricts the scalability of an algorithm because the more nodes and channels you add, the longer the convergence time to determine the best combination.

For this algorithm, if an exhaustive search was used with \( n \) relays (\( k \) being eligible candidates) and \( m \) channels it will take the following amount of time:

\[
m \cdot CCT_{time} \cdot \sum_{k=1}^{n} P(n, k) \tag{23}
\]

If we let \( n \to \infty \) then we can see the worst running time \( O(m \cdot n! \cdot e) \) where \( e \) is the base for a natural log. Thus as \( n \) goes large, the exhaustive method is infeasible. Thus, to improve the scalability of the protocol the authors proposed their own heuristic algorithm to decrease the number of relays that they must compute the CCT of.

The optimization problem can be broken down into two phases due to the independent usage statistics for different channels. The first step is to find all the candidate relays in each channel and determine the relay with the highest CTT.

The secondary step is to compare all the CTT values from all the available channels and then choose the channel with the highest CTT. The second step is usually limited so the authors focused on the first step for their optimization. To do this they proposed a lemma shown in Lemma 1 that showed that you can ignore the relay candidates that are affected by the same primary user, since they have proven that it will not lower the CTT.

**Lemma 1**: Given a feasible relay selection set \( R^c_j \), \( \forall R_{i_1}, R_{i_2} \in R^c_j \), if \( V_{R_{i_1}} > V_{R_{i_2}}, X_{R_{i_1}, R_{i_2}}^c = 1 \), then

\[
CCT(c_j, R^c_j) \geq CCT(c_j, R^c_j). \text{ Proof can be found in [12].}
\]

With this proposition the new complexity is \( O(m \cdot n^2) \) which is a lot lower.

6.5.1 Delay Simulation of Heuristic Algorithm

I wanted to see how much the heuristic helped improve the delay of the network so I plotted both the heuristic and the exhaustive algorithm maximum run time as seen in Figure 16.
You can see that the heuristic algorithm operates a lot better than the exhaustive algorithm. This makes sense since the algorithm does not have to check as many routes when trying to find the minimum. This idea should have been considered in many of the other protocol discusses in this report and their convergence time may have also been decreased.

6.6 Simulations
The algorithm was simulated in a C++ event driven simulator. The primary users were modeled as an exponential process with parameters $\frac{1}{E[T_{OFF}]}$ and $\frac{1}{E[T_{ON}]}$ and an idle rate of $\rho = \frac{E[T_{OFF}]}{(E[T_{OFF}] + E[T_{ON}])}$. The network was set up with multiple primary and secondary users placed randomly in an area of 800 x 800 m$^2$. The source and destination were chosen 700m away from each other and a constant bit rate was used. The performance was evaluated in terms of end-to-end delay and packet delivery ratio, and finally the hop count. The simulations show comparisons with SEARCH algorithm described in Section 5 and the following algorithms:

**OCR (CTT)** is the proposed algorithm where the relay and channel candidates are set jointly using the CTT metric and the heuristic algorithm.

**OCR (OPT)** is when the channel and the relay are determined using an exhaustive searching for the route that provides the largest CTT value.

**GOR**: A geographic opportunistic routing algorithm in which the secondary user first selects the channel with the greatest success probability of packet transmission. If that channel is sensed idle, then the secondary user selects a relay user on that channel based on its relay capabilities.
**GR:** This is a general geographic routing protocol. In this algorithm the secondary user first selects a frequency band by sensing for idle channels. After a channel has been chosen it selects the relay that will provide the largest distance towards the destination. The parameters use in the simulations can be found in Table 2.

<table>
<thead>
<tr>
<th>Number of Channels</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\rho_{c_1}, \rho_{c_2}, \rho_{c_3}, \rho_{c_4}, \rho_{c_5}, \rho_{c_6})</td>
<td>(0.3, 0.3, 0.5, 0.5, 0.7, 0.7)</td>
</tr>
<tr>
<td>Number of PU's per channel</td>
<td>11</td>
</tr>
<tr>
<td>PU coverage</td>
<td>250m</td>
</tr>
<tr>
<td>(E[T_{off}])</td>
<td>[100 ms, 600ms]</td>
</tr>
<tr>
<td>Number of SUs</td>
<td>[100, 200]</td>
</tr>
<tr>
<td>SU Transmission Range</td>
<td>120m</td>
</tr>
<tr>
<td>Source-Destination Distance</td>
<td>700m</td>
</tr>
<tr>
<td>SU CCC rate</td>
<td>512 kbps</td>
</tr>
<tr>
<td>SU data channel rate</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>CBR delay threshold</td>
<td>2 s</td>
</tr>
<tr>
<td>Mini-slot time</td>
<td>4 us</td>
</tr>
<tr>
<td>Per channel sensing time</td>
<td>5 ms</td>
</tr>
<tr>
<td>Channel Switching time</td>
<td>80 us</td>
</tr>
<tr>
<td>PHY header</td>
<td>192 us</td>
</tr>
<tr>
<td>(r_{max})</td>
<td>2</td>
</tr>
</tbody>
</table>

**6.6.1 Primary User Activates**

Search and CTT were compared with different channel conditions in Figure 17. We can see that the CTT algorithm greatly outperforms the SEARCH algorithm when the primary expected value decreased. This is the SEARCH algorithm is unable to keep the routing tables values relevant without causing a large amount of overhead.

**6.6.2 Multi-User Diversity**

The author’s next test was to see how node density affected the protocols relay performance. The number of secondary users was varied from 100 to 200. Table 3 gives the average number of neighbors based on the number of secondary users.

<table>
<thead>
<tr>
<th>Number of Secondary Users</th>
<th>Average Number of Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>7.0686</td>
</tr>
<tr>
<td>120</td>
<td>8.4823</td>
</tr>
<tr>
<td>140</td>
<td>9.8960</td>
</tr>
<tr>
<td>160</td>
<td>11.3097</td>
</tr>
<tr>
<td>180</td>
<td>12.7235</td>
</tr>
<tr>
<td>200</td>
<td>14.1372</td>
</tr>
</tbody>
</table>
You can see from the plot in Figure 18 that the end-to-end delay increases as the number of secondary user’s increases. This is because there are more potential routes, so there is a better chance you will find one with a smaller delay.

![Figure 18: End-to-end delay vs. Number of Secondary Users [12]](image)

We can see in Figure 18 that the OCR algorithm is significantly better than the GOR and GR in plot. In GOR and GR, a channel is selected first, and then the relay is selected. The coordination overhead increases with the number of relays that exist. Where in OCR (CTT), the channel and relay are selected jointly, which reduces the overhead. We also can see that OCR(OPT) and OCR(CTT) are very similar in terms of performance however in terms of computational expense the OCR takes 5 times as long to converge since it exhaustively checks all the routes.

**6.6.3 Effectiveness in Routing Metric**

Finally the author looked at the end-to-end delay versus the flow rate as shown in Figure 19.
We can see that as the flow rate increases a GOR and GR delay increase quite drastically. This is again because the OCR jointly considers the optimum channel and link selection, while the other two protocols select the channel and relays separately.

### 6.7 Comments and Conclusions

This algorithm is the most versatile algorithm that has been discussed in this report. Many real world networks are highly dynamic, so this is a great option. The author only considered networks constant channel conditions, thus the algorithm could be extend varying channel conditions. This would require buffer times to determine if a route has broken or something has caused poor channel conditions for a short time.

Simulations showed its comparison with SEARCH and it can be seen that routing per packet is usually better than routing tables. However, I found it quite suspicious that they did not compare the SEARCH algorithm is some of the simulations such as the end-to-end delay while increasing flow rate. I assume that SEARCH outperforms the algorithm when the network is not highly dynamic. Thus proving the point from the beginning or the report, there cannot be one general routing protocol for all networks.

### 7.0 Summary of Protocols

Table 5 (Larger version found in appendix) provides a summary of all the protocols described in the report. The table shows the progression from simple centralized solutions to a distributed opportunistic solution.

The routing metric used by each author was usually very different as shown in Table 5. This makes it difficult to compare the algorithms since each author shows his/her results based on her particular metric. I believe that authors should focus on what I think are the two most important metrics for real networks, Jitter and Delay. Table 4 shows how important each QoS requirements are for different types of network [14].
Table 4: Quality of Service Requirements [14]

<table>
<thead>
<tr>
<th>Application</th>
<th>Bandwidth</th>
<th>Delay</th>
<th>Jitter</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>File Sharing</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Web Access</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Audio Streaming</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Video Streaming</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>VOIP</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Videoconferencing</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

Jitter is the standard deviation in the delay of packet arrive times thus it is highly coupled with the delay of the network. Bandwidth is an important QoS requirement but if there is not enough bandwidth available, users usually can just decrease the quality or the speed of the download. Thus, although it is nice to have higher bandwidth, it is not a necessity in most cases.

Delay and jitter cannot be fixed by decreasing the quality of the video/ speed of download. Although jitter not an issue in applications like email or file sharing, it can cause a lot of problems with streaming audio and video. A company called “Sandvine” conducted a study in 2012 to determine the distribution of the types of internet traffic. The report “Global Internet Phenomena” [15] produced the following results shown in Figure 20.

![Figure 20: Peak Period Aggregate Traffic Composition (North America, Fixed Access)](image)

With applications such as “Netflix” accounting for over 50% of internet traffic, streaming packets are very important to consider. During streaming, if the jitter is high, then packets will arrive out of order and cause the audio or video to not be represented correctly. Thus, I feel like protocols should focus on improving the end-to-end delay first and then focus on the other requirements.

Scalability is also a very important factor in protocol design. In Table 5 you can see a lot of the algorithms covered in this report have poor scalability. The main cause for this issue is the exhaustive
Section 4.1.4.4 and 6.5.1 showed how exhaustively searching causes a large increase in convergence time when the users or channels increase.

Table 6 (Larger version found in appendix) shows a summary of the simulations for the three major protocols discussed. As mentioned above it is difficult to compare the algorithms since the authors all use different metrics to show that their algorithms work.

The layered graph simulations showed that it had very poor scalability. The decrease in throughput with only 30 node shows that this algorithm will only work in a very specific networks.

SEARCH and the opportunistic algorithm both showed their end-to-end delay based on a varying expected value of the primary user off time. We can see that when the off time is relatively high 600 ms, the two algorithms perform quite similar. This is because the routing tables are able to keep up with the changing network, thus there is not an excess of overhead from updating the tables. However, as the off time decreases, we see that the overhead from updating the routing tables cause the SEARCH’s end-to-end delay to increase drastically compared to the almost constant end-to-end delay of the opportunistic algorithm.

Table 5: Summary of the protocol presented

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Routing Metric</th>
<th>Centralized or Distributed</th>
<th>Geographical</th>
<th>Applicable in Opportunistic Networks</th>
<th>Applicable in Dynamic Networks</th>
<th>Applicable in Static Networks</th>
<th>Route Maintenance</th>
<th>Routing Methodology</th>
<th>Scalability</th>
<th>Further Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coloring Graph</td>
<td>Number of Channels</td>
<td>Centralized</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Updates graph periodically</td>
<td>Exhaustive search for best route</td>
<td>Poor</td>
<td>Ignore redundant routes when checking for best path, introduce weights for the colored edges to incorporate other specifications into the metric</td>
</tr>
<tr>
<td>Conflict Graph</td>
<td>No-Conflict Route</td>
<td>Centralized</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Updates graph periodically</td>
<td>Exhaustive search for best route</td>
<td>Poor</td>
<td>A better searching technique and route maintenance to reduce the overhead from generating new graphs</td>
</tr>
<tr>
<td>Layered Graph</td>
<td>Minimize Edge Weight</td>
<td>Centralized</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Updates graph periodically</td>
<td>Exhaustive search for best route</td>
<td>Poor</td>
<td>Introduce a technique to reduce number of routes searched to reduce complexity</td>
</tr>
<tr>
<td>Optimization Approach</td>
<td>Maximize Spectrum Reuse Factor</td>
<td>Centralized</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Optimizes updated problem</td>
<td>Solving an optimization problem that models network</td>
<td>Poor</td>
<td>Formulate into a binary mixed integer program to reduce the convergence time</td>
</tr>
<tr>
<td>SEARCH</td>
<td>Shortest Path (Latency)</td>
<td>Distributed</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Generates primary user awareness and CR user mobility into the algorithms metric</td>
<td>Medium</td>
<td>Reduce the number of routes checked to improve scalability, incorporate past primary user statistics to make better channel decisions</td>
<td></td>
</tr>
<tr>
<td>Spectrum-Aware Opportunistic Routing Algorithm</td>
<td>Relay distance advancement and per-hop transmission delay</td>
<td>Distributed</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Periodically check route to ensure it is still viable</td>
<td>Per packet routing decision</td>
<td>High</td>
<td>Consider varying channel conditions, Protection measure for determining eligible candidates, Add congestion measure to DCR metric</td>
</tr>
</tbody>
</table>
Table 6: Simulation summary of presented protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Simulation Details</th>
<th>Results and Comments</th>
</tr>
</thead>
</table>
| Layered Graph                                      | -Matlab was used to simulate the network/protocol  
-Both 15 and 30 node networks were tested  
-Nodes placed in a 100 x 100 grid  
-6 available channels with bandwidth of 10 Mbps                                                                                                                                                                                                                                                                                                                                                                                 | -For 15 nodes with load varying from 10-60 packets per second, normalized throughput decreased from 98% to 75%  
-No delay simulations presented  
-For 30 nodes with load varying from 10-60 packets per second, normalized throughput decreased from 90% to 50%  
-No delay simulations presented                                                                                                                                                                                                                                                                                                                                                                                   |
| SEARCH                                             | -Ns-2 was used to simulate the network/protocol  
-On/Off application used to model users data flows  
-400 Nodes used in a 1000 x 1000 m range  
-10 Primary Users                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -End to end delay with $E[T_{off}]$ ranging from 100 – 600 ms, delay varied from 1200 to 150 ms  
- End to end delay with load (kbps) ranging from 100 – 700 ms, delay varied from 1 to 3.8 s  
- End to end delay with number of secondary users varying from 80-120, delay decreased from 107ms to 75ms  
- End to end delay with flow rate (pps) varying from 10-70, delay increased from 50ms to 425ms                                                                                                                                                                                                                                                                                                           |
| Spectrum-Aware Opportunistic Routing Algorithm      | -A C++ event driven simulator  
-Node placed on a 800 x 800 range  
-Source and destination always 700m away  
-6 Channels with bandwidth of 2 Mbps                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -End to end delay with $E[T_{off}]$ ranging from 100 – 600 ms, delay stayed constant at 120 ms  
-End to end delay with number of secondary users varying from 80-120, delay decreased from 107ms to 75ms  
- End to end delay with flow rate (pps) varying from 10-70, delay increased from 50ms to 425ms                                                                                                                                                                                                                       |

8.0 Conclusion

Multi-hop cognitive radio networks are one of the most promising research areas in wireless communications. There are a variety of real world multi-hop networks that would benefit the implementation of cognitive radios. This report discussed the many challenges that arise when dealing with multi-hop cognitive radio networks and also characterized different types of networks, describing which types of protocols work best with each one.

A number of protocols were discussed showing the variety of different approaches authors are taking to solve the challenges of these types of networks. This variety shows that no single routing algorithm will work for all networks, since there are always tradeoffs that occur when trying to improve a certain aspect.
Future work should consider improving the scalability of centralized networks or trying to reduce the complexity of distributed algorithms while maintaining similar results. Authors should also look into channels that have uncertain usage statistics and the impact of measurement errors on protocol performance.

Finally, only YongKang Liu’s algorithm incorporated primary user channel statistics into its metric. I believe this information is very useful and should be incorporate in the other protocols as it prevents secondary user’s from using channels that have nearby Primary Users with highly dynamic on/off times.

9.0 References


10.0 Appendix

10.1 Layered Graph Matlab Code
clc;
clear all;
for o=1:10
l=6
    for j=1:l
        for i=1:j
            P(i) = round(rand(l)*15);
            if P(i) == 0
                P(i)= 1;
            end
        end
        B15(j,o) = prod(P);
P=ones(10,1);
    end
    for p=1:l
        for i=1:p
            K(i) = round(rand(l)*30);
            if K(i) == 0
                K(i)= 1;
            end
        end
        B30(p,o) = prod(K);
K=ones(10,1);
    end
end
B15Avg = sum(B15,2)/10;
B30Avg = sum(B30,2)/10;

10.2 Summary Tables
<table>
<thead>
<tr>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce the number of routes to improve scalability</td>
<td>The algorithm models mobility into the network and can be used in scenarios where mobility is an issue</td>
<td>Reduce the number of routes to improve scalability</td>
</tr>
<tr>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Implement a algorithm to reduce the number of routes reached to reduce the network traffic</td>
<td>The algorithm models mobility into the network and can be used in scenarios where mobility is an issue</td>
<td>Reduce the number of routes to improve scalability</td>
</tr>
<tr>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>A better algorithm is used to reduce the network traffic</td>
<td>The algorithm models mobility into the network and can be used in scenarios where mobility is an issue</td>
<td>Reduce the number of routes to improve scalability</td>
</tr>
<tr>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Implementation of a new algorithm to reduce the number of routes reached</td>
<td>The algorithm models mobility into the network and can be used in scenarios where mobility is an issue</td>
<td>Reduce the number of routes to improve scalability</td>
</tr>
<tr>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Further improvements to the algorithm</td>
<td>The algorithm models mobility into the network and can be used in scenarios where mobility is an issue</td>
<td>Reduce the number of routes to improve scalability</td>
</tr>
<tr>
<td>Protocol</td>
<td>Simulation Details</td>
<td>Results and Comments</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>Layered Graph</td>
<td>- Matlab was used to simulate the network/protocol</td>
<td>- For 15 nodes with load varying from 10-60 packets per second,</td>
</tr>
<tr>
<td></td>
<td>- Both 15 and 30 node networks were tested</td>
<td>normalized throughput decreased from 98% to 75%</td>
</tr>
<tr>
<td></td>
<td>- Nodes placed in a 100 x 100 grid</td>
<td>- For 30 nodes with load varying from 10-60 packets per second,</td>
</tr>
<tr>
<td></td>
<td>- 6 available channels with bandwidth of 10 Mbps</td>
<td>normalized throughput decreased from 90% to 50%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- No delay simulations presented</td>
</tr>
<tr>
<td>SEARCH</td>
<td>- Ns-2 was used to simulate the network/protocol</td>
<td>- End to end delay with $E[T_{off}]$ ranging from 100 – 600 ms,</td>
</tr>
<tr>
<td></td>
<td>- On/Off application used to model users data flows</td>
<td>delay varied from 1200 to 150 ms</td>
</tr>
<tr>
<td></td>
<td>- 400 Nodes used in a 1000 x 1000 m range</td>
<td>- End to end delay with load (kbps) ranging from 100 – 700 ms,</td>
</tr>
<tr>
<td></td>
<td>- 10 Primary Users</td>
<td>delay varied from 1 to 3.8 s</td>
</tr>
<tr>
<td>Spectrum-Aware Opportunistic</td>
<td>- A C++ event driven simulator</td>
<td>- End to end delay with $E[T_{off}]$ ranging from 100 – 600 ms,</td>
</tr>
<tr>
<td>Routing Algorithm</td>
<td>- Node placed on a 800 x 800 range</td>
<td>delay stayed constant at 120 ms</td>
</tr>
<tr>
<td></td>
<td>- Source and destination always 700m away</td>
<td>- End to end delay with number of secondary users varying from</td>
</tr>
<tr>
<td></td>
<td>- 6 Channels with bandwidth of 2 Mbps</td>
<td>80-120, delay decreased from 107ms to 75ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- End to end delay with flow rate (pps) varying from 10-70, delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>increased from 50ms to 425ms</td>
</tr>
</tbody>
</table>