

TRANSMISSION SCHEDULING FOR ROUTING PATHS AND MULTIPATHS
IN COGNITIVE RADIO MESH NETWORKS

by
Xia Zhao

A thesis submitted in partial fulfillment
of the requirements for the degree
of
Master of Science
in
Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2010

©COPYRIGHT

by

Xia Zhao

2010

All Rights Reserved

APPROVAL

of a thesis submitted by

Xia Zhao

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

Dr. Richard S. Wolff

Approved for the Department of Electrical Engineering

Dr. Robert C. Maher

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

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

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

Xia Zhao

May, 2010

TABLE OF CONTENTS

1. INTRODUCTION	1
Wireless Mesh Networks (WMNs)	1
Introduction to Cognitive Radio Networks	6
The Physical Architecture of Cognitive Radio	10
Cognitive Radio Network Architecture	12
Transmission Scheduling in Cognitive Radio	13
Our Work and Contributions	16
2. RELATED WORK.....	18
3. PROBLEM DEFINITION.....	20
Max-Path-Bitflow	22
Max-Multipath-Bitflow	23
4. THE SCHEDULING ALGORITHM AND ANALYSIS	25
Greedy Colorings	25
Odd-Even Multipath Scheduling	26
Example Path Transmission Schedule	29
Analysis	29
Scheduling Typical Paths	31
A Simple Rounding Scheme	32
5. FINDING ROUTING PATHS AND MULTIPATHS	34
Routing to Maximize Path Bottleneck Capacity	34
DFS Multipath Routing	35
Depth first search(DFS)	35
Multipath Routing.....	35
6. SIMULATION ENVIRONMENT AND NETWORK MODEL	37
Simulation Software	37
Simulation Model.....	39
7. NUMERICAL RESULTS.....	44
Scenario 1: Performance on random paths	44
Shortest Path Routing Performance	44
Bottleneck Path Routing Performance	45
Multipath Routing Performance	46

TABLE OF CONTENTS CONTINUED

Scenario 2: Comparison of the three approaches	46
Statistical analysis of the results.....	47
Scenario 3: Varying the number of channels available.....	48
Scenario 4: Network saturation	49
Scenario 5: Throughput versus path length	50
Scenario 6: Distribution of channel utilization	51
8. CONCLUSIONS.....	53
REFERENCES CITED	54

LIST OF TABLES

Table	Page
1. Maximum transmission distances by frequency and data rate	43
2. Interference ranges by frequency	43

LIST OF FIGURES

Figure	Page
1. Hybird WMNs [1]	3
2. Spectrum utilization [12]	5
3. Spectrum hole concept [12].....	8
4. Physical architecture of the cognitive radio [12]	11
5. The cognitive radio network architecture [12].....	13
6. Example graph [34]	15
7. Interference model	21
8. Example of multipath routing.....	23
9. Example of Greedy Coloring [33]	27
10. The (rounded) frame schedule produced for the Example transmission path from Section 4.....	30
11. An example of DFS	35
12. The simulated network (50 km $\times$ 50 km) and connections. Only the shortest paths are shown in this figure (not the bottleneck and multipath)	40
13. Shortest Path Routing Performance for each path shown in Figure 12 .	45
14. Bottleneck Path Routing Performance	46
15. Multipath Routing Performance	47
16. Schedule performance results for three approaches	48
17. Standard error for three approaches	49
18. Average path transmission rate versus the number of available channels	50

LIST OF FIGURES CONTINUED

Figure		Page
19.	Network saturation as the number of active connections increase	51
20.	Transmission rate versus path length.....	52
21.	Distribution of channel utilization	52

GLOSSARY

Cognitive Radio — A cognitive radio is a radio that can change its transmitter or receiver parameters based on interaction with the environment in which it operates.

Wireless Mesh Networks (WMNs) — A WMNs is dynamically self-organized and self-configured, with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves (creating, in effect, an ad hoc network).

Primary users — Primary users are the users who have the license to operate in a certain spectrum band.

Inter-flow interference — Inter-flow interference is interference among links belonging to different flows.

Intra-flow interference — Intra-flow interference is interference among links belonging to common flows.

Half-duplex data transmission — Half-duplex data transmission means that data can be transmitted in both directions on a signal carrier, but not at the same time.

Dijkstra's algorithm — Dijkstra's algorithm conceived by Dutch computer scientist Edsger Dijkstra in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree.

Depth first search(DFS) — DFS is an uninformed search that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hasn't finished exploring.

Minimum spanning tree (MST) — A minimum spanning tree (MST) or minimum weight spanning tree is a spanning tree with weight less than or equal to the weight of every other spanning tree.

ABSTRACT

Nodes in a cognitive radio mesh network may select from a set of available channels to use provided they do not interfere with primary users. This ability can improve overall network performance but introduces the question of how best to use these channels. This project addresses the following specific problem: given a routing path P , choose which channels each link in P should use and their transmission schedule so as to maximize the end-to-end data flow rate (throughput) supported by the entire path. This problem is relevant to applications such as streaming video or data where a connection may be long lasting and require a high constant throughput. The problem is hard due to the presence of both intra-flow and inter-flow interference. We have developed a new constant-factor approximation algorithm for this problem. If certain conditions on the path are met, the performance guarantee is $1/4$ of optimal. It has been shown by simulation results that the end-to-end throughput given by the proposed algorithm is often within 90% or better of optimal.

INTRODUCTION

Wireless Mesh Networks (WMNs) are considered as an economical solution to provide broadband Internet access in a large area [1]. In WMNs, nodes are comprised of mesh routers and mesh clients. Each node operates not only as a host but also as a router, forwarding packets on behalf of other nodes that may not be within direct wireless transmission range of their destinations. A WMN is dynamically self-organized and self-configured, with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves (creating, in effect, an ad hoc network). Due to its application area, a WMN is required to efficiently deliver a high volume of traffic and support various real-time multimedia applications. The distinct features and critical design factors of WMNs bring many challenging issues to communication protocols, ranging from the physical layer to the application layer. Despite recent advances in the research and development in WMNs, many challenging problems still remain, such as resource allocation, transmission scheduling and interference mitigation. Also end-to-end throughput and Quality of Service(QoS) support are critical issues in WMNs.

Wireless Mesh Networks (WMNs)

Wireless mesh networks (WMNs) have emerged as a key technology for next-generation wireless networking. They are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an ad hoc network and maintaining the mesh connectivity. WMNs are comprised of two types of nodes: mesh routers and mesh clients. Other than the routing capability for gateway/bridge functions as in a conventional wireless router, a mesh router contains additional routing functions to support mesh networking. Through multi-hop communications, the same coverage can be achieved by a mesh router with much lower transmission power.

Mesh routers have minimal mobility and form the mesh backbone for mesh clients. The hardware platform and software for mesh clients can be much simpler than

those for mesh routers. In addition to mesh networking among mesh routers and mesh clients, the gateway/bridge functionalities in mesh routers enable the integration of WMNs with various other networks. Thus, WMNs will greatly help users to be always-on-line anywhere, anytime.

WMNs diversify the capabilities of ad-hoc networks. This feature brings many advantages to WMNs, such as low up-front cost, easy network maintenance, robustness, reliable service coverage, etc. Therefore, in addition to being widely accepted in the traditional application sectors of ad hoc networks, WMNs are undergoing rapid commercialization in many other application scenarios such as broadband home networking, community networking, building automation, highspeed metropolitan area networks, and enterprise networking.

The architecture of WMNs can be classified into three types:

- *Infrastructure/Backbone WMNs.* In this architecture, mesh routers form an infrastructure for clients. It can be built using various types of radio technologies, in addition to the widely used IEEE 802.11 technologies today.
- *Client WMNs.* Client meshing provides peer-to-peer networks among client devices. In this type of architecture, client nodes constitute the actual network to perform routing and configuration functionalities as well as providing end-user applications to customers.
- *Hybrid WMNs.* This architecture is the combination of infrastructure and client meshing, as shown in Fig. 1. Mesh clients can access the network through mesh routers as well as directly meshing with other mesh clients.

The characteristics of WMNs are outlined below, where the hybrid architecture is considered for WMNs, since it comprises all the advantages of WMNs:

- *Multi-hop wireless network.* An objective in the development of WMNs is to extend the coverage range of current wireless networks without sacrificing the channel capacity. Another objective is to provide non-line-of-sight (NLOS) connectivity among the users without direct line-of-sight (LOS) links. To meet these requirements, the mesh-style multi-hopping is indispensable [2], which achieves higher throughput without sacrificing effective radio range via shorter

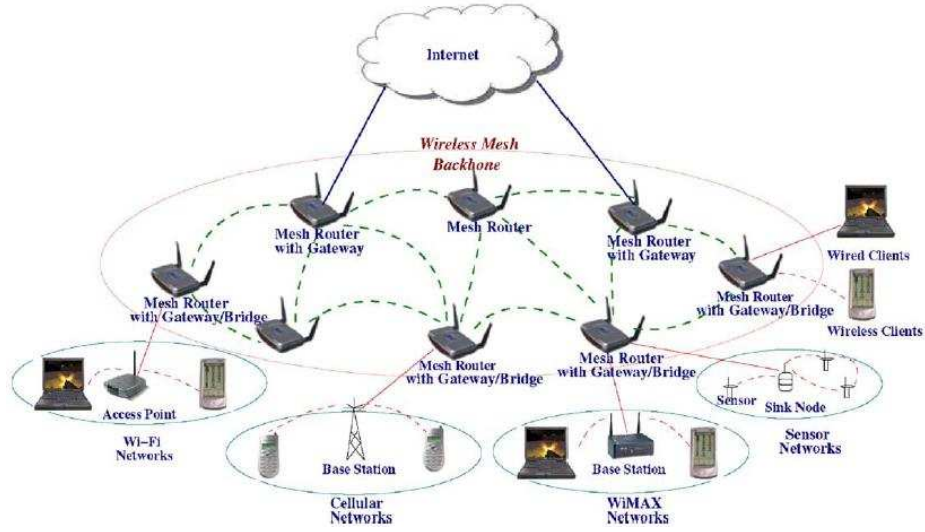


Figure 1. Hybrid WMNs [1].

link distances, less interference between the nodes, and more efficient frequency re-use.

- *Support for ad hoc networking, and capability of self-forming, self-healing, and self-organization.* WMNs enhance network performance, because of flexible network architecture, easy deployment and configuration, fault tolerance, and mesh connectivity, i.e., multipoint-to-multipoint communications [3]. Due to these features, WMNs have low upfront investment requirements, and the network can grow gradually as needed.
- *Mobility dependence on the type of mesh nodes.* Mesh routers usually have minimal mobility, while mesh clients can be stationary or mobile nodes.
- *Multiple types of network access.* In WMNs, both backhaul access to the Internet and peer-to-peer (P2P) communications are supported [4]. In addition, the integration of WMNs with other wireless networks and providing services to end-users of these networks can be accomplished through WMNs.
- *Dependence of power-consumption constraints on the type of mesh nodes.* Mesh routers usually do not have strict constraints on power consumption. However,

mesh clients may require power efficient protocols. As an example, a mesh-capable sensor [5, 6] requires its communication protocols to be power efficient. Thus, the MAC or routing protocols optimized for mesh routers may not be appropriate for mesh clients such as sensors, because power efficiency is the primary concern for wireless sensor networks [7, 8].

- *Compatibility and interoperability with existing wireless networks.* For example, WMNs built based on IEEE 802.11 technologies [9, 10] must be compatible with IEEE 802.11 standards in the sense of supporting both mesh-capable and conventional Wi-Fi clients. Such WMNs also need to be inter-operable with other wireless networks such as WiMAX, ZigBee [11], and cellular networks.

Spectrum is the most precious resource for wireless networks. It is currently managed by a static approach which assigns a fixed portion of the spectrum to a particular license holder or a wireless service for exclusive use. Over the past few years, the world has experienced a very rapid proliferation of wireless devices. Certain parts of the spectrum, such as the 2.4GHz band and the 5GHz band, are heavily used by various wireless devices, resulting in serious interference and poor network performance. On the other hand, a significant amount of spectrum remains under-utilized or even not utilized at all, which has been shown by recent studies and experiments [12] as illustrated in Fig. 2.

For example, cellular network bands are overloaded in most parts of the world, but TV broadcasting, amateur radio and paging frequencies are not. Independent studies performed in some countries confirmed that observation, and concluded that spectrum utilization depends strongly on time and place. Moreover, fixed spectrum allocation prevents rarely used frequencies (those assigned to specific services) from being used by unlicensed users, even when their transmissions would not interfere at all with the assigned service. This was the reason for the proposal of allowing unlicensed users to utilize licensed bands whenever it would not cause any interference (by avoiding them whenever legitimate user presence is sensed).

In order to achieve much better spectrum utilization and viable frequency planning for WMNs, cognitive radios or frequency-agile radios are being developed to

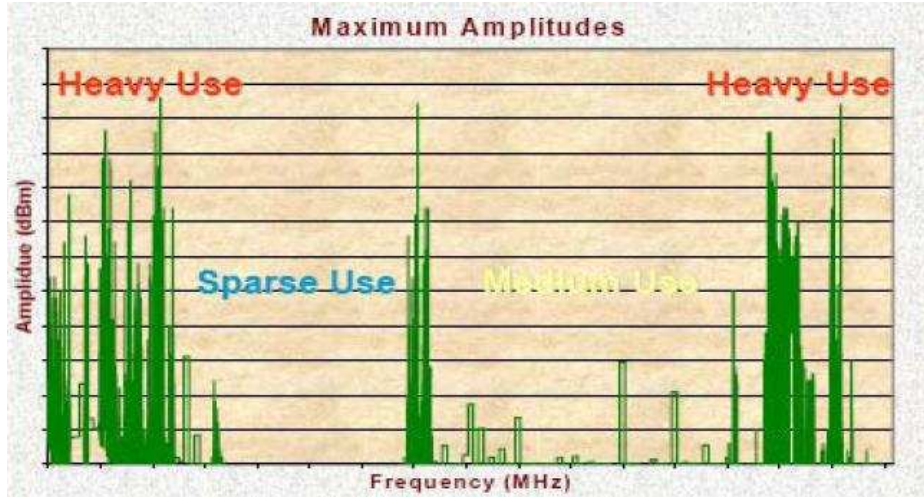


Figure 2. Spectrum utilization [12].

dynamically capture the unoccupied spectrum. Implementing cognitive radios on a software radio platform is one of the most powerful solutions, because all components of a radio, such as RF bands, channel access modes, and channel modulations, are programmable.

Emerging cognitive radios enable dynamic spectrum access. With cognitive radios, unlicensed wireless users (a.k.a secondary users) can sense and access the under-utilized spectrum bands opportunistically even if it is licensed, as long as the licensed wireless users (a.k.a primary users) in these spectrum bands are not disrupted [12]. In this way, interference can be avoided and network capacity can be significantly improved.

Cognitive radios are desirable for a WMN in which a large volume of traffic is expected to be delivered since they are able to utilize available spectrum more efficiently and therefore improve network capacity significantly [12]. However, they introduce additional complexities to bandwidth allocation and transmission scheduling. With cognitive radios, each node can access a set of available spectrum bands which may span a wide range of frequencies. Different spectrum bands (a.k.a channels) can support quite different transmission ranges and data rates, both of which have a significant impact on resource allocation.

Introduction to Cognitive Radio Networks

The idea of cognitive radio was first presented officially in an article by Joseph Mitola III and Gerald Q. Maguire, Jr in 1999. It was a novel approach in wireless communications that Mitola later described as:

The point in which wireless personal digital assistants (PDAs) and the related networks are sufficiently computationally intelligent about radio resources and related computer-to-computer communications to detect user communications needs as a function of use context, and to provide radio resources and wireless services most appropriate to those needs [13].

Cognitive radio (CR) is an intelligent wireless communication system that is aware of its surrounding environment (i.e. outside world), and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters (e.g., transmit power, carrier frequency, and modulation strategy) in real-time, with two primary objectives in mind:

Highly reliable communications whenever and wherever needed;

Efficient utilization of the radio spectrum.

CR is a form of wireless communication in which a transceiver can intelligently detect which communication channels are in use and which are not, and instantly move into vacant channels while avoiding occupied ones. This optimizes the use of available radio-frequency (RF) spectrum while minimizing interference to other users.

Possible functions of cognitive radio include the ability of a transceiver to determine its geographic location, identify and authorize its user, encrypt or decrypt signals, sense neighboring wireless devices in operation, and adjust output power and modulation characteristics.

There are two main types of cognitive radio, full cognitive radio and spectrum-sensing cognitive radio. Full cognitive radio takes into account all parameters that a wireless node or network can be aware of. A spectrum-sensing cognitive radio is used to detect channels in the radio frequency spectrum.

The Federal Communications Commission (FCC) ruled in November 2008 that unused portions of the RF spectrum (known as white spaces) be made available for public use. White space devices must include technologies to prevent interference, such as spectrum sensing and geolocation capabilities.

NeXt Generation (xG) communication networks, also known as Dynamic Spectrum Access Networks (DSANs) as well as cognitive radio networks, will provide high bandwidth to mobile users via heterogeneous wireless architectures and dynamic spectrum access techniques. The inefficient usage of the existing spectrum can be improved through opportunistic access to the licensed bands without interfering with the existing users. xG networks, however, impose several research challenges due to the broad range of available spectrum as well as diverse Quality-of-Service (QoS) requirements of applications. These heterogeneities must be captured and handled dynamically as mobile terminals roam between wireless architectures and along the available spectrum pool.

The key enabling technology of xG networks is the cognitive radio. Cognitive radio techniques provide the capability to use or share the spectrum in an opportunistic manner. Dynamic spectrum access techniques allow the cognitive radio to operate in the best available channel. More specifically, the cognitive radio technology will enable the users to (1) determine which portions of the spectrum is available and detect the presence of licensed users when a user operates in a licensed band (spectrum sensing), (2) select the best available channel (spectrum management), (3) coordinate access to this channel with other users (spectrum sharing), and (4) vacate the channel when a licensed user is detected (spectrum mobility). Once a cognitive radio supports the capability to select the best available channel, the next challenge is to make the network protocols adaptive to the available spectrum. Hence, new functionalities are required in an xG Networks to support this adaptivity. In summary, the main functions for cognitive radios in xG networks can be summarized as follows:

- *Spectrum sensing*: Detecting unused spectrum and sharing the spectrum without harmful interference with other users.

- *Spectrum management*: Capturing the best available spectrum to meet user communication requirements.
- *Spectrum mobility*: Maintaining seamless communication requirements during the transition to better spectrum.
- *Spectrum sharing*: Providing the fair spectrum scheduling method among co-existing xG users.

The ultimate objective of the cognitive radio is to obtain the best available spectrum through cognitive capability and reconfigurability as described before. Since most of the spectrum is already assigned, the most important challenge is to share the licensed spectrum without interfering with the transmission of other licensed users as illustrated in Fig. 3. The cognitive radio enables the usage of temporally unused spectrum, which is referred to as a spectrum hole or white space [14]. If this band is further used by a licensed user, the cognitive radio moves to another spectrum hole or stays in the same band, altering its transmission power level or modulation scheme to avoid interference as shown in Fig. 3.

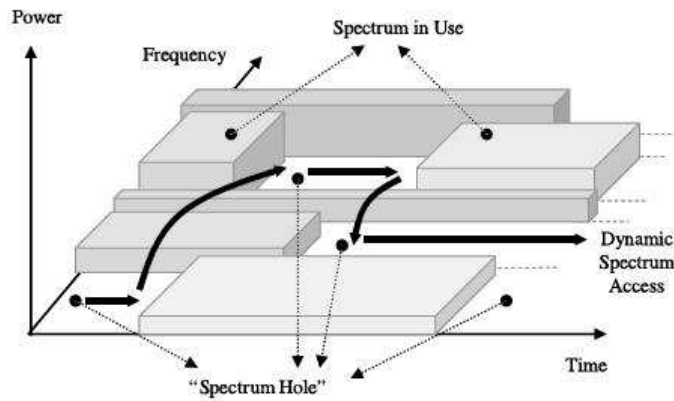


Figure 3. Spectrum hole concept [12].

The term, cognitive radio, can also formally be defined as follows [15]: *A Cognitive Radio is a radio that can change its transmitter parameters based on interaction with the environment in which it operates.* From this definition, two main characteristics of the cognitive radio can be defined [14, 16]:

Cognitive capability: Cognitive capability refers to the ability of the radio technology to capture or sense the information from its radio environment. This capability cannot simply be realized by monitoring the power in some frequency band of interest but more sophisticated techniques are required in order to capture the temporal and spatial variations in the radio environment and avoid interference to other users. Through this capability, the portions of the spectrum that are unused at a specific time or location can be identified. Consequently, the best spectrum and appropriate operating parameters can be selected. The tasks required for adaptive operation in open spectrum, which is referred to as the cognitive cycle.

Reconfigurability: The cognitive capability provides spectrum awareness whereas reconfigurability enables the radio to be dynamically programmed according to the radio environment. More specifically, the cognitive radio can be programmed to transmit and receive on a variety of frequencies and to use different transmission access technologies supported by its hardware design [17]. This capability enables the cognitive radio to adapt easily to the dynamic radio environment. There are several reconfigurable parameters that can be incorporated into the cognitive radio [15] as explained below:

- *Operating frequency:* A cognitive radio is capable of changing the operating frequency. Based on the information about the radio environment, the most suitable operating frequency can be determined and the communication can be dynamically performed on this appropriate operating frequency.
- *Modulation:* A cognitive radio should reconfigure the modulation scheme adaptive to the user requirements and channel conditions. For example, in the case of delay sensitive applications, the data rate is more important than the error rate. Thus, the modulation scheme that enables the higher spectral efficiency should be selected. Conversely, the loss-sensitive applications focus on the error rate, which necessitate modulation schemes with low bit error rate.
- *Transmission power:* Transmission power can be reconfigured within the power constraints. Power control enables dynamic transmission power configuration within the permissible power limit. If higher power operation is not necessary,

the cognitive radio reduces the transmitter power to a lower level to allow more users to share the spectrum and to decrease the interference.

- *Communication technology*: A cognitive radio can also be used to provide interoperability among different communication systems.

In the following subsections, I will explain the physical architecture of cognitive radio, the cognitive radio network architecture and related work.

The Physical Architecture of Cognitive Radio

A general architecture of a cognitive radio transceiver with RF/analog front-end is shown in Fig. 4(a) [17]. The transceiver and the RF/analog processing unit are the main components of a cognitive radio. Each component can be reconfigured via a control bus to adapt to the time-varying RF environment. In the RF front-end, the received signal is amplified, down converted and A/D converted. In the baseband processing unit, the signal is modulated/demodulated and encoded/decoded. The baseband processing unit of a cognitive radio is essentially similar to existing transceivers. But the novelty of the cognitive radio is the RF front-end, which we will focus on. The novel characteristic of the transceiver is a wideband sensing capability of the RF front-end. This function is mainly related to RF hardware technologies such as wideband antenna, power amplifier and adaptive filter. RF hardware for the cognitive radio should be capable of tuning to any part of a large range of frequency spectrum. Generally, a wideband front-end architecture for the cognitive radio has the following structure as shown in Fig. 4(b) [18]. The components of a cognitive radio RF front-end are as follows:

- RF filter: The RF filter selects the desired band by bandpass filtering the received RF signal.
- Low noise amplifier: The low noise amplifier amplifies the desired signal while simultaneously minimizing noise component.
- Mixer: In the mixer, the received signal is mixed with locally generated RF signal and converted to the baseband or the intermediate frequency.

- Voltage-controlled oscillator (VCO): The VCO generates a signal at a specific frequency for a given voltage to mix with the incoming signal. This procedure converts the incoming signal to baseband or an intermediate frequency.
- Phase locked loop: The PLL ensures that a signal is locked on a specific frequency and can also be used to generate precise frequencies with fine resolution.
- Channel selection filter: The channel selection filter is used to select the desired channel and to reject the adjacent channels. The direct conversion receiver uses a low-pass filter for the channel selection, while the superheterodyne receiver adopts a bandpass filter.
- Automatic gain control (AGC): The AGC maintains the gain or output power level of an amplifier constant over a wide range of input signal levels.

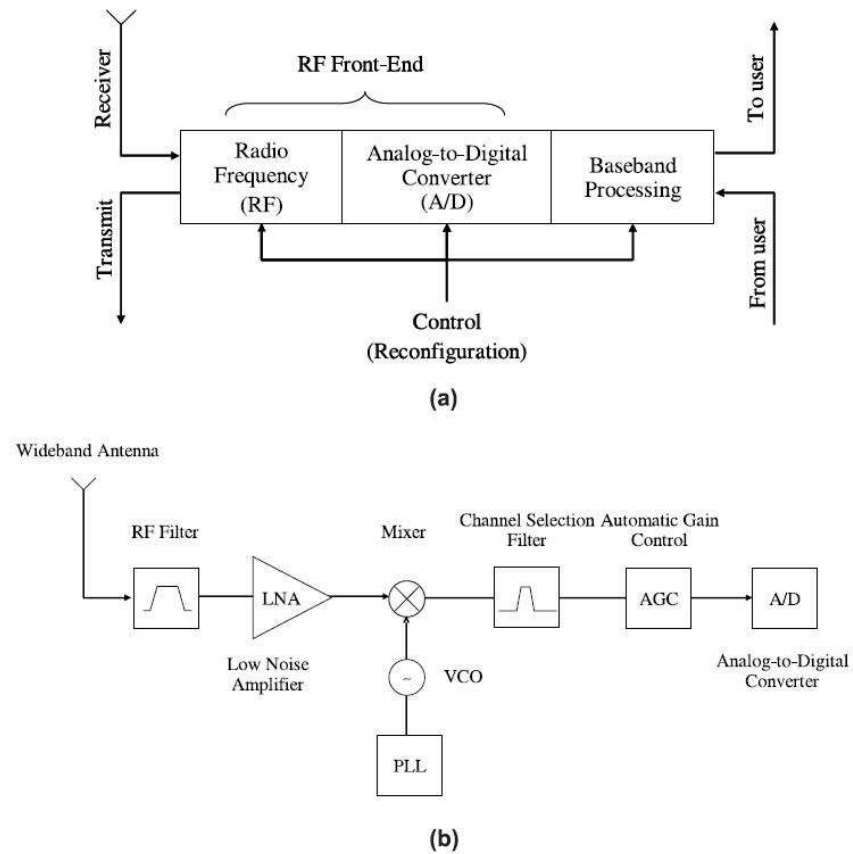


Figure 4. Physical architecture of the cognitive radio [12].

In this architecture, a wideband signal is received through the RF front-end, sampled by the high speed A/D converter, and measurements are performed for the detection of the licensed user signal. However, there exist some limitations on developing the cognitive radio front-end. The wideband RF antenna receives signals from various transmitters operating at different power levels, bandwidths and locations. As a result, the RF front-end should have the capability to detect a weak signal on a large dynamic range. The requirement of a multi-GHz speed A/D converter necessitates the dynamic range of the signal to be reduced before A/D conversion. This reduction can be achieved by filtering strong signals. Since strong signals can be located anywhere in the wide spectrum range, tunable notch filters are required for the reduction [18]. Another approach is to use multiple antennas such that signal filtering is performed in the spatial domain rather than in the frequency domain. Multiple antennas can receive signals selectively using beamforming techniques [19].

Cognitive Radio Network Architecture

The components of the CR network architecture are showed in Fig. 5.

The components of the CR network architecture, as show in Fig. 5, can be classified as two groups: the primary network and the CR network. The primary network is referred to as an existing network, in which the users have been assigned a license to operate in a certain spectrum band. Due to their priority in spectrum access, the operations of primary users should not be affected by unlicensed users (secondary users).

On the other hand, the CR network users do not have the privilege to operate in a primary users band. CR networks can also be equipped with CR basestations that provide single-hop connection to CR users. Finally, CR networks may include spectrum brokers that play a role in distributing the spectrum resources among different CR networks [20].

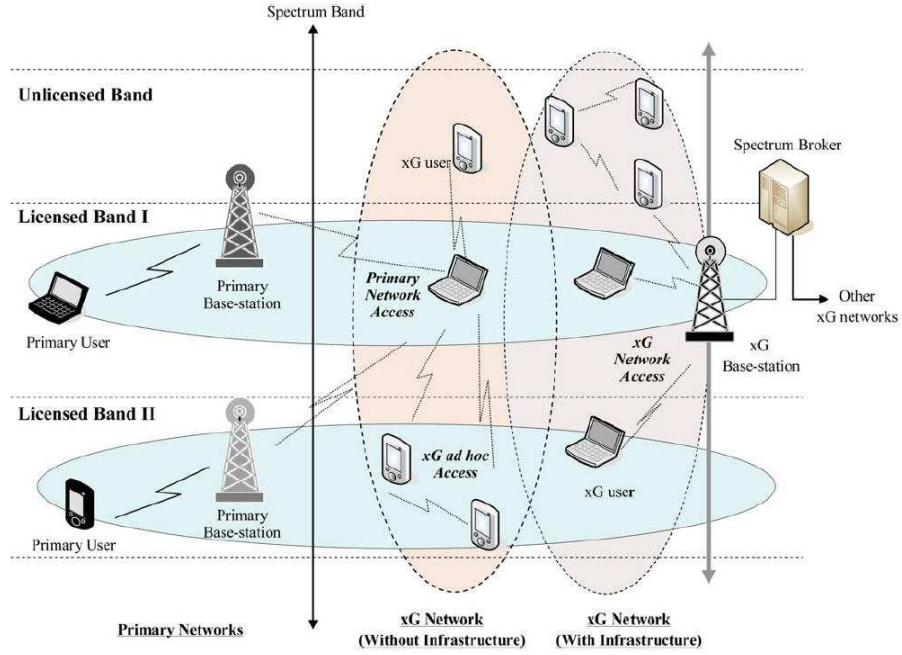


Figure 5. The cognitive radio network architecture [12].

CR users are capable of accessing both the licensed portions of the spectrum used by the licensed users and the unlicensed portions of the spectrum through wide-band access technology. Consequently, the operation types for CR networks can be classified as *licensed band operation* and *unlicensed band operation*.

- **Licensed band operation:** The licensed band is primarily used by the primary network. Therefore, the CR networks are focused mainly on the detection of primary users in this case. The channel capacity depends on the interference at nearby primary users. Furthermore, if primary users show up in the spectrum band occupied by CR users, CR users should vacate that spectrum band and move to available spectrum immediately [12, 20, 21].
- **Unlicensed band operation:** In the absence of primary users, CR users have the same right to access the spectrum. Hence, sophisticated spectrum sharing methods are required for CR users to compete for the unlicensed band [8, 12, 21].

Transmission Scheduling in Cognitive Radio

A transmission scheduling scheme should determine which CR link can transmit when there is a competition. The issue of transmission scheduling has been discussed in [22, 23, 24, 25] respectively. However, different from what is done in traditional networks, transmission scheduling in CR networks must also consider the primary usage patterns, for the channel available time seen by each CR link may be different due to the activities of the primary users.

In a multi-hop wireless network comprised of CR-enabled devices, Medium Access Control(MAC) layer scheduling for data communication involves assignment of timeslots and channels to either links or nodes in the network. The number of channels available and the channel identities vary from one node to another within the CR network. This is in contrast to the existing use of multiple channels where all the nodes have the same set of channels available (for example in IEEE 802.11 networks).

CR technology [26] enables wireless devices to dynamically adapt to spectrum availability in their geographical region. The owner of a channel is referred to as primary user and all other users of the channel as secondary users [27]. A CR node can access a channel only when the primary users are not currently using it. Therefore, the performance of a CR system is much dependant on the primary usage patterns. To increase the end-to-end throughput of a CR, the scheduling scheme should be adjusted according to the primary usage patterns.

CR technology allows secondary users to periodically scan and identify available channels in the spectrum. The secondary users can communicate among themselves using the identified available channels even though those channels have been licensed to primary users. Since the availability of channels varies widely from one region to another [28] and based on the time of the day, communication among secondary users must rely on efficient channel and timeslot assignment schemes. The varying channel availability at nodes gives rise to the following Medium Access Control (MAC) layer issues: (i) How does a node decide when and on which channel it should tune to

in order to exchange data with its neighbors? (ii) How do the nodes decide on a MAC-layer schedule in the absence of a central authority?

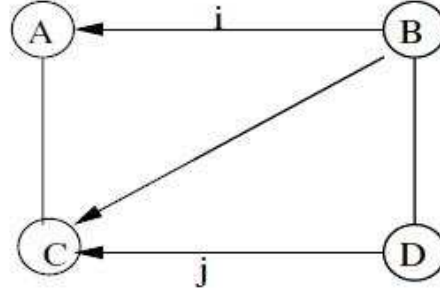


Figure 6. Example graph [34].

In a CR network, nodes can tune to different channels at different points in time. Since transmissions on different channels do not interfere with each other, using multiple channels in parallel across the network increases the overall network throughput. For instance, consider the example shown in Fig. 6. If the network has just one channel, transmissions on both links (i and j) and at the same time is not possible without causing interference. But if multiple channels are available on links and , then using two different channels, communication can take place over both the links simultaneously. In a CR network, the number of available channels is not fixed and the channels may lie anywhere in the entire spectrum. This is in contrast to the existing use of multiple channels, where the set of available channels is the same for all nodes. To enable collision-free communication among nodes in the network, either every link or every node in the network has to be assigned a combination of channel and timeslot such that there is no interference. Assigning timeslots and channels to links allows more concurrent transmissions than assigning timeslots and channels to nodes [29]. For good channel utilization and high network throughput, it is important to find a minimum length MAC layer schedule [30]. Finding an optimal MAC-layer schedule is NP-complete [31, 32, 30].

Cognitive radios have the ability to dynamically adapt to local spectrum availability. So in wireless mesh networks, in order to further increase channel capacity and mitigate the co-channel interference, impairments due to fading or delay-spread, cognitive radios with channel sensing could pick available channels, which could then be used in the allocation procedures.

Our Work and Contributions

In this thesis, we study the following specific problem: given a routing path P in a WMNs, equipped with CRs, choose which channels each link in P should use and their transmission schedule so as to maximize the end-to-end data flow rate (throughput) supported by the entire path. This problem is relevant to applications such as real-time streaming video or data where a connection may be long lasting and require a high constant throughput. This work is different from most previous works on transmission scheduling which usually deal with the problem of scheduling a set of links for link-layer throughput maximization. Here, we focus on end-to-end performance, and consider the problem of allocating resources (timeslots, channels) along a multi-hop routing path, which is a very hard problem due to the constraints related to intra-flow interference (interference among links belonging to a common flow) and inter-flow interference (interference among links belonging to different flows) [47]. We present a polynomial time constant-factor approximation algorithm for this problem. The constant is $\frac{1}{2(\Delta+1)}$ where Δ is the maximum degree of vertex found in certain subgraphs of the conflict-graph. A full definition is given in chapter 4, after the subgraphs in question are defined. If the path P satisfies certain properties, the constant factor is $\frac{1}{4}$. In practice, the algorithm typically achieves much better performance (90% of optimal or better). The algorithm can be extended to provide transmission schedules for a class of multipaths that can further improve the achieve end-to-end transmission rate.

The rest of the thesis is organized as follows. We discuss related work in Chapter 2. We formally define the problem in Chapter 3. We present the proposed algorithm

and analyze its performance in Chapter 4 and 5. Then we present the simulation environment and network model in Chapter 6, numerical results in Chapter 7 and conclude the thesis in Chapter 8.

RELATED WORK

Cognitive radio wireless networks have recently received extensive attention. In [49], the authors derived optimal and suboptimal distributed strategies for the secondary users to decide which channels to sense and access with the objective of throughput maximization under a Partially Observable Markov Decision Process (POMDP) framework. In [50], Zheng *et al.* developed a graph-theoretic model to characterize the spectrum access problem and devised multiple heuristic algorithms to find high throughput and fair solutions. In [48], the concept of a time-spectrum block was introduced to model spectrum reservation, and a centralized and a distributed protocol were presented to allocate such blocks for cognitive radio users. Tang *et al.* introduced a graph model to characterize the impact of interference and proposed joint scheduling and spectrum allocation algorithms for fair spectrum sharing based on it in [42]. In [36], a distributed spectrum allocation scheme based on local bargaining was proposed for wireless ad-hoc networks with cognitive radios.

Cross-layer schemes have also been proposed for cognitive radio wireless networks. In [38], the authors proposed the Asynchronous Distributed Pricing (ADP) scheme to solve a joint spectrum allocation and power assignment problem. In [44], Wang *et al.* presented a joint power and channel allocation scheme that uses a distributed pricing strategy to improve the network performance. In [46], a novel layered graph was proposed to model spectrum access opportunities, and was used to develop joint spectrum allocation and routing algorithms. In [45], the authors presented distributed algorithms for joint spectrum allocation, power control, routing and congestion control. A mixed integer non-linear programming based algorithm was presented to solve a joint spectrum allocation, scheduling and routing problem in [37]. A distributed algorithm was presented in [41] to solve a joint power control, scheduling and routing problem with the objective of maximizing data rates for a set of user communication sessions. In [43], a polynomial-time approximation scheme (PTAS) is presented for a more general maximum multiflow scheduling problem (maximize the total flow of

a set of commodities with no specific routing path) and several constant-factor approximations are given for special cases. This paper also points out some errors in previous work on that problem.

The differences between this work and previous works are summarized as follows: 1) we consider a channel assignment and scheduling problem for a given routing path in cognitive radio wireless mesh networks with heterogeneous channels with the objective of maximizing end-to-end throughput, which is different from those works addressing link layer (single-hop) throughput such as [36, 38, 42, 44, 48, 49, 50]. We propose provably good algorithms to solve the formulated problem. However, many related works (such as [36, 38, 44, 46]) only presented heuristic algorithms which cannot provide any performance guarantees.

PROBLEM DEFINITION

Wireless Mesh Networks (WMNs) are believed to revolutionize wireless internet connectivity by economically extending broadband internet over very large areas. In their default state, WMNs operate in the ISM band. Due to severe congestion in this unlicensed band, some of the constituent Mesh Routers (MRs) in a WMN may have over-congested operating conditions. Hence, it is desirable for such MRs to operate on a different band. There exist unused or white spaces in the spectrum used by licensed operators, who are referred to as Primary Users (PUs). Cognitive radio technology is one that can enable MRs in a WMN with poor channel conditions to shift operation to this licensed spectrum. This opportunistic sharing of the white spaces in the licensed spectrum results in higher spectral efficiency as well as reduced traffic volume in the ISM band. Most recent research only considers using omni-directional antenna at each Mesh Client (MC). This is inefficient, especially in the case of a Cognitive WMN (CWMN) as it shifts the operation of an entire cluster of nodes to the primary band and could induce inadmissible interference on the PUs. Our proposed channel transmission scheduling algorithm to increase end-to-end throughput in order to improve overall network performance and then we propose single and multipath models shows that our algorithm encourages load sharing, as well as increases throughput with efficient spectrum utilization.

We consider two related problems: scheduling transmissions along a path from a source node to a destination node so as to maximize the end-to-end throughput and a related problem of scheduling transmissions along a multipath from a source node to destination node. We first define our assumptions about the parameters of the cognitive radio wireless mesh network: Let m be the number of channels available in the network. In general, each link e_i will have only a subset of these channels available at any given time. This can be due to interference, the link distance being greater than the transmission range, or that channel is already in use on that link. We will also assume that each available channel j on link e_i has an associated bit rate $b_{e,j} \geq 0$. This bit rate can depend on the link distance and other factors.

We assume that communication in the network is done using synchronized transmission frames of a fixed length L , which means each connection lasts the same time period L and transmission starting times for each path are staggered by the same time period. We first examine a slightly idealized case where the transmission frame is infinitely divisible and then we will describe a simple rounding scheme to discretize our solution into integer numbers of time slots. Let C_e be the set of channels available to link e during the current frame. We define a variable $f_{e,j} \geq 0$ to indicate the flow amount allocated on the link e on channel j , where $j \in C_e$. A link flow $f_{e,j}$ is active if it is positive. An active link flow f must be scheduled at some point during the frame. To simplify discussion, we will assume that a scheduled link flow $f_{e,j}$ occupies a single continuous interval $[s_{e,j}, s_{e,j} + f_{e,j}) \subset [0, L)$, where $s_{e,j}$ indicates the starting time for the link flow.

We adopt the following simple interference model as shown in Fig. 7. We assume that there is an interference distance R_j for each channel j such that a link $e = (u, v)$ *interferes* with another link $e' = (u', v')$ on channel j if and only if $|u - v'| \leq R_j$ or $|u' - v| \leq R_j$. We will also consider that the nodes in question are half-duplex.

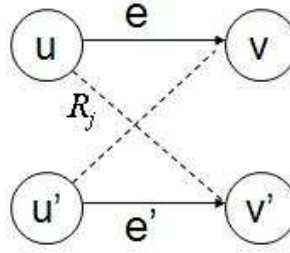


Figure 7. Interference model.

A half-duplex system provides for communication in both directions, but only one direction at a time (not simultaneously). Typically, once a party begins receiving a signal, it must wait for the transmitter to stop transmitting, before replying. It is a TDD (the application of time-division multiplexing to separate outward and

return signals. It emulates full-duplex communication over a half-duplex communication link) that has been used a lot in many radio systems, such as 802.16. For example, on a local area network using a technology that has half-duplex transmission, one workstation can send data on the line and then immediately receive data on the line from the same direction in which data was just transmitted. On the other hand, full-duplex transmission implies that data are transmitted in both directions simultaneously.

The duplexing and interference constraints impose conditions on which link flows can be active at the same time. We will summarize these conditions in a well-known *conflict graph*, $G_c = (V_c, E_c)$, where the vertices V_c are the link flow variables $f_{e,j}$ and the edges (undirected) indicate which pairs of link flows that cannot be scheduled simultaneously due to interference or duplexing constraints. For a transmission schedule to be *valid*, it must not contain any pair of conflicting scheduled link flows at any time; i.e. for any two active link flows $f_{e,j}$ and $f_{e',k}$, $(f_{e,j}, f_{e',k}) \in E_c \Rightarrow [s_{e,j}, s_{e,j} + f_{e,j}) \cap [s_{e',k}, s_{e',k} + f_{e',k}) = \emptyset$.

We can now formalize the two problems considered as follow:

Max-Path-Bitflow

We begin with the path scheduling problem: A selected path $P = v_0, \dots, v_n$ from $v_0 = s$ to $v_n = t$ is composed of path links $e_i = (v_{i-1}, v_i)$. We wish to find the highest possible bit flow along the path from the source s to the destination t . A crucial requirement is that channel assignment along the path is actually a flow in that the bit flow entering any intermediate vertex on the path is equal to the bit flow leaving the vertex (conservation of flow). This leads to the following constraint (conservation of bit flow):

$$F = \sum_j b_{e_{i-1},j} f_{e_{i-1},j} = \sum_j b_{e_i,j} f_{e_i,j}; \quad 1 \leq i \leq n \quad (3.1)$$

where F is the total bit flow along the path. We are interested in the following optimization problem:

Max-Path-Bitflow: Find a valid transmission schedule S that maximizes F subject to (3.1).

Max-Multipath-Bitflow

Consider the network consisting of a set of links and a set of routes, where each route is a collection of links. There could be multiple routes between each source-destination pair. Multipath routing enables a network's traffic to be split among two or more possibly disjoint paths in order to reduce latency, improve throughput, and balance traffic loads. A example of multipath routing in network shown in Fig. 8

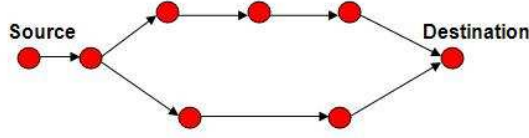


Figure 8. Example of multipath routing.

The multipath scheduling problem is defined similarly. A multipath $M = (V_M, E_M)$ from s to t is a subgraph of G such that $s, t \in V_M$ and for any $v \in V_M$, there exists a simple path in M from s to t that goes through v as an intermediate node. The problem we consider is to find a valid transmission schedule for the links in E_M that maximizes the total bit flow from s to t using links belonging to M . Let $v_{in} \subset E_M$ be the set of incoming edges to $v \in V_M$ and let v_{out} be the set of outgoing edges from v . A crucial requirement is that scheduled bit flow entering any interior vertex on the path is equal to the bit flow leaving that vertex. This leads to a constraint similar to (3.1):

$$\sum_{e \in v_{in}} \sum_j b_{e,j} f_{e,j} = \sum_{e \in v_{out}} \sum_j b_{e,j} f_{e,j}; \quad \forall v \in V_M \setminus \{s, t\} \quad (3.2)$$

Let

$$F = \sum_{e \in t_{in}} \sum_j b_{e,j} f_{e,j} \quad (3.3)$$

be the total bit flow into the destination node t (this is also equal to the total bit flow out of s).

Max-Multipath-Bitflow: Find a valid transmission schedule S that maximizes F (defined by (3.3)) subject to (3.2).

In the next section we present an algorithm for finding solutions to the Max-Path-Bitflow problem as well as a special case of the Max-Multipath-Bitflow where we assume that the multipath M has a chromatic index of 2. This means that the edges of E_M can be colored with two colors such that no two adjacent edges share the same color.

THE SCHEDULING ALGORITHM AND ANALYSIS

In this section we describe a general scheduling algorithm for both the Max-PATH-Bitflow problem and a special case of the Max-MULTIPATH-Bitflow problem where we assume that the multipath M has a chromatic index of 2; such multipaths have the property that their links can be partitioned into two sets E_M^o (odd) and E_M^e (even) such that for each $v \in V_M$, either $v_{in} \subset E_M^o$ and $v_{out} \subset E_M^e$, or $v_{in} \subset E_M^e$ and $v_{out} \subset E_M^o$. We will assume that $s_{out} \subset E_M^o$. We call such multipaths *odd-even*. Note that a simple transmission path P from s to t can be considered an odd-even multipath. Because of this, we will frame the scheduling algorithm in terms of scheduling transmissions on odd-even multipaths.

Greedy Colorings

Graph coloring is a special case of graph labeling in graph theory. It is an assignment of labels traditionally called “colors” to elements of a graph subject to certain constraints. In its simplest form, it is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color; this is called a vertex coloring. Similarly, an edge coloring assigns a color to each edge so that no two adjacent edges share the same color, and a face coloring of a planar graph assigns a color to each face or region so that no two faces that share a boundary have the same color. The terminology of using colors for vertex labels goes back to map coloring.

In the study of graph coloring problems in mathematics and computer science, a greedy coloring is a coloring of the vertices of a graph formed by a greedy algorithm that considers the vertices of the graph in sequence and assigns each vertex its first available color. Graph coloring is NP hard (non-deterministic polynomial-time hard). NP hard in computational complexity theory, is a class of problems that are, informally, “at least as hard as the hardest problems in NP”. A problem H is NP-hard if and only if there is an NP-complete problem L that is polynomial time

Turing-reducible to H. In other words, L can be solved in polynomial time by an oracle machine with an oracle for H. Informally, we can think of an algorithm that can call such an oracle machine as a subroutine for solving H, and solves L in polynomial time, if the subroutine call takes only one step to compute

Graph coloring does not in general use the minimum number of colors possible. However it has been used in mathematics as a technique for proving other results about colorings and in computer science as a heuristic to find colorings with few colors.

The greedy algorithm considers the vertices in a specific order $v_1, \dots, v_n$ and assigns to v_i the smallest available color not used by v_i 's neighbours among $v_1, \dots, v_{i-1}$, adding a fresh color if needed. The quality of the resulting coloring depends on the chosen ordering. There exists an ordering that leads to a greedy coloring with the optimal number of (G) colors. On the other hand, greedy colorings can be arbitrarily bad; for example, the crown graph on n vertices can be 2-colored, but has an ordering that leads to a greedy coloring with $n/2$ colors.

If the vertices are ordered according to their degrees, the resulting greedy coloring uses at most $\max_i \min\{d(x_i) + 1, i\}$ colors, at most one more than the graph's maximum degree. This heuristic is sometimes called the Welsh–Powell algorithm [36]. Another heuristic establishes the ordering dynamically while the algorithm proceeds, choosing next the vertex adjacent to the largest number of different colors [36]. Many other graph coloring heuristics are similarly based on greedy coloring for a specific static or dynamic strategy of ordering the vertices, these algorithms are sometimes called sequential coloring algorithms.

Odd-Even Multipath Scheduling

The general approach will be to use graph coloring to identify individual link flows that can be scheduled simultaneously and then use linear programming (a mathematical method for determining a way to achieve the best outcome in a given mathematical

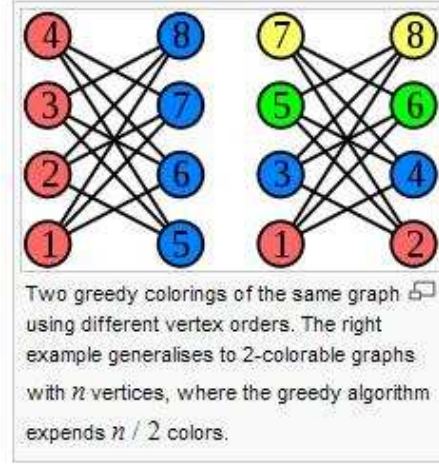


Figure 9. Example of Greedy Coloring [33].

model for some list of requirements represented as linear equations) to determine optimal link flow values and build a transmission schedule for the active link flows that maximizes the end-to-end throughput along the routing path P or multipath M .

A novel aspect of this approach is that we separately schedule the link flows that belong to E_M^o (odd links) and those that belong to E_M^e (even links). Note that if incoming links to a vertex are odd, then its outgoing links must be even and vice-versa. For each channel, we consider the subgraph of odd (even) link flows using that channel and use a greedy strategy to color these subgraphs. Link flows belonging to the same subgraph that receive the same color are scheduled simultaneously and these schedules can be superimposed to create an overall link flow schedule. Let G_c^o be the conflict graph restricted to odd flows and let G_c^e be the conflict graph restricted to even flows. An important observation is that duplexing conflicts disappear in these subgraphs; *this means that all conflicts are between link flows on the same channel*. Thus, these subgraphs both further decompose into disjoint components, one for each channel j available: $G_{c,j}^e$ and $G_{c,j}^o$. Using these subgraphs, we can now state the scheduling algorithm, termed OddEvenSchedule, for the real-valued version of the problem. The pseudo code is given in Algorithm 1.

Algorithm 1 OddEvenSchedule

- Step 1 For all channels j , color both $G_{c,j}^o$ and $G_{c,j}^e$ using a simple greedy algorithm. Suppose g_j and h_j colors are used respectively.
- Step 2 Associate *color variables* $o_1^j, \dots, o_{g_j}^j$ to the colors used in coloring $G_{c,j}^o$ and $e_1^j, \dots, e_{h_j}^j$ to the colors used in coloring $G_{c,j}^e$.
- Step 3 Solve the following linear program (LP):
1. $0 \leq x \leq L$
 2. $o_k^j \geq 0, \sum_{k=1}^{g_j} o_k^j \leq x; \forall j$
 3. $e_k^j \geq 0, \sum_{k=1}^{h_j} e_k^j \leq L - x; \forall j$
 4. Each odd link flow $f_{e,j}$ was given a color with associated variable o_k^j in Step 1. Add the constraint $0 \leq f_{e,j} \leq o_k^j$ to the LP. Likewise, each even link flow $f_{e,j}$ was associated with a color variable e_k^j . Add the constraint $0 \leq f_{e,j} \leq e_k^j$ to the LP.
 5. Include the conservation-of-flow constraint given by (3.2).
 6. Maximize F as defined in (3.1).
- Step 4 For each channel j , we define the odd starting times as follows: Let $s_1^j = 0$ and $s_k^j = s_{k-1}^j + o_{k-1}^j$ for $1 < k \leq g_j$. Similarly, the even starting times are defined by $t_1^j = x$ and $t_k^j = t_{k-1}^j + e_{k-1}^j$ for $1 < k \leq h_j$.
- Step 5 Create a schedule S for the time frame with the following rule: An odd link flow $f_{e,j}$ associated with color variable o_k^j will be active in the interval $[s_k^j, s_k^j + f_{e,j})$. An even link flow $f_{e,j}$ associated with color variable e_k^j will be active in the interval $[t_k^j, t_k^j + f_{e,j})$.
-

The main idea of the algorithm is to divide the frame schedule so that the odd flows occur at the beginning of the frame and even flows occur at the end. This approach immediately satisfies all duplexing constraints along the path. It remains to ensure that for each channel, all conflicting link flows on that channel are scheduled at different times. This is done by coloring the subgraphs $G_{c,j}^e$ and $G_{c,j}^o$ described above and further dividing the frame into non-overlapping intervals for scheduling the link flows of each color (the intervals for different colors can overlap). In Step 3, the algorithm solves a linear program (LP) to find the optimal link flows subject to

the color interval conditions. Steps 4 and 5 create the frame schedule from the LP solution.

Example Path Transmission Schedule

To illustrate the OddEvenSchedule algorithm, we consider the following example transmission path consisting of five links. Suppose that the available channels and transmission rates are given for each link in this path in the following table:

Link	Available channels	Capacity
1	{5, 6}	10 Mbps
2	{1, 3, 5, 11}	10 Mbps
3	{2, 4, 5, 12, 14, 15}	5 Mbps
4	{1, 2, 3, 12, 13}	50 Mbps
5	{2, 3, 4, 12, 13, 14}	5 Mbps

The structure of the $G_{c,j}^o$ and $G_{c,j}^e$ subgraphs are both quite simple in this example.

The only non-trivial conflict subgraphs are given below:

Odd subgraphs	Even subgraphs
$G_{c,2}^o = f_{3,2} \longleftrightarrow f_{5,2}$	$G_{c,1}^e = f_{2,1} \longleftrightarrow f_{4,1}$
$G_{c,4}^o = f_{3,4} \longleftrightarrow f_{5,4}$	$G_{c,3}^e = f_{2,3} \longleftrightarrow f_{4,3}$
$G_{c,5}^o = f_{1,5} \longleftrightarrow f_{3,5}$	
$G_{c,14}^o = f_{3,14} \longleftrightarrow f_{5,14}$	

Clearly, each of these subgraphs can be colored with two colors. OddEvenSchedule (followed by the rounding procedure described below in Chapter 4.5) produces the frame schedule shown in Fig. 10. In the solution to the LP of Step 3 of the algorithm, the variable x was rounded to 0.69 (the frame length $L = 1$ s in this example and the frame is divided in 100 time slots). As can be seen in the schedule, active link flows $f_{1,5}$ and $f_{3,5}$ were given different colors and thus not scheduled at the same time. The same is true for link flows $f_{3,14}$ and $f_{5,14}$. This schedule achieves a throughput of 12.4 Mbps (we can show using an upper bound calculation from Chapter 6 that this is at least 93% of optimal).

Analysis

It is clear by construction that this transmission schedule is valid, since no conflicting links are scheduled at the same time. Let F_S be the multipath bit flow obtained

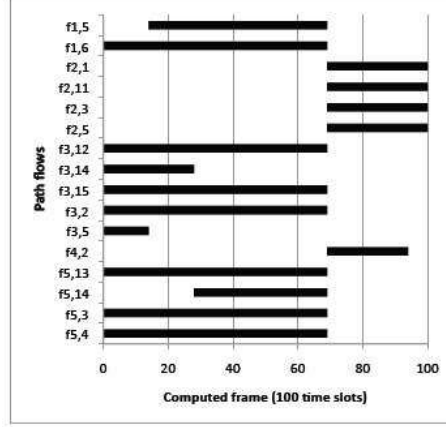


Figure 10. The (rounded) frame schedule produced for the Example transmission path from Section 4.

by the schedule created by OddEvenSchedule and let F_{S^*} be the multipath bit flow obtained by an optimal schedule S^* . We say a transmission schedule is *equal odd-even* if and only if link flows along odd edges are scheduled in $[0, L/2)$ and link flows along even edges are scheduled in $[L/2, L)$.

Lemma 1: Let S^{*oe} be an optimal equal odd-even schedule for the multipath M with associated bit flow $F_{S^{*oe}}$. Then $F_{S^{*oe}} \geq \frac{1}{2}F_{S^*}$.

A simple way to see this is to take the schedule S^* and scale it by $1/2$ to create a schedule for the half-frame $[0, L/2)$. Place another copy of the half-scaled S^* in the interval $[L/2, L)$. Delete the even flows in the first copy and odd flows in the second copy. The resulting schedule is now equal odd-even with bit flow value $\frac{1}{2}F_{S^*}$, so the optimal equal odd-even schedule will be at least this value.

It is well-known that the greedy coloring algorithm used in Step 1 provides a coloring that uses $\Delta(G) + 1$ colors, where $\Delta(G)$ is the maximum degree of a vertex in the input graph G . Let $\Delta = \max\{\Delta(G_{c,j}^o), \Delta(G_{c,j}^e)\}$. Then for all j , $g_j, h_j \leq \Delta + 1$.

Lemma 2: $F_S \geq \frac{1}{\Delta+1}F_{S^{*oe}}$

Since S^{*oe} is an equal odd-even schedule, each link flow $f_{e,j} \leq L/2$. Scale each link flow in S^{*oe} by $\frac{1}{\Delta+1}$. The resulting link flows now satisfy $0 \leq f_{e,j} \leq \frac{L}{2(\Delta+1)}$. Letting $x = L/2$ and $o_k^j = e_k^j = \frac{L}{2(\Delta+1)}$ yields a feasible solution to the LP in OddEvenSchedule with total bit flow $\frac{1}{\Delta+1}F_{S^{*oe}}$. It follows that F_S will be at least this value.

We next observe the running time of OddEvenSchedule is polynomial: Step 1 invokes a standard greedy coloring algorithm that runs in $O(|V_c| + |E_c|)$ time. Steps 2 and 3 solve a linear program with $O(|V_c|)$ variables and constraints (note: the number of color variables is bounded by $|V_c|$ since giving each link flow its own unique color is a trivially valid coloring). This can be solved in $O(|V_c|^{3.5}L)$ using Karmarkar's algorithm (L is the number of bits used to represent the input). Steps 4 and 5 require $O(|V_c|)$ time to create a valid link flow schedule. This leads to the following theorem.

Theorem 1: OddEvenSchedule is a $\frac{1}{2(\Delta+1)}$ -approximation algorithm running in polynomial time.

From Lemmas 1 and 2, OddEvenSchedule produces a schedule S that satisfies $F_S \geq \frac{1}{\Delta+1} F_{S^{*oe}} \geq \frac{1}{2(\Delta+1)} F_{S^*}$.

Scheduling Typical Paths

We provide additional analysis of a typical case when the routing path is loop-free and well-spaced, as defined below.

Definition 1: We say that the routing path P is **loop-free** if all of the subgraphs $G_{c,j}^e$ and $G_{c,j}^o$ are interval graphs.

This means that for each channel j , the odd (even) links along P can be placed on the real line such that two links conflict if and only if their corresponding intervals overlap. This is the case if P does not contain any large loops. A useful property of interval graphs is that they can be colored optimally using the simple greedy strategy of Step 1 [40]. Let $\chi = \max\{\chi(G_{c,j}^e), \chi(G_{c,j}^o)\}$, where $\chi(G)$ is the *chromatic number* of G .

Definition 2: A path P is **well-spaced** if and only if for all channels j , if $f_{e_i,k}$ conflicts with $f_{e_j,k}$, then $|i - j| < 4$.

This condition implies that the interval graphs above have a simple structure; each interval has at most one overlapping left and right neighbor. Such graphs can be colored using just two colors, so $\chi = 2$ and this coloring is found by the greedy algorithm of Step 1.

Corollary 1: If the routing path P is loop-free and well-spaced, then OddEvenSchedule is a $\frac{1}{4}$ -approximation algorithm.

A Simple Rounding Scheme

In this section, we describe a rounding scheme to round the real-valued link flow schedule obtained by OddEvenSchedule into a integer-valued schedule. Suppose that the transmission frame $[0, L)$ is evenly divided into r time slots each of size L/r . We seek a transmission schedule where active link flows are allocated a specific interval of time slots. The potential flow rate for a link is determined by how many time slots are allocated to it. With discrete time slots, the bit flow coming into a vertex may not exactly balance the bit flow going out. To address this, we interpret $f_{e,j}$ as the *effective flow* on link e , channel j . Let $t_{e,j}$ be the number of time slots allocated to link e , channel j . Then,

$$f_{e,j} \leq t_{e,j} L/r; \quad \forall e, j \quad (4.1)$$

In general, the optimal solution to the maximum bit flow problem with integer time slots can be found with integer linear programming. Instead, we will simply round the real-valued solution produced by OddEvenSchedule to generate an integer solution. A *time slot boundary point* has the form $l \frac{L}{r}$ for some integer $0 \leq l \leq r$. We round the starting and ending times of all active flows to time slot boundaries, as follows: First round x to the nearest time slot boundary point. Next we round the flows on each channel separately. To round the odd flows on channel j , we simply round each odd starting time s_k^j to its nearest boundary point $s_k'^j$. A link flow $f_{e,j}$ associated with color variable o_k^j is now allocated all time slots in the interval $[s_k'^j, s_{k+1}'^j]$ (let $s_{g_j+1}'^j = x'$). Let the resulting rounded link flow be $f_{e,j}'$. The even flows are rounded similarly and this creates a rounded schedule S' .

For path transmissions, we can bound the end-to-end throughput loss that occurs from rounding. Let $F_{S'}$ be the value of the objective function after rounding. Let $B = \max_{i=1}^n \sum_{j \in C_i} b_{e,j}$.

Lemma 3: For a transmission path P , $F_{S'} \geq F_S - B \frac{L}{r}$.

The starting time and available ending time for any active link flow is moved by at most $\frac{1}{2} \frac{L}{r}$ due to rounding. Thus any link flow is reduced by at most $\frac{L}{r}$ after rounding. Hence the total flow on any link e_i in P is reduced at most $\sum_{j \in C_i} b_{e,j} \frac{L}{r} \leq B \frac{L}{r}$. Let F^* be the optimal value of original real-valued flow schedule and let F'^* be the optimal value of (3.3) when the links are allocated integer time slots and the constraints given by (4.1) are in place. Clearly, $F'^* \leq F^*$. From Lemma 3 and Theorem 1, we have

$$F'_S \geq F_S - B \frac{L}{r} \geq \frac{1}{2(\Delta + 1)} F'^* - B \frac{L}{r}. \quad (4.2)$$

The losses due to rounding can be greater for multipath transmissions (the effective bit flow is limited by the maximum rounding loss attained across any cut separating s from t in the multipath transmission graph M).

FINDING ROUTING PATHS AND MULTIPATHS

To test our transmission scheduling algorithm, we consider two approaches for generating an end-to-end transmission path P from a source node s to a destination node t . The first approach is simply to compute the path that uses the least number of links (a shortest path). This can be done using Dijkstra's algorithm. Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1959 [51], is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing. An equivalent algorithm was developed by Edward F. Moore in 1957 [52]. The second approach attempts to find a transmission path whose links all have high estimated capacity. The details of this approach will be given below. We considered a single heuristic approach to creating a transmission multipath M based on a variation of depth-first search in which links are considered in order of their capacity and direction relative to the destination node. We provide the details of these two novel algorithms below.

Routing to Maximize Path Bottleneck Capacity

We define the capacity $c(e)$ of a link e as follows:

$$c(e) = \sum_{j \in C_e} b_{e,j}. \quad (5.1)$$

The link capacity $c(e)$ provides an upper bound on the bit flow rate achievable by link e ignoring intra-path interference. We define the *bottleneck capacity* of a path P to be $c(P) = \min_{e \in P} c(e)$. Our goal is to find a path P that maximizes $c(P)$. This is a well-known problem that can be efficiently solved by computing a minimum spanning tree T (a spanning tree with weight less than or equal to the weight of every other spanning tree) on the network graph using an edge weight function $w(e) = -c(e)$. The unique path in T from s to t will have maximum bottleneck capacity.

DFS Multipath RoutingDepth first search(DFS)

Depth-first search (DFS) is an algorithm for traversing or searching a tree, tree structure, or graph. It is an uninformed search that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hasn't finished exploring. In a non-recursive implementation, all freshly expanded nodes are added to a stack for exploration.

For the following graph:

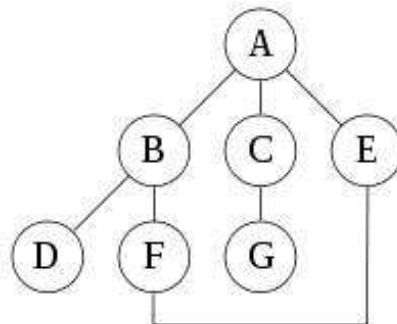


Figure 11. An example of DFS.

A depth-first search starting at A, assuming that the left edges in the shown graph are chosen before right edges, and assuming the search remembers previously-visited nodes and will not repeat them (since this is a small graph), will visit the nodes in the following order: A, B, D, F, E, C, G.

Multipath Routing

To investigate the performance of scheduling multipath routes, we adopted a simple multipath construction algorithm based on performing a depth first search (DFS) in G starting from the source node s . As subpaths to the destination node t are discovered, they are added to the multigraph M . The OddEvenSchedule algorithm requires that M be odd-even. The algorithm assigns a parity (odd or even) to each visited vertex v with the convention that all outgoing links from v have the same parity and all incoming links have the opposite parity. The source vertex s is assigned odd parity. During a DFS from a vertex u , if edge (u, v) is followed and v is already part of M , a check is made to see that u and v have opposite parities; if not, that branch of the search fails and a different outgoing edge from u must be tried (if one exists). If u and v have opposite parities then the new subpath reaching v is added to M . This ensures the constructed multipath M is odd-even. If a search path fails to reach t or a compatible existing path to t , the algorithm backtracks (clearing parity assignments as it goes back). A key decision to make while performing this search is the order in which outgoing edges are explored from any intermediate vertex u . This effects which subpaths get added to M . We use a simple heuristic rule to rank outgoing edges $e = (u, v)$ based on their capacity as well as their direction relative to the destination vertex v . Let θ_e be the angle between (u, v) (considering the link as vector) and the vector (u, t) . Then the rank of an edge is defined by

$$r(e) = c(e) \cdot \cos(\theta_e). \quad (5.2)$$

We presort the adjacency lists of outgoing edges for each vertex u into decreasing order of $r()$ -value and then conduct the depth-first search described above from the source vertex s . Once a DFS branch reaches t , that subpath is added to M . This continues until no new subpaths are discovered and DFS terminates. At this point, the odd-even multipath M is fully constructed.

SIMULATION ENVIRONMENT AND NETWORK MODEL

Simulation Software

In this thesis, we used lp-solve 5.5.0.15 to implement the proposed algorithm in Microsoft Visual C++ 6.0. Lp represents the Linear Programming, a problem is mathematically formulated as follows:

- A linear function to be maximized or minimized

e.g.

$$\text{maximize } c_1x_1 + c_2x_2$$

- Problem constraints of the following form

e.g.

$$a_{11}x_1 + a_{12}x_2 \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 \leq b_2$$

$$a_{31}x_1 + a_{32}x_2 \leq b_3$$

- Default lower bounds of zero on all variables.

The problem is usually expressed in matrix form, and then becomes:

$$\text{maximize } C^T x$$

$$\text{subject to } A x \leq B$$

$$x \geq 0$$

A linear programming model consists of one objective which is a linear equation that must be maximized or minimized. There are a number of linear inequalities or constraints.

C^T , A and B are constant matrices. x are the variables (unknowns). All are real, continue values.

Note the default lower bounds of zero on all variables x. People tend to forget this build-in default. If no negative (or negative infinite) lower bound is explicitly set on variables, they can and will take only positive (zero included) values.

Lp-solve is a free linear (integer) programming solver based on the revised simplex method and the branch-and-bound method for the integers. It was originally developed by Michel Berkelaar at Eindhoven University of Technology. Lp-solve has no limit on model size and accepts both standard lp or mps input files, but even that can be extended. Note however that some models could give lp-solve a hard time and will even fail to solve. The larger the model the more likely the chance failure. But even commercial solvers have problems with large problems.

Lp-solve can also be called as a library from different languages like C, VB, .NET, Delphi, Excel, Java, etc, and it can also be called from AMPL, MATLAB, O-Matrix, Scilab, Octave, R via a driver program. lp-solve is written in ANSI C and can be compiled on many different platforms such as linux and WINDOWS.

Basically, lp-solve is a library, a set of routines, called the API that can be called from almost any programming language to solve MILP problems. There are several ways to pass the data to the library:

- Via the API:

The API is a set of routines that can be called from a programming language to build the model in memory, solve it and return the results. There are many API routines to perform many possible tasks and set several options.

- Via input files:

Standard, lp-solve supports several input files types. The common known MPS format is supported by most solvers, but it is not very readable for humans. Another format is the lp format that is more readable. lp-solve has the unique ability to use user-written routines to input the model. See read-mps, read-freemps, read-MPS, read-free MPS and read-lp, read-LP for the API calls to read the model from file.

- Via an IDE:

Thanks to Henri Gourvest, there is now also an IDE program called LPSolve IDE that uses the API to provide a Windows application to solve models. See LPSolve IDE for its usage. With this program you don't have to know anything

of API or computer programming languages. You can just provide your model to the program and it will solve the model and give you the result.

As already stated, lp-solve can be called from many programming languages. Among them are C, C++, Pascal, Delphi, Java, VB, C, VB.NET, Excel. But let this list not be a limitation. Any programming language capable of calling external libraries (DLLs under Windows, Shared libraries (.so) under Unix/Linux) can call lp-solve.

Simulation Model

We tested the performance of our path and multipath scheduling algorithm using the routing algorithms described in Chapter 5. Specifically, we used Dijkstra's algorithm to compute shortest path routes and the algorithm from Chapter 5 to compute maximum bottleneck capacity routes. Other path routing approaches have been suggested such as [39]. We used the DFS-based approach described in Chapter 5 to compute multipath routes.

To estimate how close to optimal the schedules produced by OddEvenSchedule are in practice, we can compute an upper bound on the optimal transmission schedule by solving the following LP:

-
1. $\forall$ clique¹ $C \in G_c$, add the constraint

$$\sum_{f_{e,j} \in C} f_{e,j} \leq L \tag{6.1}$$

to the LP.

2. Include the conservation-of-flow constraint given by (3.1).
 3. Maximize F as defined in (3.1).
-

¹A subgraph in G_c with all possible edges present.

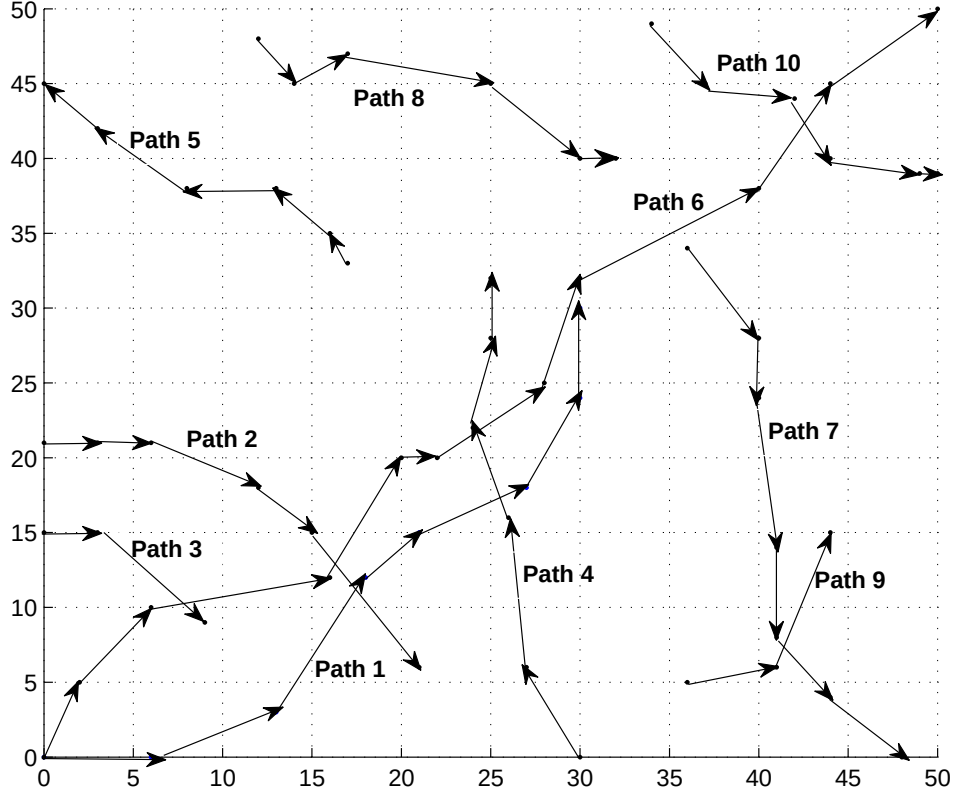


Figure 12. The simulated network ($50 \text{ km} \times 50 \text{ km}$) and connections. Only the shortest paths are shown in this figure (not the bottleneck and multipath).

Clearly, this LP provides an upper bound on the optimal transmission schedule since any transmission schedule must satisfy (6.1) for each clique present in the conflict graph. In practice, to reduce running time, we consider cliques up to size 5 and 6; in some case, including larger cliques may improve the computed bound slightly.

All of our numerical results were gathered using the example network shown in Figure 12. This network was created by placing 90 nodes at random locations in a $50 \text{ km} \times 50 \text{ km}$ grid area. Given a source and destination pair, we can obtain the routing path by using different methods, and the position of nodes on the path can decide the link length and how much intra-flow interference each link will have during

transmission time. If the position of node moves, the data rate on each link may vary a lot as well as the interference, and the end-to-end throughput will no longer stay the same.

For our experiments, we assumed there were a maximum of 13 channels available per frequency and that the link throughput for each channel was the maximum available given the link distance and frequency used. The following Tables I and II summarize our assumptions about the transmission rates and interference ranges of each frequency.

From both table 1 and table 2 we can see that we have three groups of frequency: 700 MHz, 2400 MHz and 5800 MHz. Because 2.4 GHz and 5 GHz band are heavily used by various wireless devices, which can give us some primary users and results in interference to our network performance.

700 MHz is usually used for TV broadcasting. The FCC pushed for global adoption of the 700 MHz band as the home of future mobile broadband technologies. And, it seems that much of the globe has agreed to support the 700 MHz spectrum for next-generation wireless services. All of the Americas, China, India, Europe, South Korea, and Japan are on board to utilize the upcoming spectrum. As the new globally-accepted spectrum for future wireless technologies, the 700 MHz frequency is sure to be an even more sought-after slice of the radio spectrum.

For 2400 MHz, it can be used for cell phone. The low end of this band (2390-2400) is designated for amateur radio usage and appears unused during the duration of the survey. The largest portion of this band is the Industrial, Scientific, and Medical (ISM) band (2400-2483.5 MHz). This is allocated for unlicensed low power spread spectrum communication systems. The power limit of one watt limits the range of such devices to approximately 100 feet. Devices in this band include microwave ovens, cordless phones, wireless networking and wireless instrumentation devices. The spectrum plot indicates that the band is fully used, but the occupancy (duty-cycle) figure is low. That may be due to the location of the receiver 200 feet above the ground. Also there may have been low activity in the time period under study.

For 5800 MHz, it can be used for video transmitter, it allocated in 2003 due to crowding on the 2.4 GHz band. Since the 900 MHz and 2.4 GHz band are increasingly being used for a host of other devices, including baby monitor, microwave oven, Bluetooth, wireless LAN; thus, it is likely that a cordless phone will suffer interference from signals broadcast by those devices. It is also possible for a cordless phone to interfere with the 802.11a wireless standard, as the 802.11a standard can be configured to operate in the 5.8GHz range. However, this can easily be fixed by configuring the device to work in the 5.180 GHz to 5.320 GHz band.

From the table 2 we can see that the interference range depends on the frequency. The interference range is typically assumed to be twice the maximum transmission range. This can be calculated for each of the three frequencies, assuming that the interference would be applied to the lowest channel rate (worst case) or to the highest channel rate (shortest). By assuming the shortest distance, an interferer would add sufficient noise to a 45Mb/s link to reduce it to a 20Mb/s link. And the higher the frequency is, the smaller the interference range. So if the interference range for 700 MHz become less, more channels will be available on this frequency and the throughput will as well.

These values are based on a scenario where each node transmits at 1W with a 2dBi antenna and the receiving mode antenna has a gain of 2dBi. The channel bandwidth is 10 MHz, the receiver noise figure is 5dB, and implementation losses of 3dB are assumed for each link. Path loss is calculated using line of sight and free space characteristics.

Typical 802.16 adaptive modulation and coding parameters performance parameters are used to estimate the throughput achievable as a function of CNR (carrier to noise ratio), and are then translated into the allowable path loss threshold. The maximum channel transmission rate is a function of distance and frequency (at lower frequency, the maximum distance for a given transmission rate will be greater). The table below gives maximum distances for each transmission rate, for each of three frequencies: 700 MHz, 2400 MHz and 5800 MHz.

We also varied the number of primary users in the network (not shown). Primary users were placed at random locations and assigned a random channel. This channel was then made unavailable to any link within the interference range of the primary user. The number of primary users was set to be one third of the number of available channels. For different number of primary users, the end-to-end throughput may be different, the more the primary users, the better the throughput could be. The frame length L was set to 1 second and each frame was divided into $r = 100$ time slots.

Table 1. Maximum transmission distances by frequency and data rate.

Transmission rate	700 Mhz	2400 Mhz	5800 Mhz
45 Mbps	15.4 km	4.5 km	1.8 km
40 Mbps	18.4 km	5.3 km	2.2 km
30 Mbps	30 km	8.6 km	3.6 km
20 Mbps	41 km	11.8 km	4.9 km
10 Mbps	68 km	20 km	8.2 km

Table 2. Interference ranges by frequency.

Frequency	Interference range
700 Mhz	30.8 km
2400 Mhz	9 km
5800 Mhz	3.6 km

NUMERICAL RESULTS

Scenario 1: Performance on random paths

In this scenario, ten random source-destination pairs were created and routing paths found using different approaches. Each connection lasts for a 200 second time period. The transmission starting times for each path are staggered by 100 seconds. So the second connection start from 100 seconds, the last connection start from 1000 seconds and lasts 200 seconds, and the total duration of the simulation is 1100 seconds. For example, path 1 is active in the time interval $[0, 200]$, Path 2 is active in the time interval $[100, 300]$, etc. Path 2, it will get interference from path 1 in the time interval $[100, 200]$ and at the same time, path 2 will interfere with the transmission of path 3 in the time interval $[200, 300]$. The number of available channels was set to 15, and according to Table 1, there are three frequency groups, each of which has 5 available channels. Meanwhile, there are 5 primary users that were placed at random locations in the grid. And the transmission rate and interference range can be found from the tables shown before. The clique size of the upper bound were set to 5 and 6 to study the performance of our algorithm.

Shortest Path Routing Performance

In this subsection, the routing paths are found by running Dijkstra's algorithm, and then we compare the performance results generated by our Algorithm 1 with the upper bound values. The results are shown in Figure 13. As can be seen in the figure, there is fairly large variation in the end-to-end throughputs achieved that correlates with path length. The end-to-end throughput varies a lot based on nodes position and length of the path. The longer the path is, the less throughput it will achieve. And we set the clique size value of the upper bound to 5 and 6 separately. In all cases, the schedule produced by OddEvenSchedule is within 90% of the upper bound

for the optimal solution computed by the above LP described at the beginning of Section 7.

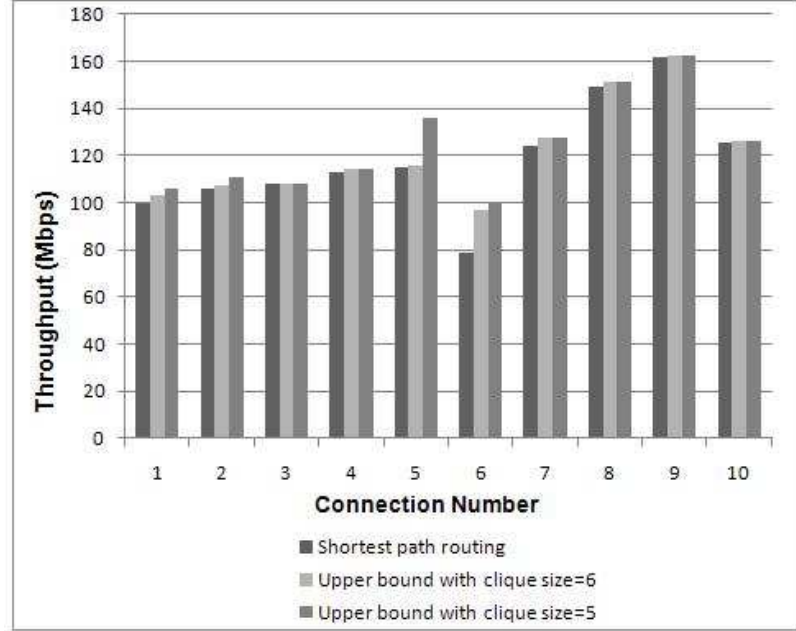


Figure 13. Shortest Path Routing Performance for each path shown in Figure 12.

Bottleneck Path Routing Performance

In this subsection, the routing paths are selected by the function of capacity shown in chapter 5: $c(e) = \sum_{j \in C_e} b_{e,j}$. Our goal is to find a path P that maximizes $c(P)$. This is a well-known problem that can be efficiently solved by computing a minimum spanning tree T on the network. To realize it, we use a binary search approach to obtain the single path for each set of source and destination nodes and then run our algorithm 1 to achieve the maximum bit flow on each path. The results are shown in Figure 14.

As can be seen in the figure, like the results with shortest path, there is also a fairly large variation in the end-to-end throughputs achieved that correlates with path length, and the results from our algorithm are close to the upper bound values.

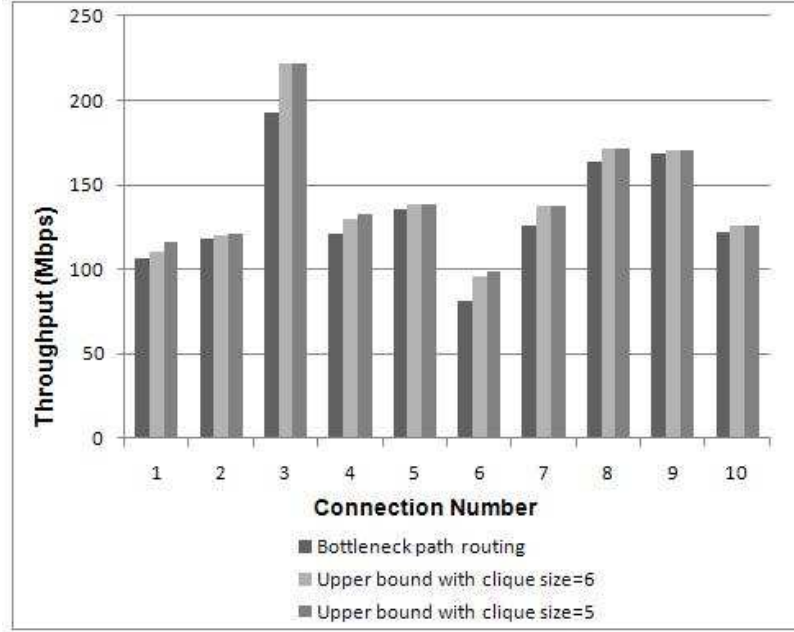


Figure 14. Bottleneck Path Routing Performance.

Multipath Routing Performance

In this subsection, we created a transmission multipath based on a variation of depth-first search in which all links are considered in order of their capacity and direction relative to the destination node. The clique size value is set to 5 and 6.

From the figure below, we can see that our algorithm still works very well, and the end-to-end throughput varies a lot based on position and length of the path. The longer the path is, the less throughput it will achieve. Also, since for each source and destination pair, there is more than one path, more traffic can be sent from the source node, and as a result, the throughput improves relative to that single path.

Scenario 2: Comparison of the three approaches

In this scenario, we compare the results of the approaches described in the previous section, the one with Dijkstra's algorithm, the one that to maximum bottleneck capacity and the multipath routing case. We calculate the average end-to-end throughput for all random paths. Figure 16 shows how close the schedule produced by

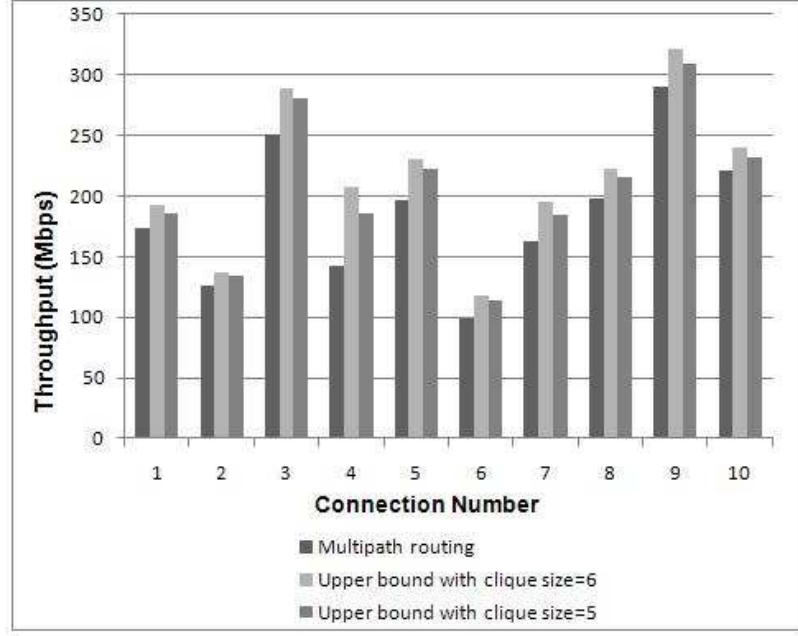


Figure 15. Multipath Routing Performance.

OddEvenSchedule is to the upper bound value. The clique size of the upper bound in this scenario is set to 5, and results from our algorithm are close to the upper bound values.

From the Figure 16 below we can see that compared to shortest routing path and bottleneck routing path, the multipath routing path has the best performance, because for each pair of source and destination nodes, multipath routing may have several paths rather than only one single path, which means that more data can be transmitted.

Statistical analysis of the results

The Figure 17 below is the standard error in the simulation. We calculated the standard for all the ten random paths. In the result figure, black bar is the standard error for throughput achieved by OddEvenSchedule algorithm, light grey bar is standard error for upper bound with clique size of 5 and dark grey bar is the standard error for upper bound with clique size of 6. The number of available channel is set to 15, and in this scenarion each pair of nodes is a separate sample. The routing

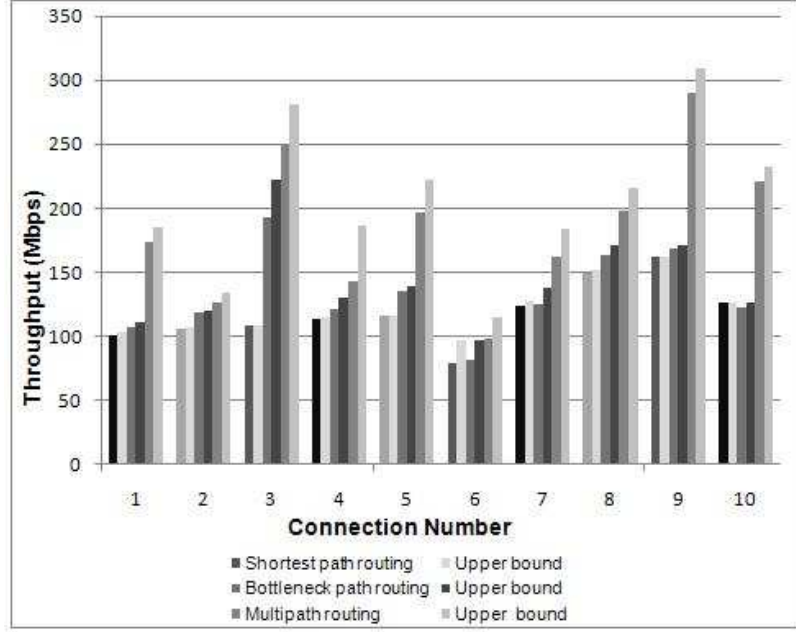


Figure 16. Schedule performance results for three approaches.

path was created by using Dijkstra's algorithm. From the results we can note that the multipath routing has the largest standard error than the other two approaches. If we tighten the upper bound, making the clique size to 7 or 8, the standard error may become larger.

Scenario 3: Varying the number of channels available

In this scenario, the number of channels was varied from 9 to 39, which means that for each simulation run, there are two new channels added into each frequency group. For example, for the group with frequency of 700MHz, the number of available channels will be 3, 5, 7, 9, 11, 13. The same routing paths as given in Scenario 1 were used. In this scenario, we calculate the average end-to-end throughput for all random paths with the same number of available channels and then see how the throughput changes as the number of available channels increases. The average end-to-end transmission rate for all paths is reported. The number of primary users was set to be $1/3$ the number of channels. The results, shown in Figure 18, indicate that

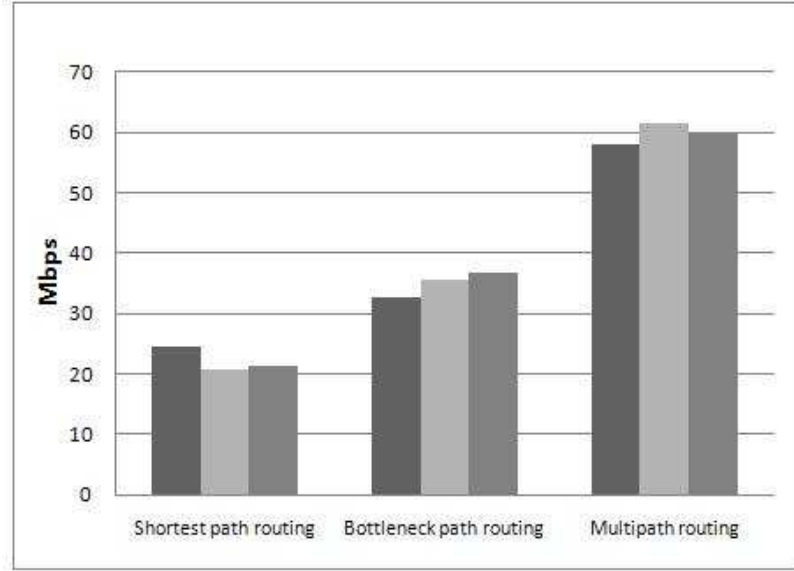


Figure 17. Standard error for three approaches.

an almost linear improvement in terms of additional throughput is gained by adding additional channels to the network.

Scenario 4: Network saturation

In this scenario, we considered how quickly the network would saturate as additional traffic was added to it. we used the original 10 connection pairs from Scenario 1 and also created 10 new random paths in the network. We further assumed that once a connection was established that it would stay active for the remainder of the simulation. So, instead of each connection having a duration of 200 seconds, the transmission is active continuous once it begins. As a result, path 1 will keep transmitting data after 200 seconds, and path 3 will be active at the same time, so in this scenario, we keep adding new paths to the network while the transmission on each path is still on. Because the existing traffic is continuous, the new path will be affected by interference by old traffic more in this scenario. The later the new path is active, the more interference it will experience from all the other old connections. The total number of available channels is set to 15.

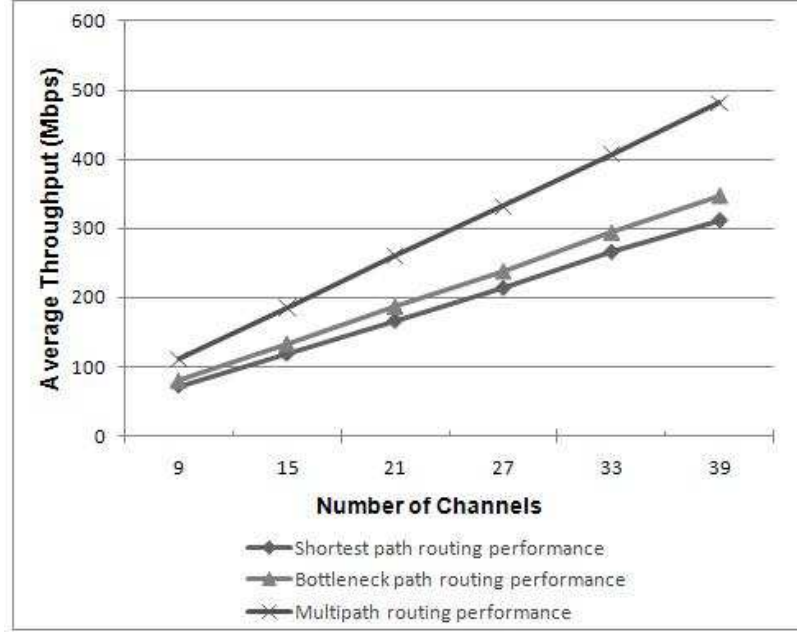


Figure 18. Average path transmission rate versus the number of available channels.

Results are shown in the Figure 19 below. Instead of being interfered with by only one old path, a new path in this scenario will be interfered with by several paths, so the end-to-end throughput of the new path is less than that in scenario 2, which gets interference only from one path. For all of these three approaches, as the number of continuous connection increases, the network becomes saturated, and as a result, the total throughput will no longer increase.

Scenario 5: Throughput versus path length

In this scenario, we considered Path 6 in Figure 12 from (0,0) to (50,50). We considered each subpath P_i consisting of the first i links of Path 6 and used OddEven-Schedule to calculate a transmission schedule for each P_i , assuming no other traffic in the network. The results are shown in Figure 20. We note that the dramatic drop seen in going from P_1 to P_2 is due to the fact that half-duplexing is not required when the path consists of a single link. As the number of links increases, the end-to-end

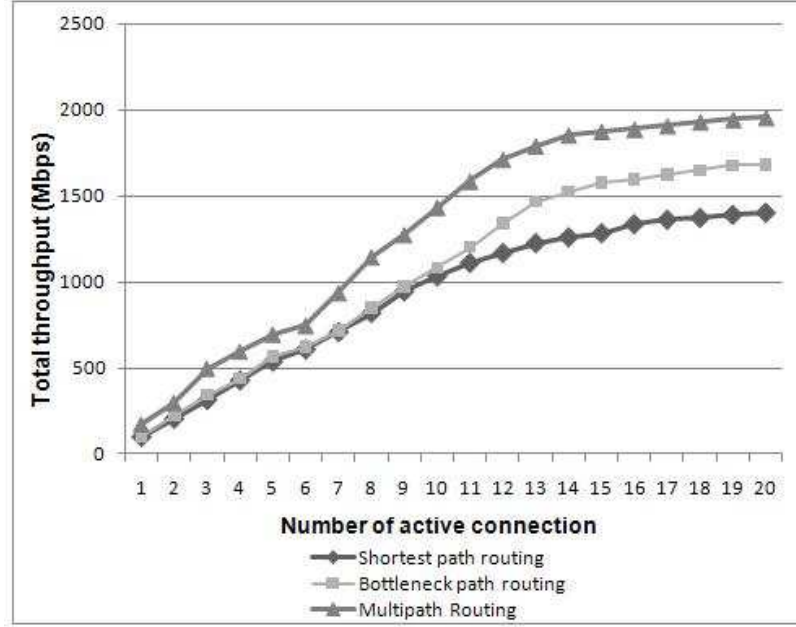


Figure 19. Network saturation as the number of active connections increase.

throughput decreases and there is more intra-flow interference for each link on the path.

Scenario 6: Distribution of channel utilization

In this scenario, we considered the distribution of channel utilization. There are three frequency, and we assume each frequency group has five available channels. For each path in the network we used OddEvenSchedule to calculate a transmission schedule, and each path was generated by using Dijkstra's algorithm. Then found the average times that channels in each frequency group were used during transmission. The results are shown in Figure 21, it gives the distribution of channel utilization as a function of frequency. We note that even though the transmission rate for the same distance is higher at 700 MHz than that of the other two frequency groups, most of traffic happens in the higher frequencies, because of the smaller interference ranges.

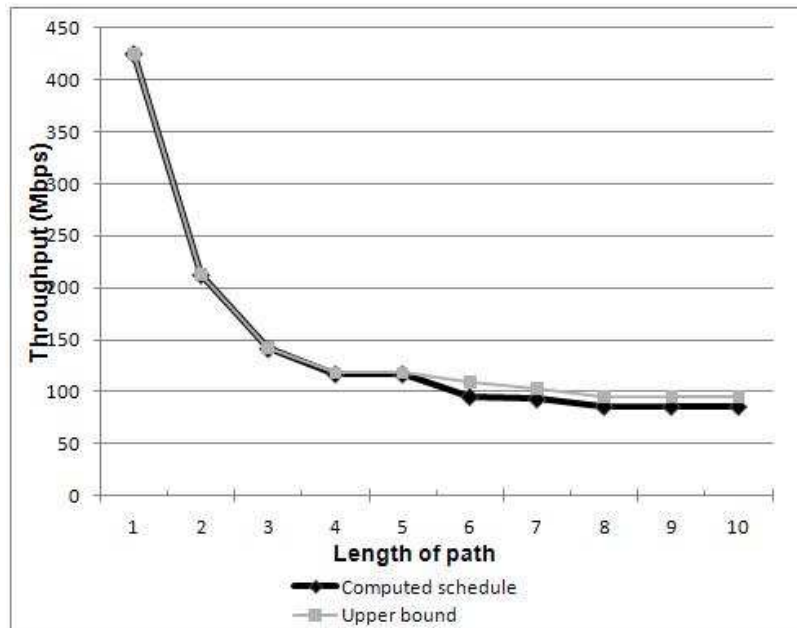


Figure 20. Transmission rate versus path length.

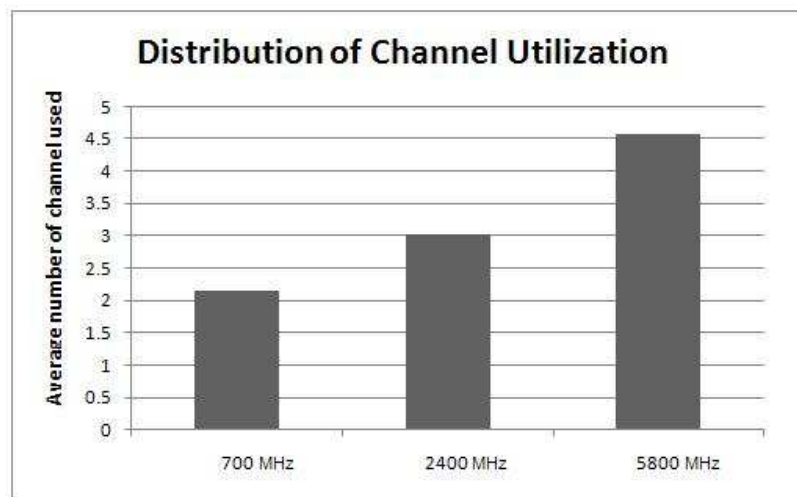


Figure 21. Distribution of channel utilization.

CONCLUSIONS

In this thesis, we propose the first polynomial-time approximation algorithm for the problem of scheduling transmissions along a path or multipath in a cognitive radio mesh network so as to maximize end-to-end throughput. We considered routing via shortest paths, bottleneck paths and odd-even multipaths. As shown in all cases, the schedule provided was very close to an upper bounded computed on optimal solution possible. Bottleneck paths have improved performance over shortest paths and multipath transmissions could achieve almost twice the performance of a shortest path transmission. Finally, there is a gap between the theoretical bound and actual performance in practice; it may be possible to tighten the bound with further analysis, such as we can set the clique size of upper bound to 7 or even higher, but in our programming, it may take too much time to run the simulation.

REFERENCES CITED

- [1] I. F. Akyildiz, X. Wang and W. Wang, Wireless mesh networks: a survey, *Journal of Computer Networks*, Vol. 47, No. 4, 2005, pp. 445–487.
- [2] L. Krishnamurthy, S. Conner, M. Yarvis, J. Chhabra, C. Ellison, C. Brabenac, E. Tsui, Meeting the demands of the digital home with high-speed multi-hop wireless networks, *Intel Technology Journal*, 2002, pp. 57–68.
- [3] S. Tierney, Mesh Networks, whitepaper of communitynetworking. org.
- [4] J. Jun, M.L. Sichitiu, The nominal capacity of wireless mesh networks, *IEEE Wireless Communications*, Vol. 10, No. 5, 2003, pp. 8–14
- [5] R. Poor, Wireless mesh networks, *Sensors*, February 2003.
- [6] R. Poor, Wireless mesh links everyday devices, *Electronic Engineering Times*, 5 July 2004.
- [7] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, *Computer Networks* 38 (4), 2002, pp. 393–422.
- [8] I.F. Akyildiz, I.H. Kasimoglu, Wireless sensor and actor networks: research challenges, *Ad Hoc Networks* 2 , 2004, pp. 351–367.
- [9] J. Walker, Wi-Fi mesh networks, the path to mobile ad hoc. Available from: <http://www.wi-fitechnology.com/Wi-Fi-Reports-and-Papers/Mesh-Networks-References.html>.
- [10] Intel Inc., Multi-Hop Mesh Networks a new kind of Wi- Fi network.
- [11] The ZigBee Alliance. Available from: <http://www.zigbee.org/>.
- [12] I. F. Akyildiz, W-Y Lee, M. C. Vuran and S. Mohanty, NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey, *Journal of Computer Networks*, Vol. 50, No. 13, 2007, pp. 2127-2159.
- [13] <http://www.it.kth.se/~jmitola/Mitola-Dissertation8-Integrated.pdf>.
- [14] S. Haykin, Cognitive radio: brain-empowered wireless communications, *IEEE Journal on Selected Areas in Communications* 23 (2) (2005) pp. 201–220.

- [15] FCC, ET Docket No 03-222 Notice of proposed rule making and order, December 2003.
- [16] R.W. Thomas, L.A. DaSilva, A.B. MacKenzie, Cognitive networks, in: Proc. IEEE DySPAN 2005, November 2005, pp. 352–360.
- [17] F.K. Jondral, Software-defined radio-basic and evolution to cognitive radio, EURASIP Journal on Wireless Communication and Networking 2005.
- [18] D. Cabric, S.M. Mishra, R.W. Brodersen, Implementation issues in spectrum sensing for cognitive radios, in: Proc. 38th Asilomar Conference on Signals, Systems and Computers 2004, November 2004, pp. 772C776.
- [19] D. Cabric, R.W Brodersen, Physical layer design issues unique to cognitive radio systems, in: Proc. IEEE Personal Indoor and Mobile Radio Communications (PIMRC) 2005, September 2005.
- [20] I. F. Akyildiz, W. Lee, M. Vuran and S. Mohanty, A Survey on spectrum management in cognitive radio networks, *IEEE Communications Magazine*, Vol. 46, Issue. 4 pp. 40–48.
- [21] C. Li and C. Li, Opportunistic Spectrum Access in Cognitive Radio Networks, *IEEE International Joint Conference on Neural Networks'2008*, pp. 3412–3415.
- [22] Baker, D.J., Wieselthier, J.E., and Ephremides, A., A Distributed Algorithm for Scheduling the Activation of Links in a Self-Organizing Mobile Radio Network, IEEE Int. Conference Communications, 1982, pp. 2F6.1–2F6.5.
- [23] Hajek, B., and Sasaki, G., *Link Scheduling in Polynomial Time*, IEEE Transactions on Information Theory, 34(5), pp. 910-917, 1988.
- [24] Murali Kodialam, Thyaga Nandagopal, *Characterizing Achievable Rates in Multi-hop Wireless Networks: The Joint Routing and Scheduling Problem*, in International Conference on Mobile Computing and Networking, ACM Mobicom, 2003.
- [25] Qiwei Wang, Haitao Zheng, *Route and Spectrum Selection in Dynamic Spectrum Networks*, Consumer Communications and Networking Conference, CCNC 2006.
- [26] J. Mitola. Cognitive radio: An integrated agent architecture for software-defined radio, 2000.

- [27] R. W. Broderson et al. Corvus: A cognitive radio approach for usage of virtual unlicensed spectrum, Webpage - [http://bwrc.eecs.berkeley.edu/Research/MCMA/CR White paper final1.pdf](http://bwrc.eecs.berkeley.edu/Research/MCMA/CR%20White%20paper%20final1.pdf).
- [28] FCC, Spectrum Policy Task Force Report, ET Docket No. 02-155, November 2002.
- [29] Shashidhar Gandham, Milind Dawande and Ravi Prakash. Link Scheduling in Sensor Networks: Distributed Edge Coloring Revisited. In IEEE INFOCOM, March 2005.
- [30] S. Ramanathan, E. Lloyd. Scheduling Algorithms for Multihop Radio Networks. In IEEE/ACM Transactions on networking, pages 166C177, April 1993.
- [31] E. Arikan. Some complexity results about packet radio networks. In IEEE Transactions Information Theory, pages 910C918, July 1987.
- [32] S. Even, O. Goldreich, S. Moran, and P. Tong. On the npcompleteness of certain network testing problems. In Networks, pages 1C24, 1984.
- [33] <http://en.wikipedia.org/wiki/Graph-coloring>
- [34] Thoppian. M, Venkatesan. S, Prakash. R, Chandrasekaran. R, *MAClayer scheduling in cognitive radio based multihop wireless networks*, World of Wireless, Mobile and Multimedia Networks, 2006.
- [35] McGill University: School of Computer Science, Winter 1997 Class Notes for 308-251B, DATA STRUCTURES AND ALGORITHMS.
- [36] L. Cao and H. Zheng, Distributed spectrum allocation via local bargaining, *Proceedings of IEEE Secon'2005*, pp. 475–486.
- [37] Y. T. Hou, Y. Shi and H. D. Sherali, Optimal spectrum sharing for multi-hop software defined radio networks, *Proceedings of IEEE Infocom'2007*, pp. 1–9.
- [38] J. Huang, R. A. Berry and M. L. Honig, Spectrum sharing with distributed interference compensation, *Proceedings of IEEE DySPAN'2005*, pp. 88–93.
- [39] H. Khalife, S. Ahuja, N. Malouch, and M. Krunz, Probabilistic path selection in opportunistic cognitive radio networks, *Proceedings of IEEE Globecom'2008*, pp. 1–5.
- [40] S. Olariu, An optimal greedy heuristic to color interval graphs, *Inf. Process. Lett.*, vol. 37, no. 1, 1991, pp. 21–25.

- [41] Y. Shi and Y. T. Hou, A distributed optimization algorithm for multi-hop cognitive radio networks, *Proceedings of IEEE Infocom'2008*, pp. 1292–1300.
- [42] J. Tang, S. Misra and G. Xue, Joint spectrum allocation and scheduling for fair spectrum sharing in cognitive radio wireless networks, *Journal of Computer Networks*, Vol. 52, No. 11, 2008, pp. 2148–2158.
- [43] P.-J. Wan, Multiflows in multihop wireless networks, *Proceedings of MobiHoc'2009*, pp. 85–94.
- [44] F. Wang, M. Krunz and S. Cui, Spectrum sharing in cognitive radio networks, *Proceedings of IEEE Infocom'2008*, pp. 1885–1893.
- [45] Y. Xi and E. M. Yeh, Distributed algorithms for spectrum allocation, power control, routing, and congestion control in wireless networks, *Proceedings of ACM MobiHoc'2007*, pp. 180–189.
- [46] C. Xin, B. Xie, C-C. Shen, A novel layered graph model for topology formation and routing in dynamic spectrum access networks, *Proceedings of IEEE DySPAN'2005*, pp. 308–317.
- [47] Y. Yang and R. Kravets, Contention-aware admission control for ad hoc networks, *IEEE Transactions on Mobile Computing*, Vol. 4, No. 4, 2005, pp. 363–377.
- [48] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda and Y. Wu, Allocating dynamic time-spectrum blocks in cognitive radio networks, *Proceedings of ACM MobiHoc'2007*, pp. 130–139.
- [49] Q. Zhao, L. Tong, A. Swami, Decentralized cognitive MAC for dynamic spectrum access, *Proceedings of IEEE DySPAN'2005*, pp. 224–232.
- [50] H. Zheng and C. Peng, Collaboration and fairness in opportunistic spectrum access, *Proceedings of IEEE ICC'2005*, pp. 3132–3136.
- [51] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". *Numerische Mathematik* 1: 269C271. <http://www-m3.ma.tum.de/twiki/pub/MN0506/WebHome/dijkstra.pdf>.
- [52] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "Section 24.3: Dijkstra's algorithm". *Introduction to Algorithms* (Second ed.). MIT Press and McGraw-Hill. pp. 595C601. ISBN 0-262-03293-7.