Design and Implementation of An Emulation Testbed for Optimal Spectrum Sharing in Multi-hop Cognitive Radio Networks

Tong Liu

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science & Applications

Dr. Y. Thomas Hou, Chair

Dr. Anil Vullikanti, Co-Chair

Dr. Madhav Marathe

July 9, 2007

Blacksburg, Virginia

Keywords: Cognitive Radio, Spectrum Sharing, Cross-layer Optimization, Emulation Testbed

Design and Implementation of An Emulation Testbed for Optimal Spectrum Sharing in Multi-hop Cognitive Radio Networks

Tong Liu

ABSTRACT

Cognitive Radio (CR) [1–4] capitalizes advances in signal processing and radio technology and is capable of reconfiguring RF and switching to desired frequency bands. It is a frequency-agile data communication device that is vastly more powerful than existing multi-channel multi-radio (MC-MR) technology [2].

In this thesis, we investigate the important problem of multi-hop networking with CR nodes. In a CR network, each node has a set of frequency bands (not necessarily of equal size) that may not be the same as those at other nodes. The uneven size of frequency bands prompts the need of further division into sub-bands for optimal spectrum sharing. We characterize behaviors and constraints for such multi-hop CR network from multiple layers, including modeling of spectrum sharing and sub-band division, scheduling and interference constraints, and flow routing. We give a formal mathematical formulation with the objective of maximizing the network throughput for a set of user communication sessions. Since such problem formulation falls into mixed integer non-linear programming (MINLP), which is NP-hard in general, we develop a lower bound for the objective by relaxing the integer variables and linearization. Subsequently, we develop a near-optimal algorithm to this MINLP problem. This algorithm is based on a novel *sequential fixing* (SF) procedure, where the integer variables are determined iteratively via a sequence of linear

program (LP).

In order to implement and evaluate these algorithms in a controlled laboratory setting, we design and implement an emulation testbed. The highlights of our experimental research include:

- Emulation of a multi-hop CR network with arbitrary topology;
- An implementation of the proposed SF algorithm at the application layer;
- A source routing implementation that can easily support comparative study between SF algorithm and other schemes;
- Experiments comparing the SF algorithm with another algorithm called Layered Greedy Algorithm (LGA);
- Experimental results show that the proposed SF significantly outperforms LGA.

In summary, the experimental research in this thesis has demonstrated that SF algorithm is a viable algorithm for optimal spectrum sharing in multi-hop CR networks.

Acknowledgments

First, I would like to thank my advisor, Prof. Y. Thomas Hou, for leading me into this exciting research area and giving me the opportunity to work on this exciting project. I am grateful for his guidance, encouragement, and financial support throughout the course of this work. He has made numerous constructive suggestions on this thesis. I also deeply appreciate his time on editing this thesis.

I also want to thank Prof. Anil Vullikanti for serving as co-advisor for my thesis work. Special thanks also go to Prof. Madhav Marathe for serving on my thesis committee.

I am especially grateful for Yi Shi and Xiaojun Wang's sincere help and encouragement throughout the course of this thesis. I would like to thank my fellow students in Prof. Hou's research group, including Sastry Kompella and Jia Liu for their help and encouragement.

Finally, I want to thank my parents and my girlfriend for their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cognitive Radio Networks

1.1.1 Charateristics of Current Spectrum Usage

In recent years, there has been an exponential rise in the number of wireless standards such as 802.11 a,b,g,n (and more), 802.15 (Bluetooth, Zigbee, UWB), WCDMA (and several modes), CDMA 2000 (1xEV-DV, 1xEV-DO), Wireless USB, and 802.16 (various modes) to name a few. The abundance of standards today is the result of a desire in the industry to support new applications, with each standard custom-tailored to a specific operating environment and spectrum. On the other hand, since 1920s, each new wireless service is assigned a fixed block of spectrum for its exclusive use [1,5]. The static spectrum allocation scheme is successful in eliminating interference between different wireless technologies. However, it is difficult to make accurate projections of

future demands. Over the years, the remaining useful spectrum available for new wireless services is being exhausted. It is a serious problem since new wireless services and devices are now rolling out at an unprecedented pace. In particular, building large multi-hop networks has been fundamentally impeded because of the insufficient network capacity due to the limited bandwidth of static allocated channels. This issue will became much more critical as wireless networking becomes pervasive in the future.

Furthermore, the static spectrum management has led to extremely inefficient use of certain allocated spectrum in certain geographical areas. Such frequency bands are called spectrum "white space" (or "hole"). Many measurements and much analysis have been conducted to investigate the severity of sporadically spectrum usage. A recent study done by Shared Spectrum Company [6] found that even in the most crowed area near downtown Washington, DC, where both government and commercial spectrum use is intensive, 62% of the spectrum remains white space. Here, a band is considered white space if it is wider than 1 MHz and remains unoccupied for at least 10 minutes. Another study [7] done by M. McHenry and D. McCloskey indicated the average spectrum utilization for frequencies below 6 GHz is 15% in New York City.

To address problems associated with static allocation, the Federal Communications Commission (FCC) has put forth a landmark Spectrum Policy Task Force (SPTF) report that outlines a variety of possible new initiatives [8], such as trying to make maximum flexibility in the use of spectrum, defining spectrum users' rights and responsibilities, and accounting for all potential dimensions of spectrum usage.

1.1.2 Advances in Cognitive Radio Technology

In May 2003, the FCC followed up on the previously mentioned report [9] and released its proposals in Docket 03-108 exploring Cognitive Radio (CR) technology for spectrum sharing. The idea of using CR to enhance the flexibility of spectrum usage was first introduced by Joseph Mitola III [4]. A CR or Software Defined Radio (SDR) refers to the class of reprogrammable or reconfigurable radio [1]. It capitalizes on advances in signal processing and radio technology, as well as recent advancements in spectrum policy.

A CR is a frequency-agile data communication device with a rich control and monitoring (spectrum sensing) interface. A frequency-agile radio module is capable of reconfiguring RF and switching to newly-selected frequency bands. Thus, a CR can be programmed to tune to and operate on specific frequency bands over a wide range of spectrum [1]. An even more profound advance in CR technology is that *there is no requirement that selected frequencies/channels be contiguous:* the radio can send packets over non-contiguous frequency bands. From an application perspective, CR allows a single radio to provide a wide variety of applications, acting as a cell phone, broadcast receiver, GPS receiver, wireless data terminal or node, etc.

Specifically, a CR is capable of:

- spectrum sensing: determine which portions of the spectrum is available;
- spectrum decision: select the best available band;
- spectrum sharing: coordinate access to a band with other users;

- spectrum mobility: vacate the band when a licensed user is detected;

1.2 Related Work

1.2.1 Multi-Channel Multi-Radio Network

In a Multi-Channel Multi-Radio (MC-MR) network [10, 11, 19, 20], each node is equipped with multiple interfaces. MC-MR nodes are able to communicate simultaneously with neighboring nodes at the same time by using multiple channels. Since each active interface at a node uses a frequency band that is different from that used by another active interface at the same node, there is no interference between them.

It is important to understand that a CR is vastly more powerful and flexible than existing MC-MR technology. First, MC-MR platform employs traditional *hardware-based* radio technology (i.e., signal processing, modulation etc. are all implemented in hardware) and thus each radio can only operate on a single channel at a time and *there is no switching of channel on packet level.* As a result, the number of concurrent channels that can be used at a wireless node is limited by the number of hardware-based radios. In contrast, the radio technology in CR is software-based; a soft-radio is capable of switching frequency bands on packet level. As a result, the number of concurrent frequency bands that can be shared by a single soft-radio is typically much larger than that can be supported by MC-MR. Second, due to the nature of hardware-based radio technology in MC-MR, a common assumption in MC-MR is that there is a set of “common channels” available for every

node in the network; each channel typically has the same bandwidth. However, such assumption is hardly true for CR networks, in which each node may have a different set of frequency bands, each of un-equal size. Due to this difference, CR is required to work on a set of "heterogeneous" channels that are scattered on widely-separated slices of the frequency spectrum with different bandwidths.

It is important to understand that these differences between MC-MR and CR make algorithm design for a CR network much more complex than that under MC-MR. An MC-MR network can be viewed as a special case of a CR network. Therefore, it is possible to tailor the algorithms for CR network for MC-MR network while the converse is not true.

1.2.2 Cognitive Radio Network

Although there has been extensive research on CR at the physical Layer or point-to-point communication, research for CR network in a multi-hop setting remains limited.

In [12], Zhao et al. designed a distributed coordination approach for spectrum sharing. They showed that this approach offers throughput improvement over a dedicated channel approach.

In [13], Ugarte and McDonald studied the network capacity problem for multi-hop CR-based networks and found an upper bound, although it is not clear how tight this bound is.

In [14], Xin et al. studied how to assign frequency bands at each node to form a topology such that a certain performance metric can be optimized. A layered graph was proposed to model frequency bands available at each node and to facilitate topology formation and achieve optimization objec-

tive. The authors considered the so-called fixed channel approach whereby the radio is assumed to operate on only one channel at a specific time.

In [15], Steenstrup studied three different frequency assignment problems: common broadcast frequencies, non-interfering frequencies for simultaneous transmissions, and frequencies for direct source-destination communications. Each is viewed as a graph-coloring problem, and both centralized and distributed algorithms are presented.

In [16], Shi and Hou considered how to perform routing, scheduling, and power control to minimize the network wide resoucre (BFP) for a given set of user communication sessions. A mathematical model on scheduling and power control is provided based on protocol interference model. Subsequently, this problem is formulated and solved via a branch-and-board solution procedure.

1.3 Problem

1.3.1 Objective

Our focus in this thesis is optimal spectrum sharing for a multi-hop CR network. Such network is constructed by CR nodes, each with a set of available spectrum bands. Due to uneven size of these bands, each band may need to be further divided into sub-bands. Consider a set of active user communication sessions in the network. A fundamental question to ask is what is maximum network throughput that can be supported by the network, in terms of a scaling factor of all sessions' rates?

1.3.2 Approach and Solution

To formulate the problem mathematically, we characterize behaviors and constraints from multiple layers for a general multi-hop CR network. Special attention is given to modeling of spectrum sharing and (un-even) sub-band division, scheduling and interference modeling, and multi-path routing. We formulate an optimization problem with the objective of maximizing the network throughput in terms of a common scaling factor for a set of session rates. Since such problem formulation falls into *mixed integer non-linear programming* (MINLP), which is NP-hard in general, we aim to develop near-optimal solutions.

As for preparation, we first develop upper bound for the objective by relaxing the integer variables and linearization. This upper bound can be used as a measure for the quality of any solution. Subsequently, we propose a novel *sequential fixing* (SF) solution procedure where the determination of integer variables is performed iteratively through a sequence of linear programs (LP). Upon fixing of the integer variables, other variables in the optimization problem can be solved with an LP.

1.4 Experimental Research

1.4.1 Objective

In order to implement and evaluate the ideas and algorithms in a controlled laboratory setting, we design and implement an emulation testbed.

The topology of the CR network that we are going to emulate is determined by transmission power

spectral density at each node. In other words, whether or not communication exists between two nodes is determined by their physical distance and the transmitting range of the node. In this thesis, connectivity among the nodes are controlled by a so called Dynamic Switch [25] using Ethernet connections.

1.4.2 Design Highlights

Our emulation testbed is constructed by a cluster of computers (12 Windows-based Dell Optiplex desktops) and a Linux-based Dynamic Switch (a Dell Poweredge Workstation).

Network Topology

The network topology is set by the Dynamic Switch, which connects all 12 desktops via three Intel Pro/1000 Quad Ports Ethernet adapters. At the network level, the dynamic switch is invisible to those desktops.

Graphic User Interface

We provide graphic user interface (GUI) on both source nodes and destination nodes. With the GUI, users can easily change the spectrum setting and select their desired algorithm. The GUI also provides the option for video streaming.

Video Coding

Currently, a number of video coders are available such as MPEG-4 [29], VP6 [30], H.261 [31]. In our testbed, we choose H.263 [27] coder for the video encoding and decoding. H.263 is designed

for a low-bitrate compressed encoding solution. Its source code is easy to be integrated in our program.

Source Routing

Source routing refers to that the source determines the entire path toward destination node. Under source routing, routing information is stored in the packet header. Each intermediate node simply looks up the next hop information in the packet header and forwards the packet based on this information.

1.4.3 Experiments and Results

The objective of our experiments is to validate the performance of SF algorithm in a testbed setting. We also want to compare the SF algorithm with other algorithms. As an example, a widely used approach to cross-layer optimization problem is the decoupled approach, where solution to problem at one layer is first solved (in isolation) before solution for a different layer is devised. In the emulation testbed, we design a Layered Greedy Algorithm (LGA) as a representative of such decouple approach and compare it to SF algorithm.

- In the first experiment, we will run SF algorithm under a given set of spectrum frequency bands available at each node in the network. SF program aims to maximize network throughput in terms of a scaling factor of each flow's bit rate. A video clip will be encoded by such rate and transmitted for the active user sessions. We expect that SF algorithm finds routing paths with excellent network throughput and the video will be encoded and transmitted very

smoothly.

- Under the same spectrum setting as in the first experiment, we will run LGA. The video clip will be encoded by the rate provided by LGA and then transmitted. The bit rate (or scaling factor) achieved by SF and LGA are compared at the receiver side.

- The last experiment shows the flexibility of SF algorithm. The set of spectrum available at each node is changed to a worse condition. Under the new environment (less available spectrum), we compare the bit rates (or scaling factor) achieved by SF and LGA again. We show that SF provides much better result than LGA.

1.5 Thesis Outline

The rest of this thesis will be organized as follows. Chapter 2 presents the theoretical foundation. The spectrum sharing problem is formulated, and the proposed SF algorithm is presented. Simulation results for SF algorithm are also given. Chapter 3 presents the design and implementation of our emulation testbed. Chapter 4 presents the experiments results from the emulation testbed. Chapter 5 summarizes this thesis.

Chapter 2

Optimal Spectrum Sharing: Theory and Algorithm

2.1 Problem Statement

In this thesis, we investigate the important problem of multi-hop networking with CR nodes. For the multi-hop CR network we are investigating, a fundamental performance objective is the overall maximum network throughput in terms of a scaling factor for flow rates assigned to all the active user sessions. For each CR node with a set of available frequency bands, we characterize behaviors and constraints at multiple layers, and formulate a mathematical optimization problem. The formulation includes how to optimally divide the set of available frequency bands at each node, the scheduling of sub-bands for transmission and reception, and multi-path routing for each session so as to maximize the scaling factor.

In this chapter, we will first present the multi-hop CR network model, and then formulate the problem with a number of constraints. Finally, we develop an algorithm based on SF technique. Table 2.1 lists all notations used in this Chapter.

2.2 Network Model

2.2.1 Spectrum Sharing and Sub-band Division

In a multi-hop CR network, each node's view of the available spectrum depends on its location and time. The set of available frequency bands of each node are usually not the same as those of other nodes at a specific time. Further, the size (or bandwidth) of each available frequency band may differ drastically.

The uneven size of bandwidth in the available spectrum band calls for further division of the bands into small sub-bands for more flexible and efficient frequency allocation and utilization. A band can be divided either equally or unequally. An equal sub-band division method is likely to reduce the complexity. On the other hand, an unequal division method may offer more flexibility and therefore produces better performance than equal division. In this project, we allow unequal division of frequency bands.

More formally, we model the total spectrum consisting a set of $\mathcal{M}$ un-equal bands (see Figure 2.1). $\mathcal{M}_i \subseteq \mathcal{M}$ is the set of bands available at node $i \in \mathcal{N}$. The set of available bands at one node is likely to be different from that at another node. For instance, for nodes $i, j \in \mathcal{N}$, $\mathcal{M}_i$ may consist

Table 2.1: Notation.

Symbol	Definition
β	The scaling factor of session rates in the network
$\mathcal{L}$	The set of active user sessions in the network
$r(l)$	Rate of session $l \in \mathcal{L}$
$s(l), d(l)$	Source and destination nodes of session l
$f_{ij}(l)$	Data rate that is attributed to session l on link (i, j)
$\mathcal{M}_i$	The set of available bands at node $i \in \mathcal{N}$
$\mathcal{M}$	The set of available bands in the network, i.e., $\bigcup_{i \in \mathcal{N}} \mathcal{M}_i$
M	$= \lvert\mathcal{M}\rvert$, the number of available bands in set $\mathcal{M}$
$\mathcal{M}_{ij}$	The set of available bands on link (i, j), i.e., $\mathcal{M}_i \bigcap \mathcal{M}_j$
$\mathcal{N}$	The set of nodes in the network
$K^{(m)}$	Total number of sub-bands in band m
$W^{(m)}$	Bandwidth of band $m \in \mathcal{M}$
d_{ij}	Distance between nodes i and j
n	Path loss index
g_{ij}	Propagation gain from node i to node j
Q	Transmission power spectral density at a transmitter
η	Ambient Gaussian noise density
Q_T	The minimum threshold of power spectral density to decode a transmission at a receiver
Q_I	The maximum threshold of power spectral density for interference to be negligible at a receiver
R_T, R_I	Transmission range and interference range, respectively
$\mathcal{T}_i^m$	The set of nodes that can use band m and are within the transmission range to node i
$\mathcal{T}_i$	$= \bigcup_{m \in \mathcal{M}_i} \mathcal{T}_i^m$
$\mathcal{I}_j^m$	The set of nodes that can use band m and are within the interference range to node j
$u^{(m,k)}$	The fraction of bandwidth for the k-th sub-band in band m
$x_{ij}^{(m,k)}$	Binary indicator to mark whether or not sub-band (m, k) is used by link (i, j).

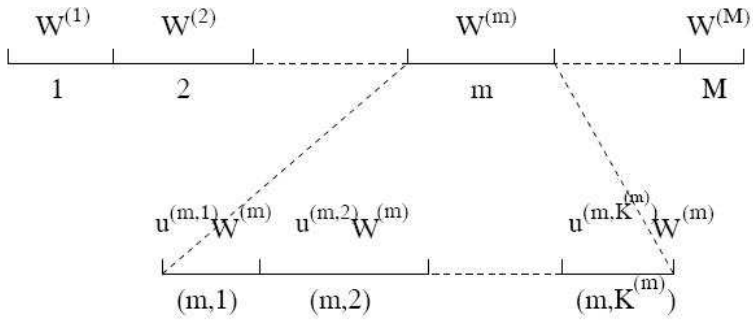


Figure 2.1: A schematic illustrating bands and sub-bands concepts in spectrum sharing

of bands I, II, III, VI, and V while $\mathcal{M}_j$ may consist of bands I, II, VI. In this case, $\mathcal{M}_i \neq \mathcal{M}_j$. $W^{(m)}$ is the bandwidth of band $m \in \mathcal{M}$. As discussed above, it is assumed that a band $m \in \mathcal{M}$ can be further divided into $K^{(m)}$ unequal sub-bands. Denote $u^{(m,k)}$ as the fraction of bandwidth for the k-th sub-band in band m. Then we have the spectrum of $M = |\mathcal{M}|$ bands effectively divided into a total of $\sum_{m=1}^{M} K^{(m)}$ sub-bands, with

$$\sum_{k=1}^{K^{(m)}} u^{(m,k)} = 1 \ .$$

An unequal sub-band division example is illustrated in Figure 2.1.

2.2.2 Transmission Range and Interference Range

A multi-hop CR network topology can be modeled as a graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. A sample topology is shown in Figure 3.3. The network topology is determined by each node's transmission range. The power spectral density from the transmitter of a CR node is Q. In this thesis, we assume that all nodes use the same power spectral density for

transmission. A widely-used model for power propagation gain is

$$g_{ij} = d_{ij}^{-n} \ , \tag{2.1}$$

where n is the path loss index and d_{ij} is the distance between nodes i and j. We assume a data transmission is successful only if the received power spectral density at a receiver is at least Q_T. Likewise, we assume interference will become non-negligible only if it produces a power spectral density over a threshold of Q_I at a receiver. Based on the threshold Q_T, the transmission range for a node is thus $R_T = (Q/Q_T)^{1/n}$, which is calculated from $g_{ij} \cdot Q = Q_T$ and $g_{ij} = d_{ij}^{-n}$. Similarly, based on the threshold Q_I, which is smaller than Q_T, the interference range for a node is $R_I = (Q/Q_I)^{1/n}$. Since $Q_I < Q_T$, we have $R_I > R_T$.

2.2.3 Scheduling

Scheduling addresses how a node uses sub-bands and can be done either in time domain or frequency domain. In this thesis, we consider how to assign sub-band in frequency domain. Our scheduling technique is very similar to the "coloring problem" in graph theory, where two connected areas cannot share the same color. A feasible scheduling of frequency bands insures no interference at the same node and among the nodes. Suppose that band m is available at both node i and node j, i.e., $m \in \mathcal{M}_i \bigcap \mathcal{M}_j$. Denote

$$x_{ij}^{(m,k)} = \begin{cases} 1 & \text{If node } i \text{ transmits data to node } j \text{ on sub-band } (m,k), \\ 0 & \text{otherwise.} \end{cases}$$

In a multi-hop CR network, the "coloring" includes three cases:

- A node i cannot use the same sub-band to transmit to different receiver. Mathematically, this can be modeled as follows.

$$\sum_{q \in \mathcal{T}_i^m} x_{iq}^{(m,k)} \leq 1 \ . \tag{2.2}$$

where $\mathcal{T}_i^m$ is the set of nodes that are within the transmission range of node i, i.e.,

$$\mathcal{T}_i^m = \{j : d_{ij} \leq R_T, j \neq i, m \in \mathcal{M}_j\} \ .$$

- A relay node cannot use the same frequency sub-band for transmission and reception, due to self-interference at the physical layer. Mathematically, this can be modeled as if $x_{ij}^{(m,k)} = 1$, then for any $q \in \mathcal{T}_j^m$, $x_{jq}^{(m,k)}$ must be 0. In other words, we have

$$x_{ij}^{(m,k)} + \sum_{q \in \mathcal{T}_j^m} x_{jq}^{(m,k)} \leq 1 \ . \tag{2.3}$$

Note that in (2.3), we are referring to a specific node j to which node i is transmitting. If $x_{ij}^{(m,k)} = 1$, then $\sum_{q \in \mathcal{T}_j^m} x_{jq}^{(m,k)} = 0$, i.e., node j cannot use the same frequency sub-band (m, k) for transmission. On the other hand, if $x_{ij}^{(m,k)} = 0$, then $\sum_{q \in \mathcal{T}_j^m} x_{jq}^{(m,k)} \leq 1$, i.e., node j may use frequency sub-band (m, k) for transmission, but can only use it for one receiving node $q \in \mathcal{T}_j^m$ (same as in (2.2)).

- In addition to the above constraints at the *same* node, there are also constraints on frequency sub-band use due to potential interference *among the nodes* in the network. In particular, for a frequency sub-band (m, k), if node i uses this sub-band for transmitting data to a node $j \in \mathcal{T}_i^m$, then any other node that can produce interference on node j should not use this sub-band.[1] To model this constraint, we denote $\mathcal{P}_j^m$ the set of nodes that can produce interference

[1]Note that the so-called "hidden terminal" problem is a special case under this constraint.

on node j on band m, i.e.,

$$\mathcal{P}_j^m = \{p : d_{pj} \leq R_I, p \neq j, \mathcal{T}_p^m \neq \emptyset\} \ .$$

The physical meaning of $\mathcal{T}_p^m \neq \emptyset$ in the above definition for $\mathcal{P}_j^m$ is that the interference node p can use band m for a valid transmission to a node in $\mathcal{T}_p^m$. Then we have

$$x_{ij}^{(m,k)} + \sum_{q \in \mathcal{T}_p^m} x_{pq}^{(m,k)} \leq 1 \quad (p \in \mathcal{P}_j^m, p \neq i) \ . \tag{2.4}$$

In (2.4), if $x_{ij}^{(m,k)} = 1$, i.e., node i using frequency sub-band (m, k) to transmit to node j, then any node p that can produce interference on node j should not transmit on this sub-band, i.e., $\sum_{q \in \mathcal{T}_p^m} x_{pq}^{(m,k)} = 0$. On the other hand, if $x_{ij}^{(m,k)} = 0$, (2.4) degenerates into (2.2), i.e., node p may transmit on sub-band (m, k) to one node $q \in \mathcal{T}_p^m$, i.e., $\sum_{q \in \mathcal{T}_p^m} x_{pq}^{(m,k)} \leq 1$. Note that a node cannot receive from multiple nodes on the same frequency band, which is also ensured by (2.4).

It is important to understand that in the interference constraint (2.4), if $x_{ij}^{(m,k)} = 0$, two nodes that can produce interference on node j but are far apart and outside each other's interference range can use the same sub-band (m, k) for transmission. We use an example to illustrate this point. In Fig. 2.2, suppose node 1 is transmitting to node 2 on sub-band (m, k), then any node that can produce interference on node 2 (i.e., node 3 or 5) cannot use the same sub-band for transmission. On the other hand, if node 1 is not using sub-band (m, k) to transmit to node 2, then node 3 may use this sub-band to transmit (to node 4) as stated in (2.4). Likewise, node 5 may also use this sub-band to transmit (to node 6) as stated in (2.4). That is, both nodes 3 and 5 may use the same sub-band for transmission.

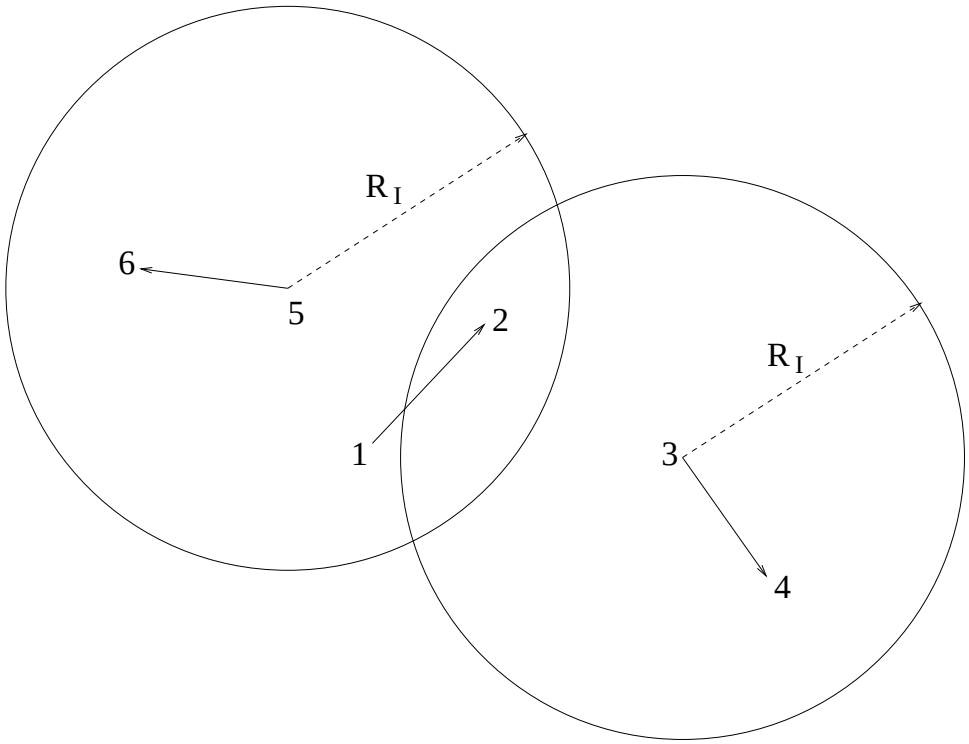


Figure 2.2: An example illustrating interference among links

2.2.4 Routing

A source node may need a number of relay nodes to route the data stream toward its destination node. Note that single path routing may be overly restrictive and may not yield optimal results for spectrum sharing. In this thesis, we consider multi-path, which is much more flexible to route the traffic from a source node to its destination. Thus, the flow routing problem needs to consider the following constraints:

- The flow balance equation at each node $i \in \mathcal{N}$ must be preserved for each session $l \in \mathcal{L}$.
- The link capacity must not be exceeded by the aggregated flow rates on each radio link.

Denote $f_{ij}(l)$ the data rate on link (i, j) that is attributed to session l, where $i \in \mathcal{N}, j \in \bigcup_{m \in \mathcal{M}_i} \mathcal{T}_i^m$, and $l \in \mathcal{L}$. To simplify notation, let $\mathcal{T}_i = \bigcup_{m \in \mathcal{M}_i} \mathcal{T}_i^m$. If node i is the source node of session l, i.e., $i = s(l)$, and β is the scaling factor for $r(l)$, then

$$\sum_{j \in \mathcal{T}_i} f_{ij}(l) - \beta \cdot r(l) = 0 \ . \tag{2.5}$$

If node i is an intermediate relay node for session l, i.e., $i \neq s(l)$ and $i \neq d(l)$, then

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(l)} f_{ij}(l) = \sum_{p \in \mathcal{T}_i}^{p \neq d(l)} f_{pi}(l) \ . \tag{2.6}$$

If node i is the destination node of session l, i.e., $i = d(l)$, then

$$\sum_{p \in \mathcal{T}_i} f_{pi}(l) - \beta \cdot r(l) = 0 \ . \tag{2.7}$$

It can be easily verified that if both (2.5) and (2.6) are satisfied, (2.7) must also be satisfied. As a result, it is sufficient to list (2.5) and (2.6) in the formulation.

In addition to the above flow balance equations at each node i for each session l, the aggregate flow rates on each radio link cannot exceed this link's capacity. To model this mathematically, we need to first find the capacity on link (i, j) in sub-band (m, k). If node i sends data to node j on sub-band (m, k), i.e., $x_{ij}^{(m,k)} = 1$, then the capacity on link (i, j) in sub-band (m, k) is

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

where η is the ambient Gaussian noise density. Note that the denominator inside the log function contains only η. This is due to one of our interference constraints stated earlier, i.e., when node i is transmitting to node j on sub-band (m, k), then all the other neighbors of node j within its interference range are refrained from using this sub-band. This interference constraint significantly

helps to simplify the calculation of link capacity $c_{ij}^{(m,k)}$. When $x_{ij}^{(m,k)} = 0$, we have $c_{ij}^{(m,k)} = 0$. Thus, $c_{ij}^{(m,k)}$ can be written in the following compact form.

$$c_{ij}^{(m,k)} = x_{ij}^{(m,k)} \cdot u^{(m,k)} W^{(m)} \log_2 \left(1 + \frac{g_{ij} Q}{\eta}\right). \tag{2.8}$$

Now getting back to our earlier requirement that the aggregate data rates on each link (i, j) cannot exceed the link's capacity, we have

$$\begin{aligned} \sum_{l \in \mathcal{L}}^{s(l) \neq j, d(l) \neq i} f_{ij}(l) &\leq \sum_{m \in \mathcal{M}_{ij}} \sum_{k=1}^{K^{(m)}} c_{ij}^{(m,k)} \\ = \sum_{m \in \mathcal{M}_{ij}} \sum_{k=1}^{K^{(m)}} x_{ij}^{(m,k)} &\cdot u^{(m,k)} W^{(m)} \log_2 \left(1 + \frac{g_{ij} Q}{\eta}\right). \end{aligned}$$

2.3 Problem Formulation

For the multi-hop CR networks we are investigating, various performance objectives can be used as optimization objective. In this thesis, we use the network throughput as measured by a scaling factor of session rates in the network as our performance objective. In this case, network throughput can be measured in terms of the multiplication of the scaling factor and the initial session rates. It is not hard to see that the solution procedure in this thesis is not constrained by the particular objective. The solution procedure can be extended to tailor other performance objectives if needed.

To re-cap, we are given a set of source-destination pairs (user communication sessions) in the network each with a certain assigned rate. Each node in the network has a set of available frequency bands that it can use for communication. We want to find the optimal solution to divide the set of available frequency bands in the network, the scheduling of these sub-bands for transmission

and reception at each node, and multi-hop routing for each flow such that network throughput (as measured by a scaling factor of session rates) in the network is maximized. Mathematically, we have the following optimization problem.

$$\textbf{Max} \quad \beta$$

$$\textbf{s.t.} \quad \sum_{k=1}^{K^{(m)}} u^{(m,k)} = 1 \qquad (m \in \mathcal{M})$$

$$\sum_{q \in \mathcal{T}_i^m} x_{iq}^{(m,k)} \leq 1 \qquad (i \in \mathcal{N}, m \in \mathcal{M}_i, 1 \leq k \leq K^{(m)}) \tag{2.9}$$

$$x_{ij}^{(m,k)} + \sum_{q \in \mathcal{T}_p^m} x_{pq}^{(m,k)} \leq 1 \qquad (i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{T}_i^m, 1 \leq k \leq K^{(m)}, p \in \mathcal{I}_j^m, p \neq i) \tag{2.10}$$

$$\sum_{l \in \mathcal{L}}^{s(l) \neq j, d(l) \neq i} f_{ij}(l) - \sum_{m \in \mathcal{M}_{ij}} \sum_{k=1}^{K^{(m)}} W^{(m)} \log_2\left(1 + \frac{g_{ij} Q}{\eta}\right) x_{ij}^{(m,k)} u^{(m,k)} \leq 0 \quad (i \in \mathcal{N}, j \in \mathcal{T}_i)$$

$$\sum_{j \in \mathcal{T}_i} f_{ij}(l) - \beta \cdot r(l) = 0 \qquad (l \in \mathcal{L}, i = s(l))$$

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(l)} f_{ij}(l) - \sum_{p \in \mathcal{T}_i}^{p \neq d(l)} f_{pi}(l) = 0 \qquad (l \in \mathcal{L}, i \in \mathcal{N}, i \neq s(l), d(l))$$

$$x_{ij}^{(m,k)} = 0 \text{ or } 1, u^{(m,k)} \geq 0 \qquad (i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{T}_i^m, 1 \leq k \leq K^{(m)})$$

$$\beta, f_{ij}(l) \geq 0 \qquad (l \in \mathcal{L}, i \in \mathcal{N}, i \neq d(l), j \in \mathcal{T}_i, j \neq s(l)) \ ,$$

where $W^{(m)}, g_{ij}, Q$, η, and $r(l)$ are all constants and β, $x_{ij}^{(m,k)}$'s, $u^{(m,k)}$'s, and $f_{ij}(l)$'s are all optimization variables. In (2.10), $\mathcal{I}_j^m$ is defined as

$$\mathcal{I}_j^m = \{p : d_{pj} \leq R_I, \mathcal{T}_p^m \neq \emptyset\},$$

which is equivalent to

$$\mathcal{I}_j^m = \begin{cases} \mathcal{P}_j^m \bigcup \{j\} & \text{If } \mathcal{T}_j^m \neq \emptyset \ , \\ \mathcal{P}_j^m & \text{otherwise} \ . \end{cases}$$

This compact form enables (2.10) to include both (2.3) and (2.4) in the formulation.

The above optimization problem is in the form of mixed-integer non-linear programming (MINLP) problem. MINLP models are NP-hard in general [17]. Although an MINLP problem on a small-sized network, e.g., several nodes, can be solved by existing software, e.g., BARON [21], it is not feasible to solve such problem in a larger network with software due to its time complexity.

Our approach to this problem is as follows. We first relax the integer variables in the previous formulation so as to make it a standard LP problem . Using the result from the LP problem as an upper bound, we design an SF algorithm, which we show achieves near-optimal result.

2.4 Solution

2.4.1 Relaxed Formulation

The integer variables $x_{ij}^{(m,k)}$ and product of variables $x_{ij}^{(m,k)}u^{(m,k)}$ in the formula make the problem nonpolynominal. In the relaxed formulation, we can relax the integer requirement on $x_{ij}^{(m,k)}$ with $0 \leq x_{ij}^{(m,k)} \leq 1$ and replace $x_{ij}^{(m,k)}u^{(m,k)}$ with a single variable, say $s_{ij}^{(m,k)}$, i.e., $s_{ij}^{(m,k)} = x_{ij}^{(m,k)}u^{(m,k)} \leq u^{(m,k)}$. Then new problem formulation becomes:

$$
\begin{array}{llll}
\textbf{Max} & \beta & & \\
\textbf{s.t.} & \sum_{k=1}^{K^{(m)}} u^{(m,k)} = 1 & (m \in \mathcal{M}) & \\
& \sum_{q\in\mathcal{T}_i^m} s_{iq}^{(m,k)} - u^{(m,k)} \leq 0 & (i\in\mathcal{N}, m\in\mathcal{M}_i, 1\leq k\leq K^{(m)}) & (2.11) \\
& s_{ij}^{(m,k)} + \sum_{q\in\mathcal{T}_p^m} s_{pq}^{(m,k)} - u^{(m,k)} \leq 0 & (i\in\mathcal{N}, m\in\mathcal{M}_i, j\in\mathcal{T}_i^m, & \\
& & 1\leq k\leq K^{(m)}, p\in\mathcal{I}_j^m, p\neq i) & (2.12) \\
& \sum_{l\in\mathcal{L}}^{s(l)\neq j, d(l)\neq i} f_{ij}(l) - \sum_{m\in\mathcal{M}_{ij}} \sum_{k=1}^{K^{(m)}} W^{(m)} \log_2\left(1+\frac{g_{ij}Q}{\eta}\right) s_{ij}^{(m,k)} \leq 0 & (i\in\mathcal{N}, j\in\mathcal{T}_i) & \\
& \sum_{j\in\mathcal{T}_i} f_{ij}(l) - \beta\cdot r(l) = 0 & (l\in\mathcal{L}, i=s(l)) & \\
& \sum_{j\in\mathcal{T}_i}^{j\neq s(l)} f_{ij}(l) - \sum_{p\in\mathcal{T}_i}^{p\neq d(l)} f_{pi}(l) = 0 & (l\in\mathcal{L}, i\in\mathcal{N}, i\neq s(l), d(l)) & \\
& u^{(m,k)}, s_{ij}^{(m,k)} \geq 0 & (i\in\mathcal{N}, m\in\mathcal{M}_i, j\in\mathcal{T}_i^m, 1\leq k\leq K^{(m)}) & \\
& f_{ij}(l) \geq 0 & (l\in\mathcal{L}, i\in\mathcal{N}, i\neq d(l), j\in\mathcal{T}_i, j\neq s(l)) &
\end{array}
$$

This new formulation falls into a standard LP problem, which can be solved in polynomial time. The solution to this LP problem can be used as an upper bound to the objective function of the original MINLP problem.

2.4.2 An Algorithm Based on Sequential Fixing

Before we start describing the algorithm, we now go back and review some helpful properties of the original MINLP problem formulation in section 2.3. Notice that it will be reduced into an LP problem if the integer variable has been determined either 0 or 1. Once the binary variables are

chosen, i.e., whether or not a sub-band in certain node will be occupied for transmission, the rest of computation can be solved with an LP problem, which can be done in polynomial time.

We propose an algorithm based on an SF manner. It is a two-step solution procedure:

- Fix the binary values for $x_{ij}^{(m,k)}$ iteratively in polynomial time.
- Find the final solution once x variables are all fixed.

It is important to decide how to fix the binary values in this two-step solution. With the relaxed formulation in Section 2.4.1, we can get a value within the [0,1] region instead of the binary values for all $x_{ij}^{(m,k)}$. Our main idea is to set the relaxed value, i.e., $0 \leq x_{ij}^{(m,k)} \leq 1$ to a binary value, i.e., $x_{ij}^{(m,k)} = 1$ or $x_{ij}^{(m,k)} = 0$. During each iteration, at least one binary value for some $x_{ij}^{(m,k)}$ will be set to 1. Specifically, in the first iteration, we relax all binary variables $x_{ij}^{(m,k)}$ to $0 < x_{ij}^{(m,k)} < 1$ by the linear relaxation. A solution with all $x_{ij}^{(m,k)} = s_{ij}^{(m,k)}/u^{(m,k)}$ being a fraction between 0 and 1 is solved after the first iteration. Among all these x fractions, we select $x_{ij}^{(m,k)}$ with the largest value, and then we fix it to 1. Subsequently, we can set some other x variables to 0 based on the constraints brought in section 2.2.4, i.e.,

- fix $x_{iq}^{(m,k)}$ to 0 for $q \in \mathcal{T}_i^m$ and $q \neq j$ by the coloring constraint in (2.9), and
- fix $x_{pq}^{(m,k)}$ to 0 for $p \in \mathcal{I}_j^m, p \neq i$, and $q \in \mathcal{T}_p^m$ by the generalized coloring constraint in (2.10).

After the first iteration, some x variables will be fixed to 1 or 0, while the remaining x variables have not been fixed. We update the LP problem for the second iteration as follows. For those $x_{ij}^{(m,k)}$

Sequential Fixing (SF) Algorithm

1. Set up and solve the initial relaxed LP problem as shown in Section 2.4.1.
2. Suppose $x_{ij}^{(m,k)}$ is the largest among all the x values that remain to be fixed, fix this $x_{ij}^{(m,k)} = 1$.

   Also fix $x_{iq}^{(m,k)} = 0$ (for $q \in \mathcal{T}_i^m$ and $q \neq j$) and $x_{pq}^{(m,k)} = 0$ (for $p \in \mathcal{I}_j^m, p \neq i$, and $q \in \mathcal{T}_p^m$).
3. If all x values are fixed, go to Step 5.
4. Otherwise, reformulate and solve a new relaxed LP problem with newly fixed x variables and go to Step 2.
5. Formulate an LP problem based on all fixed x values. Obtain a solution to this LP problem.

Figure 2.3: Sequential Fixing (SF) algorithm.

that are already fixed to 1, since $s_{ij}^{(m,k)} = x_{ij}^{(m,k)} u^{(m,k)} = u^{(m,k)}$, we can replace the corresponding $x_{ij}^{(m,k)}$ by $u^{(m,k)}$'s. For those $x_{iq}^{(m,k)}$'s and $x_{pq}^{(m,k)}$ that are fixed to 0, we can set $s_{iq}^{(m,k)} = 0$ and $s_{pq}^{(m,k)} = 0$. Now all the terms in the LP involving these s variables can be removed and the corresponding constraint in (2.11) and (2.12) can also be removed.

In the second iteration, we solve the updated LP problem and fix more x variables to either 1 or 0 based on the same technique in the first iteration. This process can be done iteratively as long as there are any remaining x variables to be fixed. Eventually the iteration will terminate and we have all x variables fixed to either 0 or 1.

Once all x variables are fixed, we can solved all the other variables with a single LP. The detailed algorithm is shown in Figure 2.3.

2.5 Simulation Results

2.5.1 Simulation Settings

In this section, we present simulation results for our SF algorithm and compare it to the upper bound obtained in Section 2.4.1. The units for distance, rate, and power density are shown in Table 2.2. We consider $|\mathcal{N}| = 12$ nodes in the network. Among these nodes, there are $|\mathcal{L}| = 2$ active sessions, where each session has an initial flow bit rate of 20 Kb/s. We assume the transmission range of each node is 50 m and the interference range is 75 m. The path loss index n is assumed to be 4. The white Gaussian noise η is set to 10^{-9} W/KHz. The transmission power spectral density at a node is 10^{-3} W/KHz.

Table 2.2: Units involved in the simulation study.

Item	Unit	Range
Distance	m	$10^1 - 10^2$
Bandwidth	KHz	$10^2 - 10^3$
White Gaussian noise	W/KHz	10^{-9}
Flow	Kb/s	$10^2 - 10^3$

Table 2.3 shows the frequency bands available in the network in the simulation.

Table 2.3: Available bands among all nodes in the network in the simulation study.

Band Index	Bandwidth (KHz)	Allowed Number of Sub-bands
I	130.0	2
II	300.0	3
III	417.5	4
IV	625.0	4
V	925.0	5

2.5.2 Results for Two Specific Instances

In this section, we show two specific instances, which are called “Scenario A” and “Scenario B”. Table 2.4 shows the X and Y position of all nodes in the network in these two scenarios. Table 2.5 shows the available bands at each node under each scenario. In general, there are more spectrum bands available to nodes in the network in Scenario A than those in Scenario B.

In both scenarios, we consider two active communication sessions. The first session is from node 1 to node 12 and the second session is from node 2 to node 10. We assign the initial session rate as 20 Kb/s for both sessions. Table 2.6 and Table 2.7 show the SF solution for these two scenarios. In Scenario A, the network throughput calculated by SF algorithm is 281.54 Kb/s, the upper bound obtained by linear relaxation is 281.54 Kb/s and the scaling factor is 14.077. In Scenario B, the network throughput is 155.20 Kb/s, the upper bound is 155.20 Kb/s and the scaling factor is 7.76.

Table 2.4: X and Y position of all nodes in Scenario A and B.

Node Index	X position (m)	Y position (m)
1	0	55.5
2	15.73	9.25
3	32.04	37
4	15.73	92.5
5	50.54	6.85
6	50.54	106
7	32.04	74
8	69.04	37
9	85.29	18.5
10	85.29	101.8
11	69.04	74
12	101.1	55.5

Table 2.5: Available bands setting.

Node Index	Available Bands in Scenario A	Node Index	Available Bands in Scenario B
1	3, 5	1	3
2	2, 4, 5	2	1, 4
3	1, 3, 4 ,5	3	2, 1
4	1, 4 ,5	4	4, 5 ,1
5	2	5	2
6	1	6	1
7	3, 4	7	4, 3
8	3, 4	8	2, 4, 5
9	2	9	2
10	2, 3, 4	10	2
11	1, 2, 3, 5	11	2
12	2	12	2

Table 2.6: Solution for Scenario A.

Link	Band	Sub-band	Bandwidth(KHz)	Flow $f_{nodei,nodej}(session)$	Rate(Kb/s)
$1 \rightarrow 2$	3	1	138.43	$f_{1,2}(1)$	39.63
$1 \rightarrow 3$	3	3	25.37	$f_{1,3}(1)$	137.59
$1 \rightarrow 3$	5	2	161.22	$f_{1,7}(1)$	104.32
$1 \rightarrow 7$	5	5	141.48	$f_{2,3}(1)$	39.63
$2 \rightarrow 3$	2	2	288.30	$f_{3,8}(1)$	177.22
$3 \rightarrow 7$	3	1	242.79	$f_{7,11}(1)$	104.32
$3 \rightarrow 8$	5	3	379.50	$f_{8,12}(1)$	177.32
$7 \rightarrow 11$	4	1	384.30	$f_{11,12}(1)$	104.32
$8 \rightarrow 11$	3	4	139.10	$f_{2,3}(2)$	281.54
$8 \rightarrow 12$	4	2	240.69	$f_{3,7}(2)$	178.99
$11 \rightarrow 10$	3	1	138.43	$f_{3,8}(2)$	102.55
$11 \rightarrow 10$	3	2	114.57	$f_{7,11}(2)$	178.99
$11 \rightarrow 12$	1	2	130.00	$f_{8,11}(2)$	102.55
$11 \rightarrow 12$	2	1	11.69	$f_{11,10}(2)$	281.54

Table 2.7: Solution for Scenario B.

Link	Band	Sub-band	Bandwidth(KHz)	Flow $f_{nodei,nodej}(session)$	Rate(Kb/s)
$1 \rightarrow 3$	3	1	210.47	$f_{1,3}(1)$	155.20
$1 \rightarrow 7$	3	4	130.40	$f_{3,8}(1)$	155.20
$2 \rightarrow 1$	4	4	366.11	$f_{8,12}(1)$	155.20
$2 \rightarrow 3$	4	3	52.49	$f_{2,3}(2)$	58.98
$3 \rightarrow 7$	1	2	80.00	$f_{3,7}(2)$	58.98
$3 \rightarrow 8$	2	1	160.52	$f_{1,7}(2)$	96.22
$3 \rightarrow 8$	1	1	49.99	$f_{2,1}(2)$	96.22
$7 \rightarrow 11$	4	2	210.51	$f_{7,11}(2)$	155.20
$8 \rightarrow 12$	4	1	25.42	$f_{11,10}(2)$	155.20
$8 \rightarrow 12$	5	1	185.36		
$11 \rightarrow 10$	2	2	139.47		

2.5.3 Complete Simulation Results

In Table 2.8, we show simulation results for 100 data sets for 12-node networks that can produce feasible solutions. For each data set, the network topology, source/destination pair and bit rate of each session, and available frequency bands at each node are randomly generated. As discussed, we compare the result obtained by SF algorithm with the upper bound developed in Section 2.4.1.

Figure 2.4 shows the ratio for the result obtained by the SF algorithm with respect to the upper bound for 100 data sets. The average ratio for the 100 simulations is 0.92. The standard deviation is 0.08. We find that the ratio of the solution obtained by SF to the upper bound solution is close to 1 (in many cases, they coincide with each other). Based on these results, we can make two observations.

- First, since the optimal solution (unknown) is between the solution obtained by the SF algorithm and the upper bound, the upper bound is very tight;
- Second, the SF solution must be even closer to the optimum.

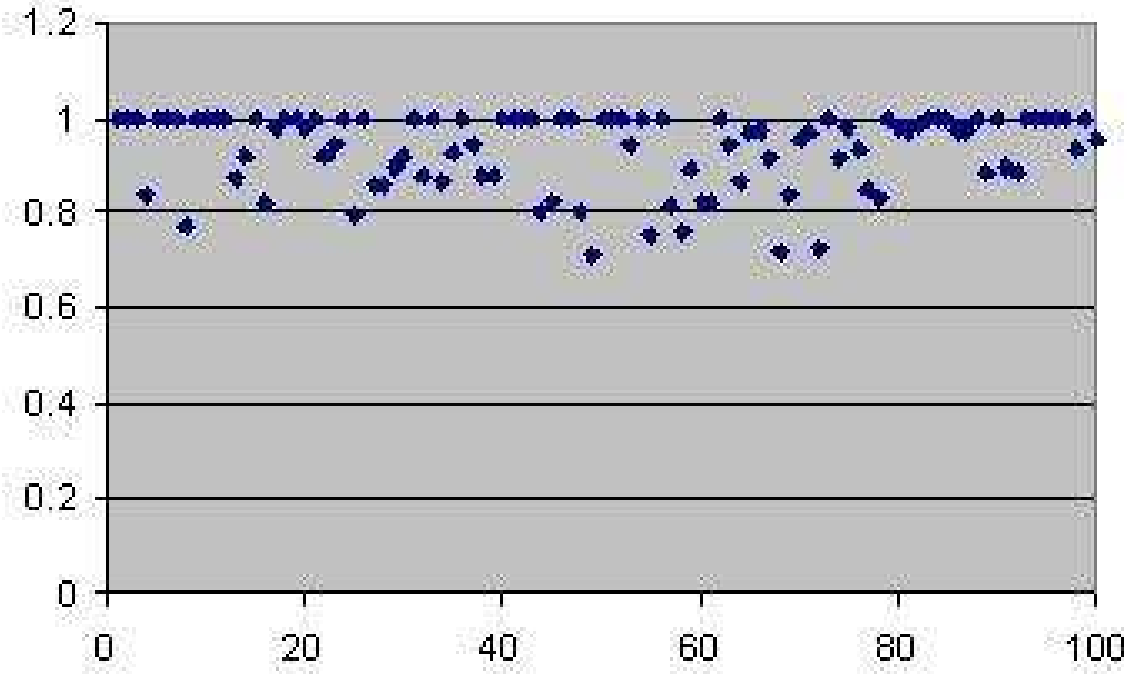


Figure 2.4: Ratio of results obtained by SF algorithm and results obtained by upper bound solution for 100 simulation runs.

Table 2.8: Simulation results for the 100 data sets of 12-node networks.

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
1	281.54	281.54	14.077
2	155.2	155.2	7.76
3	359.88	359.88	17.994

Continued on Next Page. . .

Table 2.8 – Continued

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
4	270.5	322.75	13.525
5	203.85	203.85	10.1925
6	133.56	133.56	6.678
7	245.02	245.15	12.251
8	204.86	264.96	10.243
9	271.17	271.17	13.5585
10	185.8	185.8	9.29
11	150.58	150.58	7.529
12	214.26	214.26	10.713
13	291.09	333.37	14.5545
14	235.83	255.83	11.7915
15	151.61	151.61	7.5805
16	144.69	176.78	7.2345
17	260.12	265.55	13.006
18	263.98	263.98	13.199
19	357.66	357.66	17.883

Continued on Next Page. . .

Table 2.8 – Continued

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
20	204.73	209.21	10.2365
21	134.61	134.61	6.7305
22	225.58	245.58	11.279
23	232.43	246.43	11.6215
24	284.26	284.26	14.213
25	196.69	247.68	9.8345
26	127.26	127.26	6.363
27	236.97	275.97	11.8485
28	214.37	248.37	10.7185
29	265.11	295.07	13.2555
30	198.64	216.13	9.932
31	109.21	109.21	5.4605
32	232.86	264.56	11.643
33	244.18	244.18	12.209
34	215.58	248.68	10.779
35	133.52	144.15	6.676

Continued on Next Page. . .

Table 2.8 – Continued

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
36	135.66	135.66	6.783
37	295.09	312.78	14.7545
38	226.83	257.83	11.3415
39	215.97	245.97	10.7985
40	187.58	187.58	9.379
41	44.24	44.24	2.212
42	267.98	267.98	13.399
43	246.05	246.05	12.3025
44	212.76	264.86	10.638
45	82.85	100.01	4.1425
46	102.1	102.1	5.105
47	244.43	245.43	12.2215
48	293.62	366.15	14.681
49	295.09	412.78	14.7545
50	202.96	202.96	10.148
51	51.83	51.83	2.5915

Continued on Next Page. . .

Table 2.8 – Continued

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
52	229.37	229.37	11.4685
53	294.86	312.22	14.743
54	243.98	243.98	12.199
55	184.91	244.51	9.2455
56	43.35	43.35	2.1675
57	266.29	326.13	13.3145
58	272.88	359.03	13.644
59	217.43	243.83	10.8715
60	82.81	99.95	4.1405
61	143.65	175.12	7.1825
62	255.53	255.53	12.7765
63	273.23	289.01	13.6615
64	271.5	312.75	13.575
65	201.35	207.85	10.0675
66	129.76	133.56	6.488
67	272.3	297.66	13.615

Continued on Next Page. . .

Table 2.8 – Continued

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
68	293.93	409.92	14.6965
69	270.17	322.13	13.5085
70	235.8	246.39	11.79
71	180.58	186.19	9.029
72	293.62	405.15	14.681
73	265.91	265.91	13.2955
74	284.25	310.02	14.2125
75	177.61	181.24	8.8805
76	155.69	166.67	7.7845
77	233.97	275.97	11.6985
78	219.98	263.98	10.999
79	241.05	241.05	12.0525
80	204.73	209.21	10.2365
81	131.31	134.61	6.5655
82	242.58	245.58	12.129
83	244.43	244.43	12.2215

Continued on Next Page. . .

Table 2.8 – Continued

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
84	260.26	260.26	13.013
85	186.69	186.69	9.3345
86	124.26	127.26	6.213
87	243.97	249.97	12.1985
88	244.37	244.37	12.2185
89	235.97	265.97	11.7985
90	168.64	168.64	8.432
91	129.21	144.11	6.4605
92	264.86	299.86	13.243
93	233.18	233.18	11.659
94	235.58	235.58	11.779
95	123.37	123.37	6.1685
96	125.66	125.66	6.283
97	265.91	265.91	13.2955
98	280.25	300.02	14.0125
99	245.97	245.97	12.2985

Continued on Next Page. . .

Table 2.8 – Continued

Index	**Result by SF (Kb/s)**	**Upper Bound (Kb/s)**	**Scaling Factor**
100	177.28	184.85	8.864

Chapter 3

Emulation Testbed Design and Implementation

3.1 Objective

The objective of the emulation testbed is to validate the SF algorithm proposed in Section 2. By "emulation" testbed, we mean a testbed setup in a lab environment where real protocols and algorithms are implemented on each node. Although the testbed is maintained in an indoor lab environment, we are still able to test how an actual wireless network would behave by configuring the network connectivity topology. In fact, an emulation testbed combines the best benefits of both real world experimental trial and purely software-based network simulations. Like real world implementation, real protocol are used at each node in the testbed and experiments. Unlike real world implementation, the actual experiments can be conducted in the convenience and comfort of

an indoor lab environment rather than in an outdoor open field. Furthermore, an emulation testbed can easily create the desired network operating environment with reproducible experiment results.

In this chapter, we first introduce the hardware setup and software architectures, we will also describe key software components within the software architecture.

3.2 Testbed Hardware Setup

Our emulation testbed consists of 12 Dell Optiplex desktops and one Dell Precision Workstation, as shown in Figure 3.1. The Dell Precision Workstation serves as Dynamic Switch [25]. It is

Figure 3.1: Testbed hardware

equipped with three Intel Pro/1000 GT Quad ports Ethernet adapters that interconnect 12 Dell Optiplex desktops (see Figure3.2).

Figure 3.2: Dynamic Switch

Among the 12 Dell Optiplex desktops, two of them are used as the source nodes and another two are used as the corresponding destination nodes.

Dynamic Switch was designed by T. Lin et. al [25]. It serves as a physical layer switch that interconnects multiple hosts based on a given network topology. The operating system on a Dynamic Switch machine is Linux Redhat 9 with a modified kernel. Dynamic Switch is capable of emulating a multi-hop wireless network without any changes on the end nodes. All the end nodes are connected through Ethernet cable to the Dynamic Switch. The Dynamic Switch forwards packets to the proper outgoing network interfaces based on the incoming network interface. Using this switch, researchers can dynamically construct different network topologies by simply uploading different topology files. Dynamic Switch machine is transparent to all the connected end nodes.

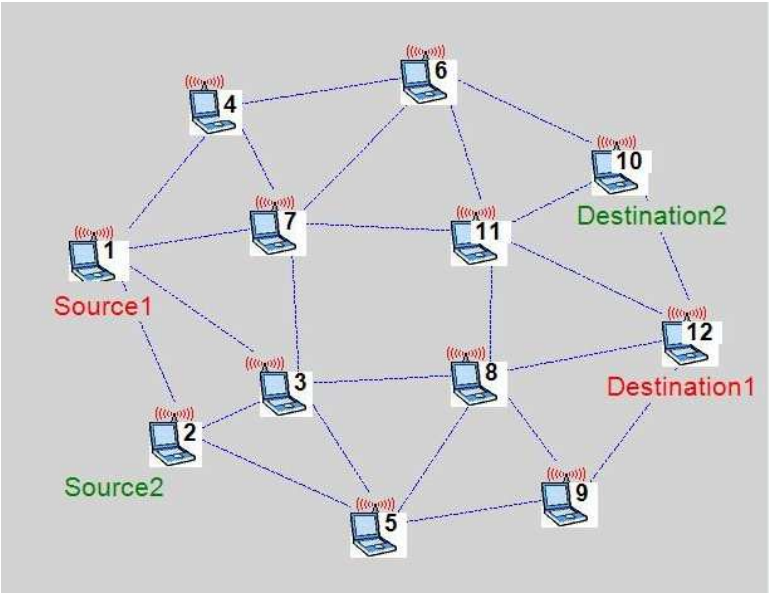


Figure 3.3: An experimental topology

Such transparency makes it possible to implement the desired protocols on the nodes based on experimental needs. A sample topology set up by Dynamic Switch is shown in Figure 3.3.

3.3 Software Architecture

Figure 3.4 shows the protocol layer and software architecture of our testbed. The software architecture contains three main components: source node, wireless router and destination node. The communication within these three components is based on UDP/IP over an Ethernet environment. The connectivity of the Ethernet ports of these end nodes are controlled by a Dynamic Switch.

The end-to-end communication process consists of the following componnets:

1. The source nodes takes an input file that specifies available spectrum bands at each node.

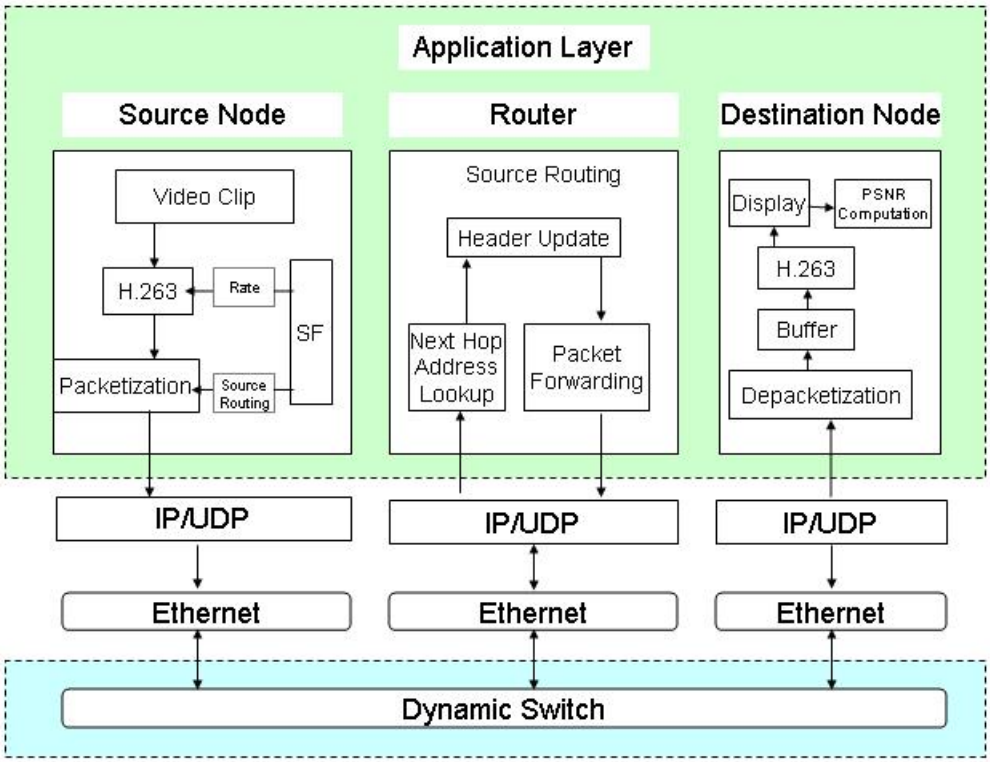


Figure 3.4: The software architecture of our testbed.

2. The algorithm at the source node computes the solution for spectrum allocation, scheduling, routing and flow rate.

3. H.263 encoder encodes the video clip based on the calculated rate.

4. Video frames are packetized and source routing is used for routing.

5. Packets goes through UDP/IP and Ethernet stack and are sent to the Dynamic Switch.

6. The Dynamic Switch performs routing based on information in the packet header.

7. The destination node receives packets and saves them in buffer.

8. Packets are read out from the buffer, decoded and displayed.

9. PSNR curve for the video stream is computed and displayed.

In the rest of this section, we describe these components at source, intermediate node, and destination, respectively.

3.3.1 Source Node

A source node needs to know the spectrum information at each node for the entire network. A source node needs to update the available spectrum table if available spectrum at a node changes in the network. With the available spectrum information as an input file, a user starts by first choosing the desired algorithm (i.e, SF algorithm or LGA). Either algorithm generates text file as its output: containing bit rate for video encoding and routing paths for the video. H.263 encoder take the rate

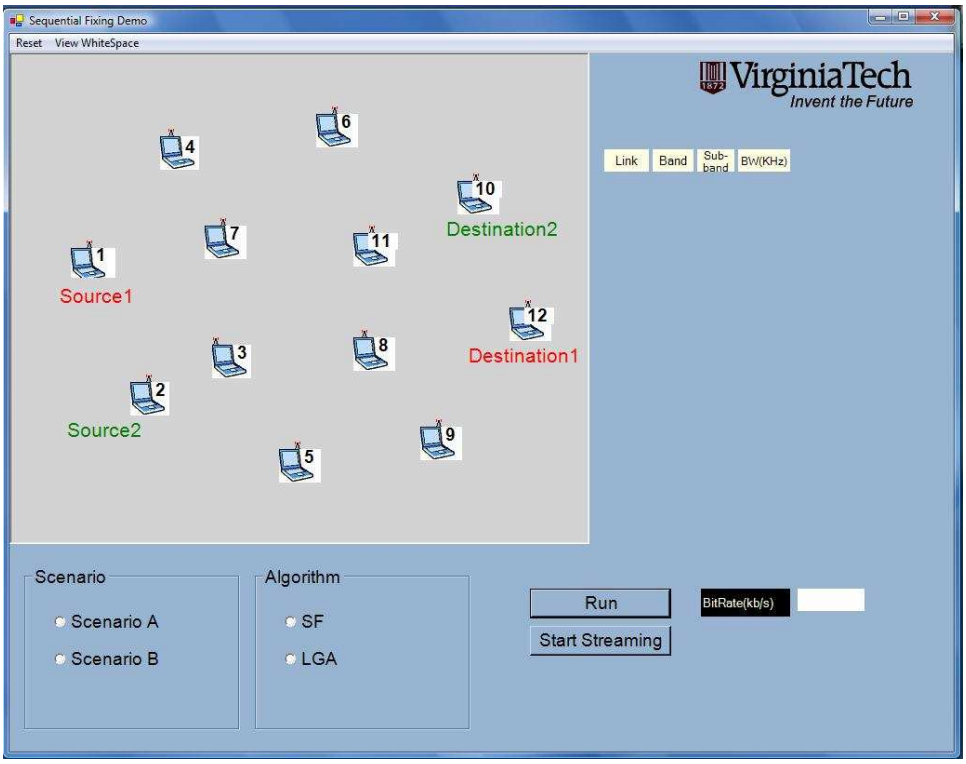


Figure 3.5: The Control Panel.

text file as input, and a video clip is encoded into frames with such bit rate. The control panel will packetize each frame and the corresponding routing information is inserted into the packet header. Packets are forwarded to the next hop via IP/UDP socket.

The software design and implementation at a source node is the core of our implementation effort. It includes the design and implementation of Control Panel, SF algorithm, LGA algorihtm, H.263 encoder, packetization, and inter-node communication.

- **Control Panel**

  Figure 3.5 shows the GUI of the Control Panel program. The Control Panel provides an interface to the user to view and configure the network topology and available spectrum at

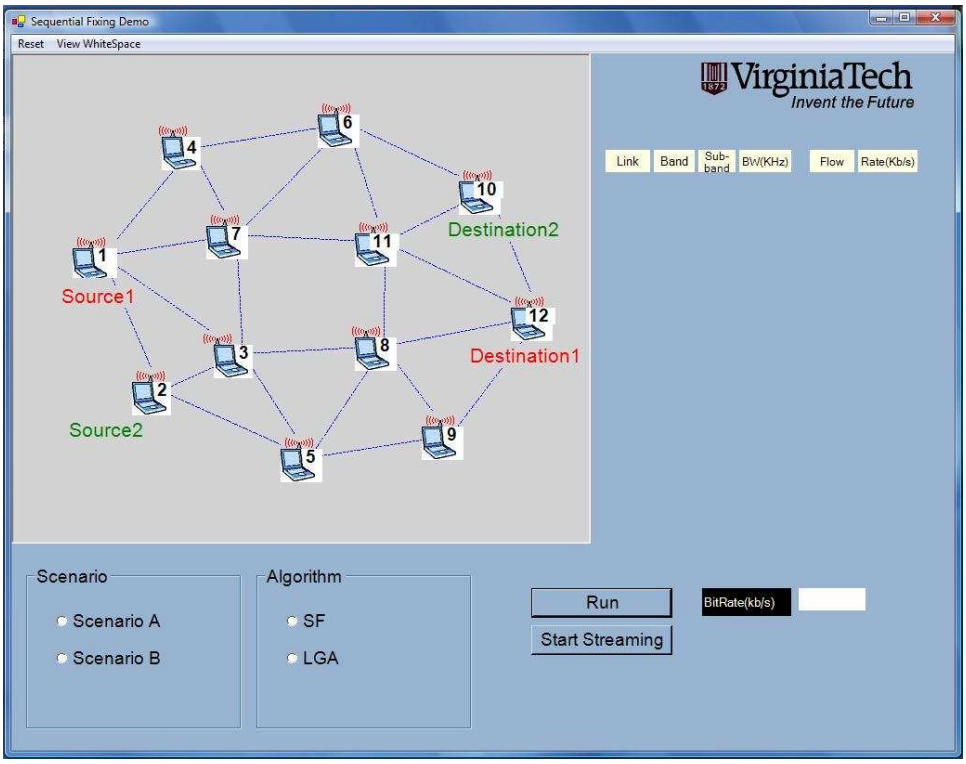


Figure 3.6: The Control Panel: Topology.

each node.

In the emulation testbed, we consider 12-node network and use the physical settings in Table 2.4.

The network topology is shown in Figure 3.6, and the available spectrum bands at each node can be set in Figure 3.7. In this testbed, each node has a set of available bands which can be changed. For instance, in scenario A, node 1 has bands 3 and 5 as available band, and node 2 has bands 2 ,4 and 5. In scenario B, only band 3 is available at node 1, and bands 1 and 4 are available at node 2. The bandwidth (KHz) and the number of allowed sub-bands for each band are also shown in the middle of Figure 3.7. Active communication sessions

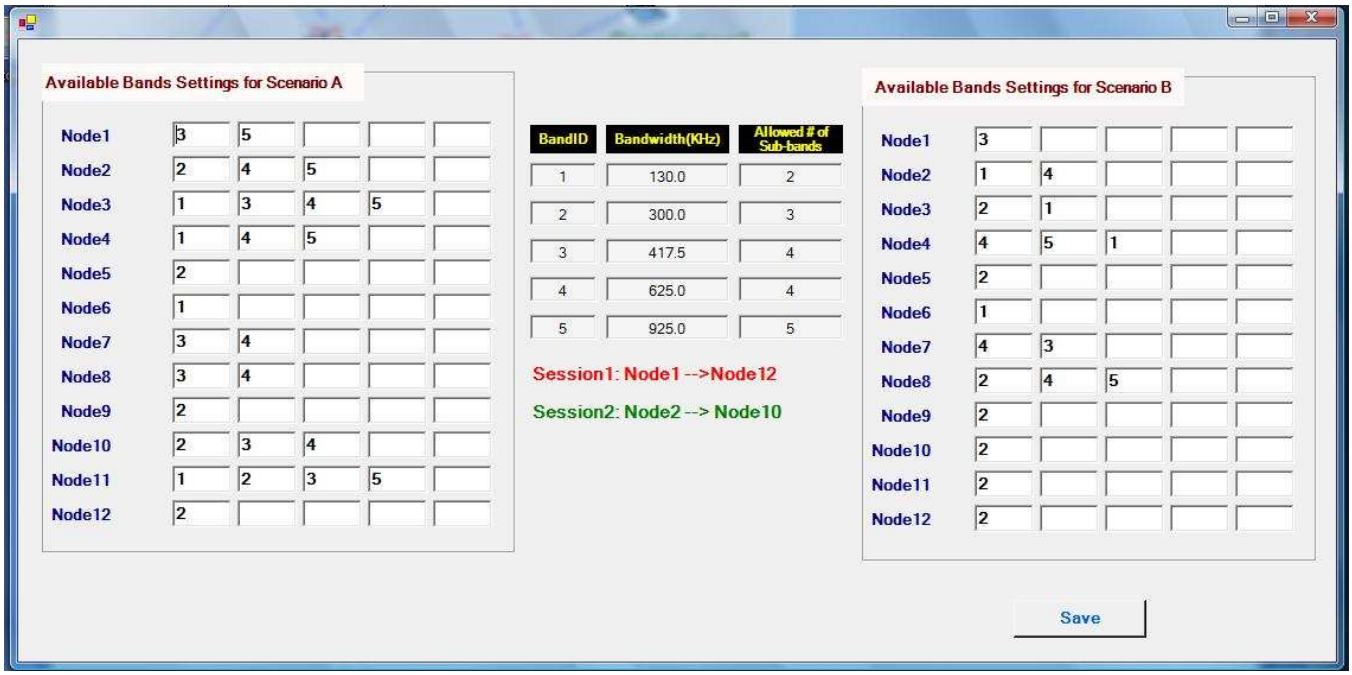


Figure 3.7: Available spectrum at each node

in the network are also shown in the window. All these spectrum availability, number of sub-bands and active session information are saved in a text file, which is used as input for computation.

On the Control Panel, we can select different scenarios as well as different algorithms to run. After we make these settings, the source node Control Panel will broadcast the spectrum information and the chosen algorithm to all the source nodes in the network.

- **Sequential Fixing Algorithm**

The key part in the implementation of SF algorithm is to solve an LP problem. LINDO Application Programming Interface (API) [26] is incorporated for this purpose. LINDO API is designed to solve a wide range of optimization problems including linear programs, mixed

integer programs, quadratic programs and so forth. Solving an LP problem with LIDNO API involves the following steps:

- create LINDO environment;
- create and specify a model in the environment;
- perform the optimization;
- retrieve the status and model solution.

The main task here is to build up a model, along with its specification. This includes setting of the number of constraints, the number of variables, the direction of the optimization, the coefficients of the objective function, the constant in the objective function, the right-hand of the constraints, the coefficients, and the upper and lower bounds on the variables. In our implementation, LINDO API is integrated with C.

- **H.263 Encoder**

  H.263 [27] is a low bit rate video codec designed by the ITU-T. It is an evolutionary codec based on H.261 [31] standards. It supports a wider range of custom source formats and provides a higher coding efficiency than H.261. Moreover, H.263 supports scalable video coding which can improve the delivery of video information in error-prone, packet-lossy, or heterogeneous environments by allowing multiple display rates, bit rates, and resolutions. The high coding efficiency makes the H.263 family widely employed by video conferencing applications, such as the Microsoft NetMeeting.

  In our emulation testbed, we use the H.263 source code from [28]. The H.263 encoder

encodes a YUV format video clip in a given frame rate and target rate. We set the frame rate at 20 frames per second, and the target rate is determined by the Control Panel.

- **Packetization**

  With the paths inforation from the output file of the algorithm, we employ source routing technique in the packetization procedure. After SF algorithm decides the routing paths, packetization program gives each packet a sequence number and adds the list of Node ID into the packet header.

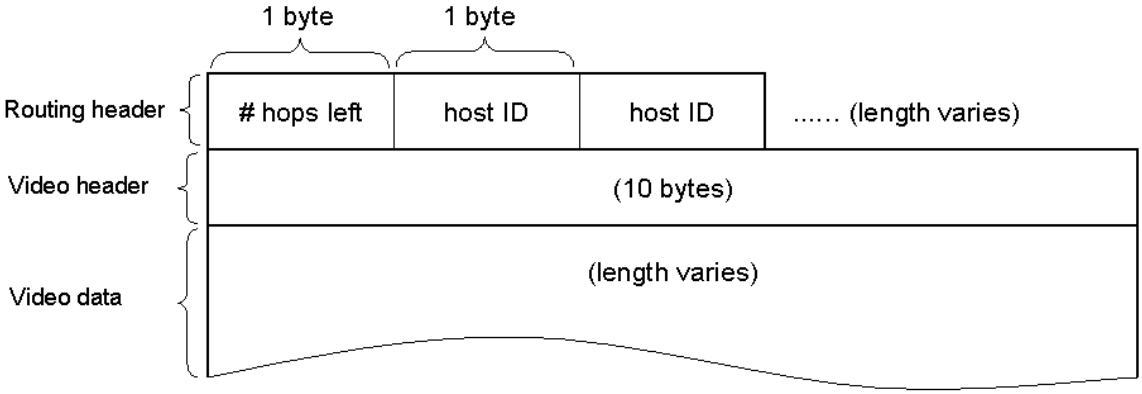


Figure 3.8: Format of video packet [35].

Similar to the technique explored in [35], the video packet header is shown in Figure 3.8, which consists of two parts: routing header and video header. The routing header contains the information about source routing and is used by all the routers on the path to carry out forwarding. Specifically, it contains

  – number of remaining hops.

  – host ID.

In the emulation testbed, we use a subnet mask of 255.255.255.0 for all the IP address of the nodes in the network (the first three bytes of a node's IP address is 192.168.1). In the routing header, only the host ID (identified by the last byte of the node's IP address, instead of the whole IP address) is stored. When a router checks the routing header for next hop address, the router gets the host ID of the next hop node. Combining the next hop ID with the first three bytes of IP address, the next hop IP can be calculated.

Since the routing solution for the SF algorithm may be multi-path, the video packetization program will assign each packet an end-to-end path based on the routing solution. In the packetizaiton implementation we use the following formula to determine how many packets are assigned to each path:

Given n paths, the rate for path i is R_i, where $1 \leq i \leq n$. In each round there are k packets being sent through these n paths. The ith path will send $k \cdot R_i / \sum_{i=1}^{n} R_i$ packets. For example, suppose that the SF algorithm results in two paths A and B, where the bit rate on A is 60 Kb/s and bit rate on B is 40 Kb/s, the packetization server will send the first 3 packets through path A, and the next 2 packets through path B, and so forth.

3.3.2 Intermediate Relay Node

The router program is the kernel program of our emulation testbed. Router keeps listening to the incoming sockets. Upon receiving a video packet, the router will check the packet header for the next hop address and forward the packet.

When a router decides to forward a video packet, it will decrease the number of hops field by one, and remove the first host ID (which is the host ID of the next hop) from the header. After updating the packet header, this packet is forwarded to the next node along the path.

3.3.3 Destination Node

A destination node decodes received frames and displays the video. Since in the SF algorithm, the packets are forwarded through multi-path, video packets must be buffered before they are used for decoding. In the receiver initialization phase, a block of memory buffer is allocated for storing frames. When a video packet is received, the buffer management module in the destination program will save the packet in the correct position of the buffer according to the packet sequence number. When the buffer is half full, we will start decoding and display each frame. In our implementation, we set the buffer to be 500 frames so that the video can be streamed smoothly. In case of missing frames, we simply re-play the last decoded frame.

To compare different scheme, we display the current streaming and previously received video clip in the same window (Figure 3.9), the latter of which was saved in a temp file. To compare the video qualitatively, both the PSNR curves of the current stream and replayed stream are displayed. The GUI at destination decodes and displays the received video, as shown in Figure 3.9.

Computing PSNR

Peak Signal-to-Noise Ratio (PSNR) is the ratio between the maximum possible value of a signal and the value of corrupting noise that affects the fidelity of its representation [32–34]. PSNR can

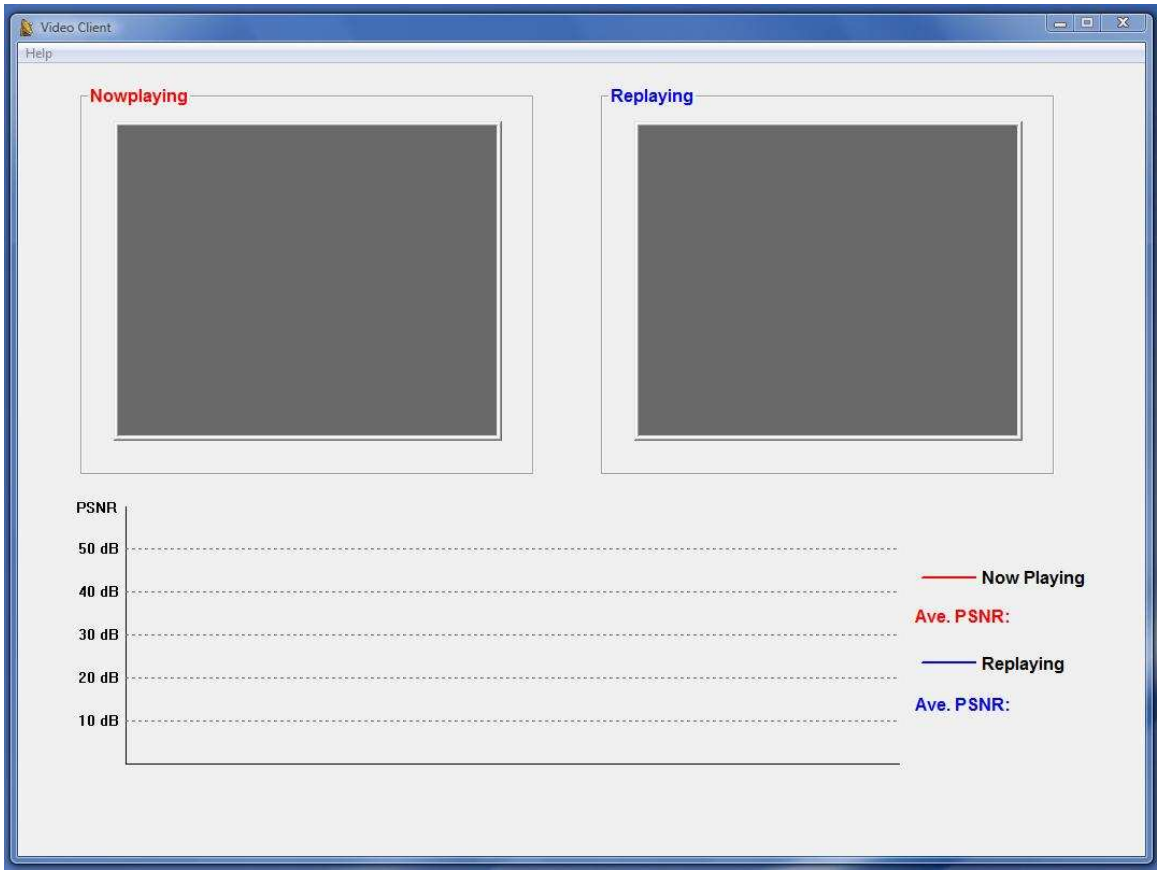


Figure 3.9: Output window at destination node.

be used as a measurement of the quality of a reconstructed image, and is a popular metric for image compression and video quality measurement.

With the wide dynamic signal range, PSNR is usually expressed in terms of the logarithmic decibel scale [34]. Assume the source image $f(i,j)$ has a format of $M \times N$ and a reconstructed image $F(i,j)$, which is reconstructed by decoding the encoded version of $f(i,j)$. For YUV images, error metrics are computed on the luminance signal (Y) only, so the pixel values $f(i,j)$ range between black (0) and white (255).

PSNR is most easily defined via the mean-squared-error (MSE) of the reconstructed image which is considered a noisy approximation of the original one. Mathematically, it is defined as follows:

$$MSE = \frac{\Sigma_{i=0}^{M-1}\Sigma_{j=0}^{N-1}[f(i,j) - F(i,j)]^2}{M \times N} \ .$$

PSNR in decibels (dB) is then defined as

$$PSNR = 10log_{10}(\frac{{MAX_I}^2}{\sqrt{MSE}}) = 20log_{10}(\frac{MAX_I}{\sqrt{MSE}}) \ .$$

Generally, when samples are represented with B bits/sample, maximum possible value of MAX_I is $2^B - 1$ [34]. In our case, MAX_I is the maximum pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. For a typical image compression, a PSNR value is between 30 to 50 dB.

Chapter 4

Experiments

4.1 Objective

In this chapter, we conduct experiments to evaluate our proposed approach and demonstrate its potential. Specifically, we want to demonstrate

- SF algorithm is a viable approach to compute near-optimal spectrum sharing for all nodes in the network;
- SF can be practically implemented on a testbed;
- SF outperforms another algorithm that is based on layered (decoupled) approach.

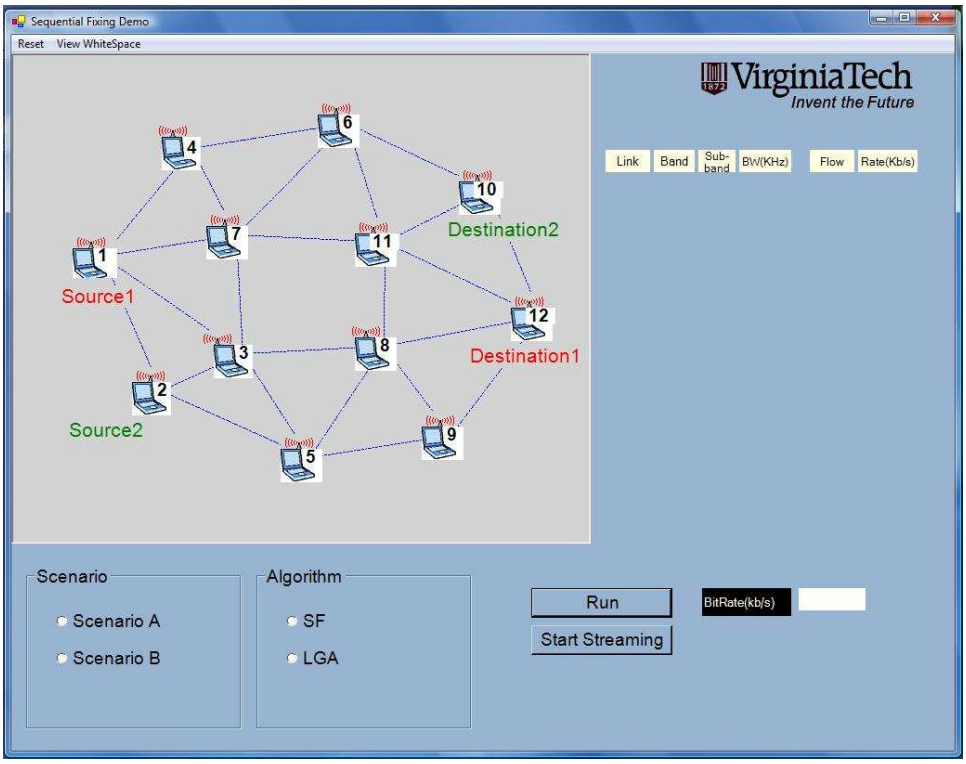


Figure 4.1: Network topology in experiments.

4.2 Network Setting

Throughout the following three experiments, we use the following network topology as shown in Figure 4.1.

In the emulation experiments, we keep the same units for distance, rate, bandwidth and power density as the simulation experiment in Section 2.5. We consider 12 nodes in the network, among which there are two active user communication sessions. An initial rate of 20 Kb/s is allocated to each of these sessions.

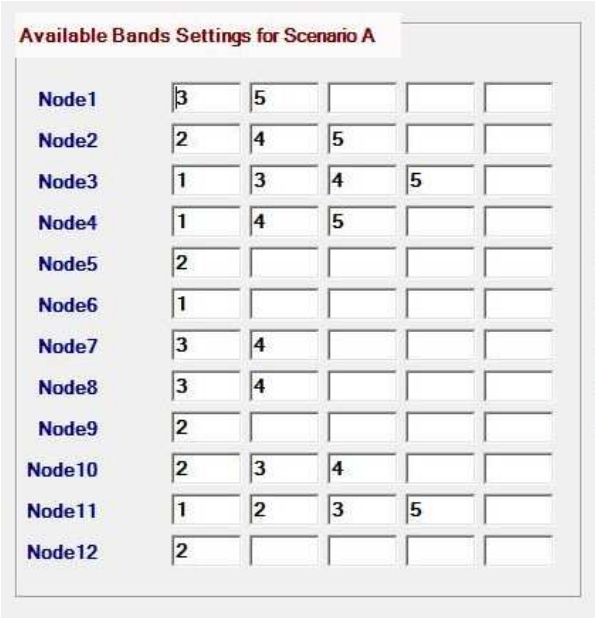
The transmission range at each node is identical, which we assume to be 50 m. The interference

range is set to 75 m. The path loss index is assumed to be 4. The background noise η is assumed to be $1 \cdot 10^{-9}$W/KHz. The power spectral density is assigned to be $1 \cdot 10^{-3}$W/KHz.

We assume there are five bands that can be used for the entire network. In the experiment, each CR node has a subset of these five bands available based on its location. The available bands at any two nodes in the multi-hop CR network may not be identical. We assume band I to V can be further divided into 2, 3, 4, 4, and 5 sub-bands although other desirable divisions can also be used. Details of the frequency spectrum allocation are shown in Table 2.3. Under SF algorithm, within each band, the sub-bands may be of unequal size.

4.3 Video Codec

Once SF gives a solution, we stream video clip through the network based on this solution. We create two video clips from movie “World Trade Center” and “True Lies” in YUV format for each active user session. The H.263 coder will encode the video clip with the bit rate provided by the SF algorithm. The frame rate of H.263 Codec is 20 frames/sec. The resolution of video is in QCIF format (176×144). During the demo, 1500 frames will be transmitted from the source node to the destination node.



Available Bands Settings for Scenario A

Node1	3	5			
Node2	2	4	5		
Node3	1	3	4	5	
Node4	1	4	5		
Node5	2				
Node6	1				
Node7	3	4			
Node8	3	4			
Node9	2				
Node10	2	3	4		
Node11	1	2	3	5	
Node12	2				

Figure 4.2: Spectrum setting in Scenario A.

4.4 Experiments

In this thesis, we conduct three experiments comparing the SF and the Layered Greedy Algorithm (LGA). The details of LGA is given in Appendix A.

4.4.1 Experiment 1: SF Algorithm

In this experiment we want to show performance of the SF algorithm. Given the spectrum setting in Figure 4.2 (Scenario A), we first run the SF algorithm (which can be done within 10 seconds) and then transmit the video for each session. The bandwidth allocation and the flow bit rate result is shown in Figure 4.3, and the video stream from source node 1 to destination node 1 is displayed at destination node 1 (see Figure 4.4).

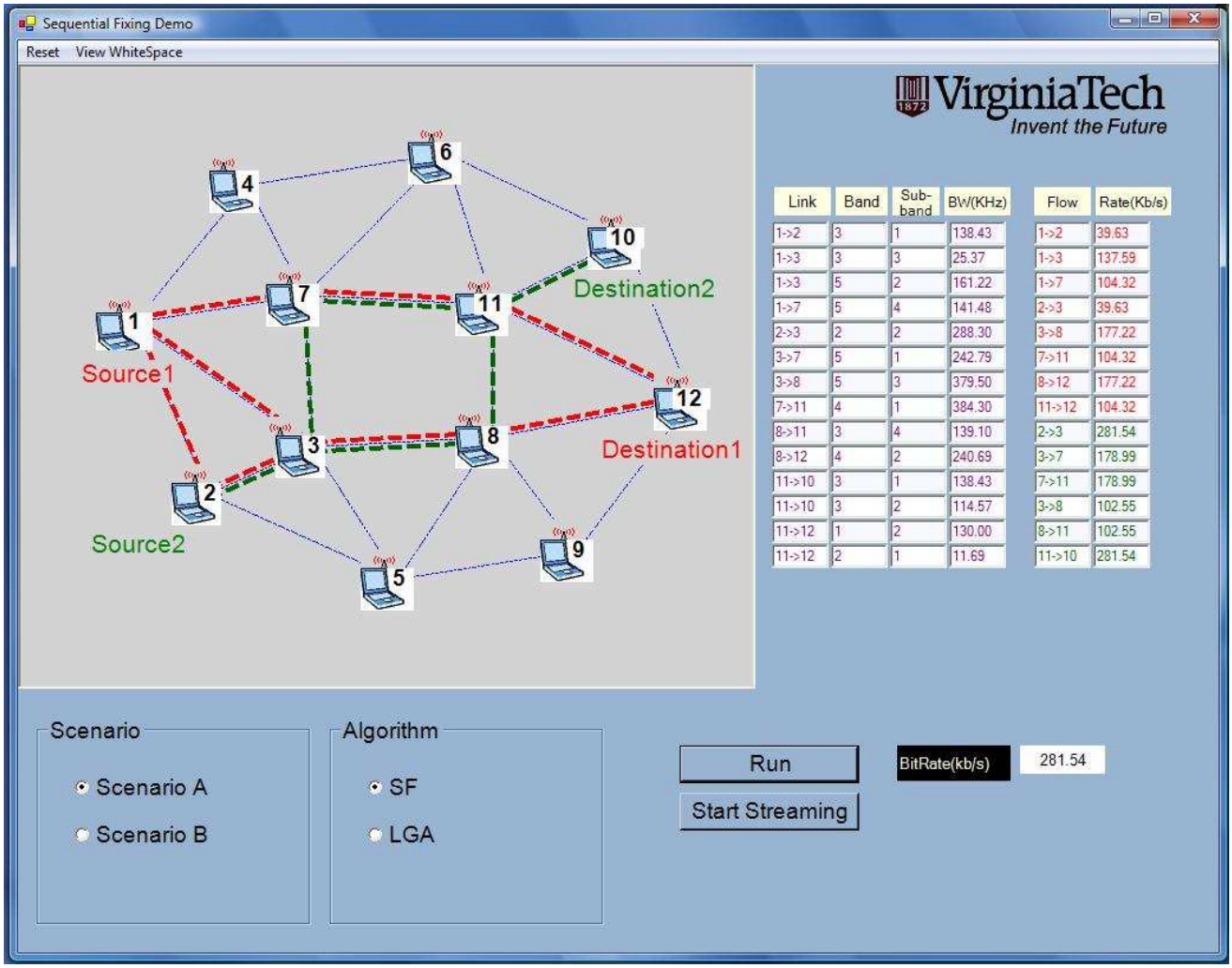


Figure 4.3: SF result for Scenario A.

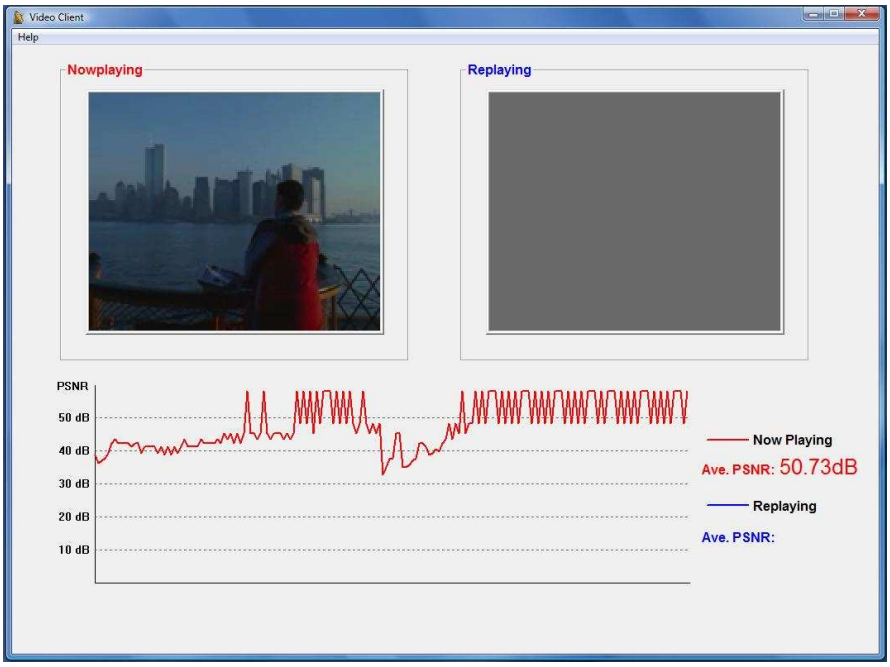


Figure 4.4: A snap shot of video displayed for scenario A under SF algorithm.

4.4.2 Experiment 2: Comparison of SF to LGA Under Scenario A

LGA is tested and compared to SF in the experiment. Figure 4.5 shows the bandwidth allocation and the flow bit rate information in each session. Once the flow bit rate is determined by LGA, we encode the video according to this rate. Figure 4.6 shows a snap shot of the video under LGA (left window), with the right window displayed the playback of the video under SF algorithm. As shown in Figure 4.6, the average PSNR under SF algorithm is 50.43 dB, which is better than the average PSNR under LGA (45.01 dB).

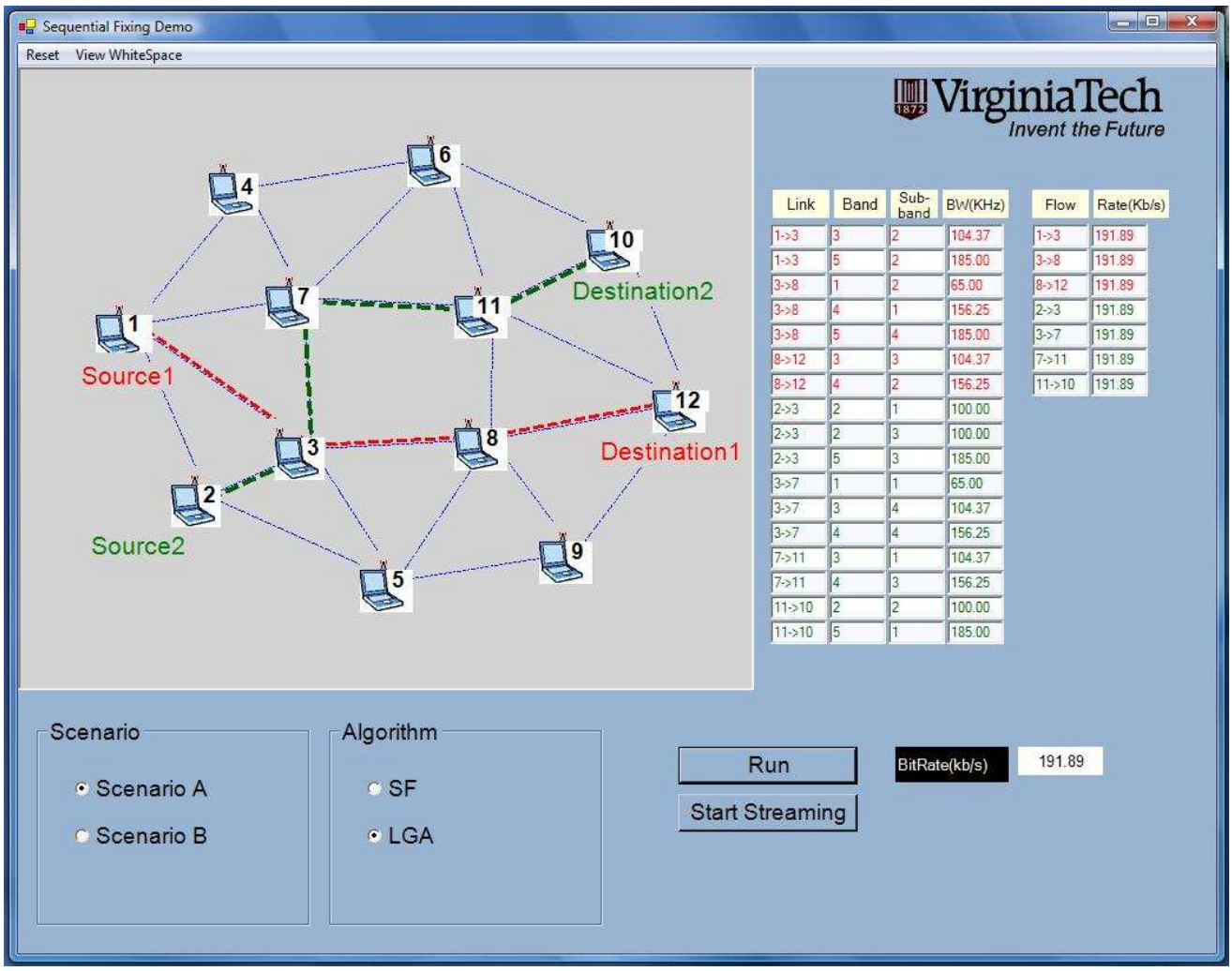


Figure 4.5: LGA result for Scenario A.

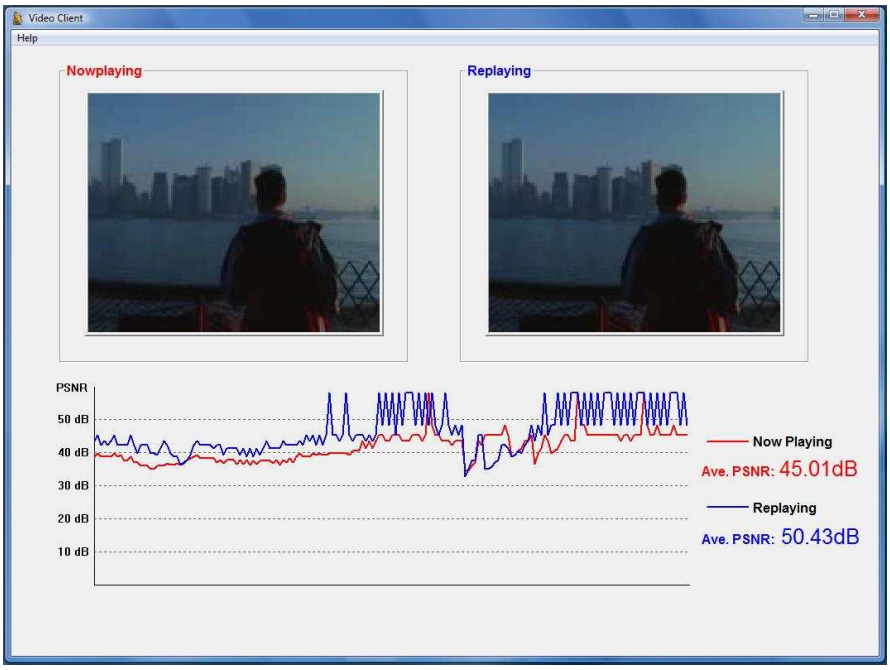


Figure 4.6: A snap shot of video displayed for scenario A comparing LGA with SF algorithm.

4.4.3 Experiment 3: SF vs LGA Under Scenario B

We now let spectrum change to scenario B. Details for Scenario B are shown in Figure 4.7. Comparing with Scenario A, Scenario B has less available spectrum in the network.

First we run SF algorithm again as we did in the first experiment. The result is shown in Figure 4.8 and Figure 4.9. The solution provided by SF algorithm has a network throughput of 155.20 Kb/s. We then test how LGA algorithm performs in Scenario B. As discovered from Figure 4.10 and Figure 4.11, the network throughput by LGA is 47.91 Kb/s. The PSNR average for the video encoded by the SF algorithm rate is 44.16 dB, and the PSNR average for the one encoded by LGA rate is 37.13 dB. Thus, we find that the result calculated by SF algorithm is much better than LGA does.

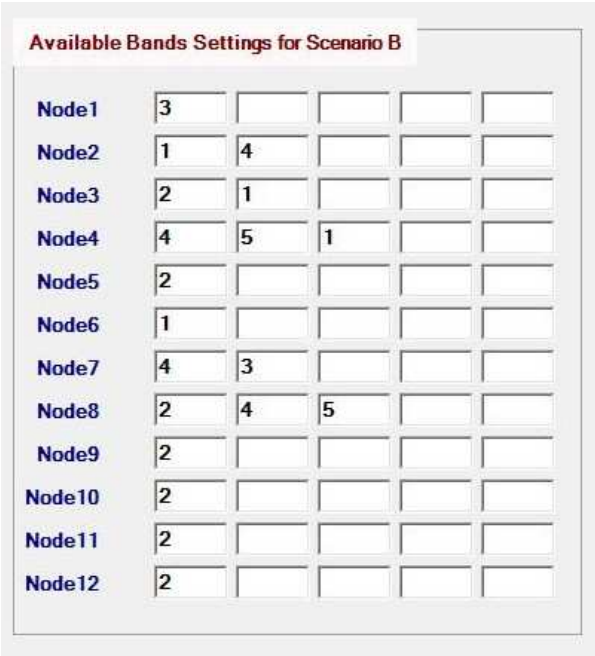


Figure 4.7: Spectrum setting in Scenario B.

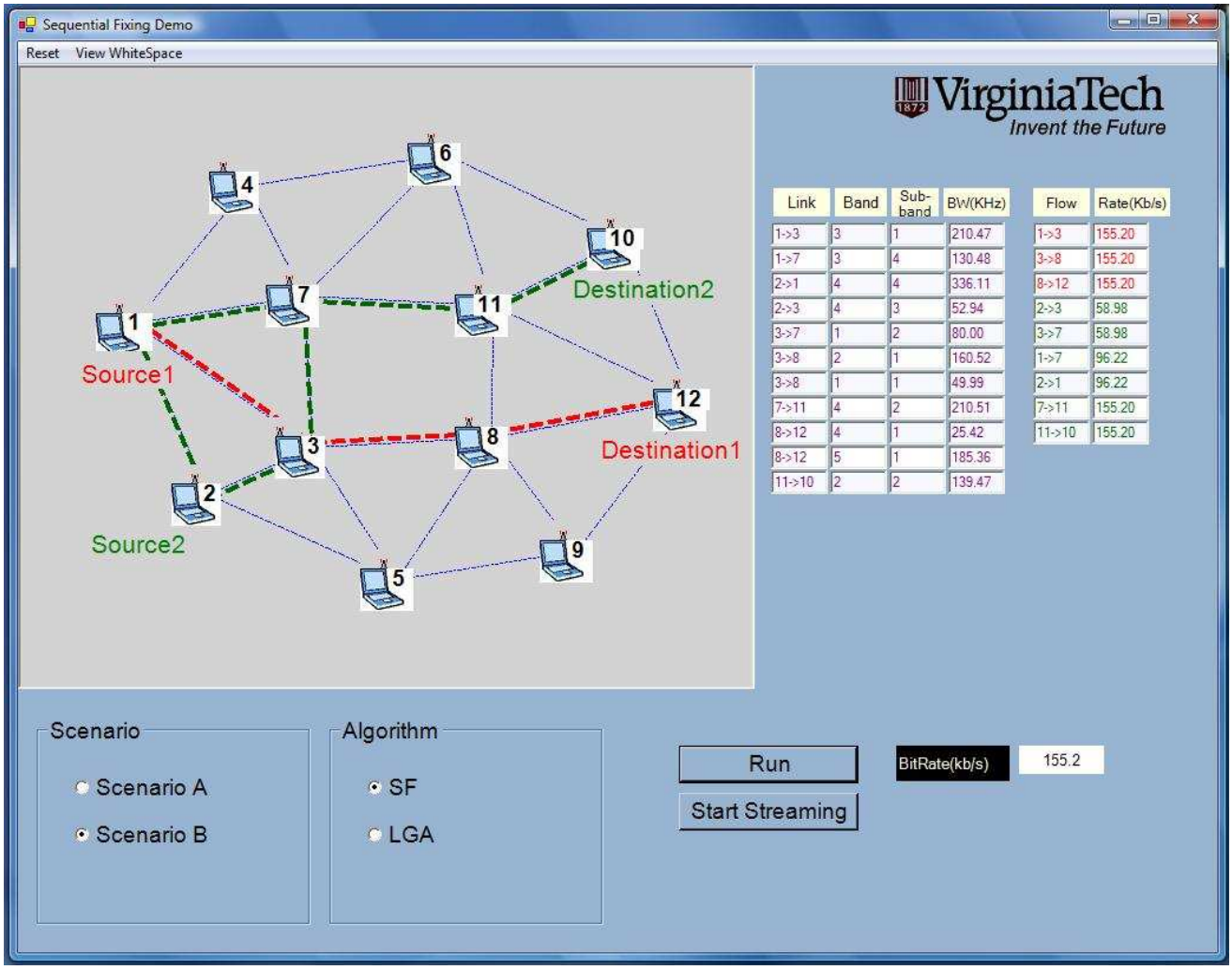


Figure 4.8: SF result for Scenario B.

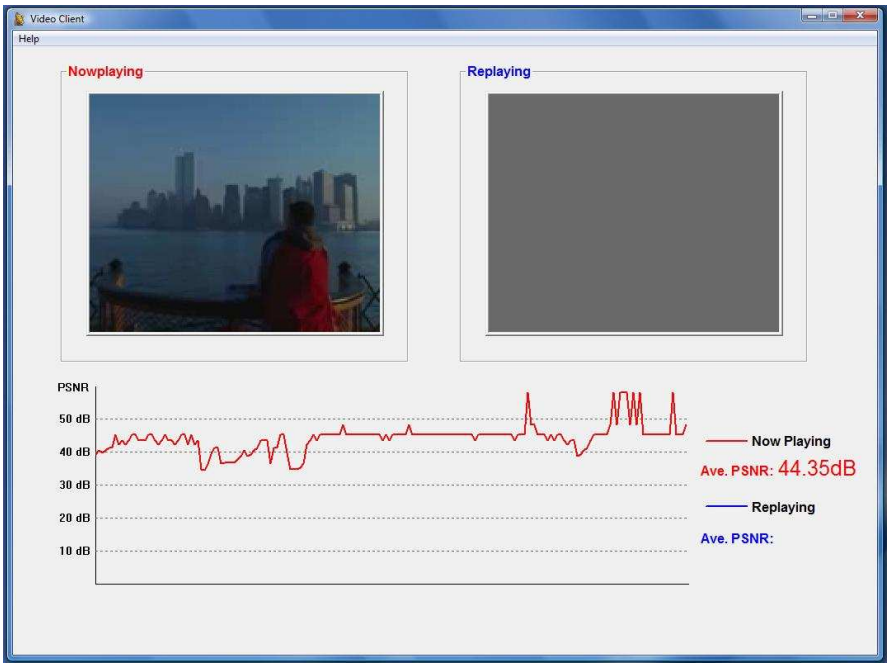


Figure 4.9: A snap shot of video displayed for scenario B under SF algorithm.

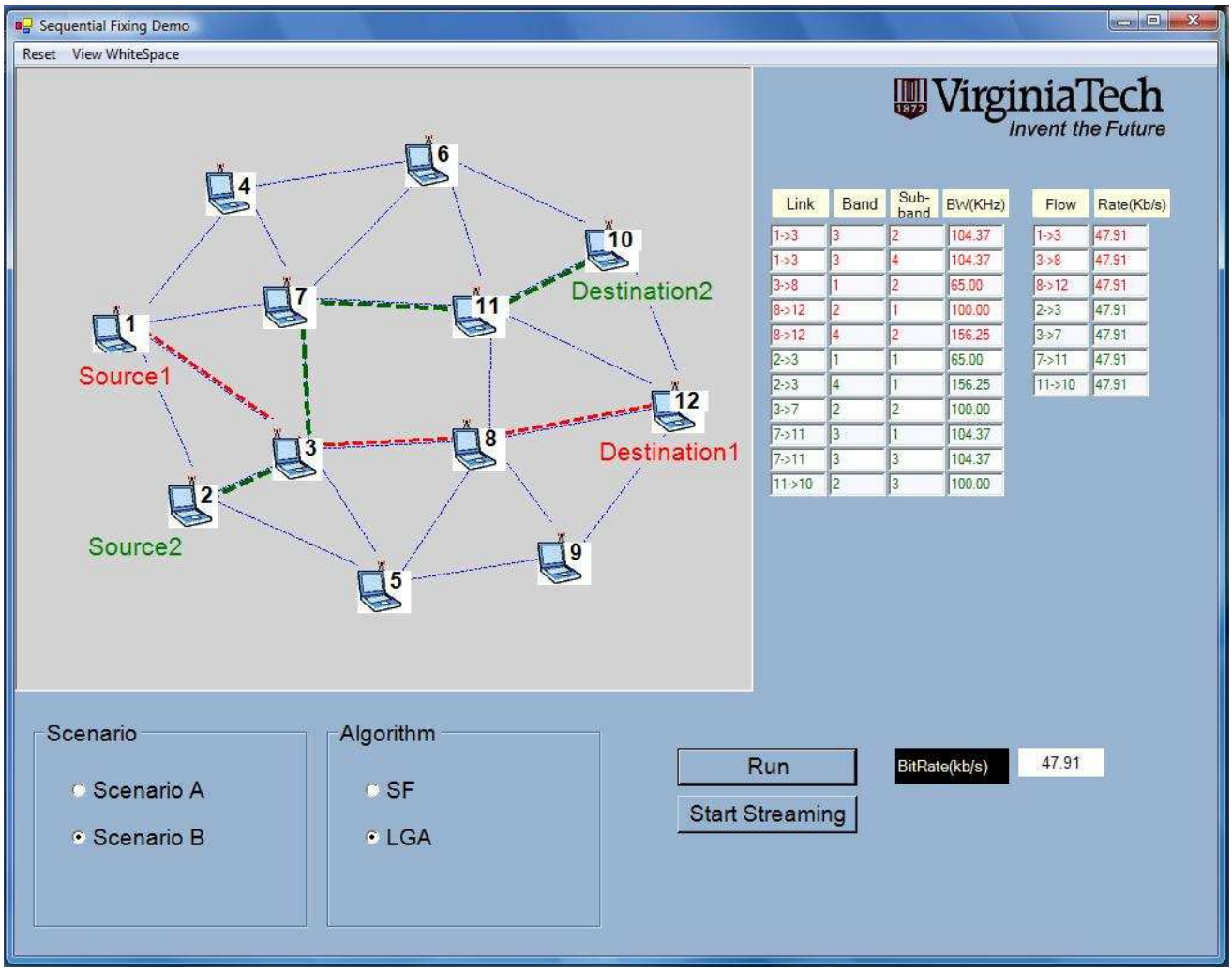


Figure 4.10: LGA result for Scenario B.

Figure 4.11: A snap shot of video displayed for Scenario B comparing LGA with SF algorithm.

Chapter 5

Conclusions

In this thesis, we studied the important problem of spectrum sharing in a multi-hop CR network. The programmability of radio nodes enabled by CR plays a key role in efficient spectral sharing. The behaviors and constraints for a multi-hop CR network is characterized and modeled. A mathematical optimization problem is formulated with the objective being maximizing the network throughput for a set of user sessions. The optimal solution includes the division of available frequency bands into sub-band in the network, the scheduling of sub-bands for transmission and reception at each node, and multi-path routing for each session. Since the problem formulation is a mixed integer non-linear programming (MINLP), which is NP-hard in general. We design a centralized algorithm based on an SF technique, where the integer variables are determined iteratively via a sequence of linear programs. Simulation results show that solutions obtained by this algorithm are very close to the upper bounds obtained via relaxation, thus suggesting that the solution produced by the algorithm is near-optimal.

For comparison purpose, we also modified and implement another algorithm, called Layered Greedy Algorithm (LGA). To evaluate the performance of our approach, we designed and implemented an emulation testbed. This testbed is able to conduct demo experiments with video applications in multi-hop CR networks. Based on the experiment results, we find the SF algorithm is a viable approach to maximize network throughput, which outperforms LGA.

Bibliography

[1] J.H. Reed, *Software Radio: A Modern Approach to Radio Engineering*, Prentice Hall, NJ, May 2002.

[2] Y. T. Hou, Y. Shi, and H. D. Sherali, "Optimal spectrum sharing for multi-hop software defined radio networks," *in Proc. IEEE Infocom*, Anchorage, AL, May 6–12, 2007.

[3] I.F. Akyildiz, W.Y. Lee, M.C. Vuran and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks Journal (Elsevier)*, volume 50, issue 13, pp. 2127–2159, September 2006.

[4] J. Mitola and G.Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, August 1999.

[5] Federal Communications Commission Strategic Goals: http://www.fcc.gov/sptf/reports.html.

[6] M. McHenry , "Spectrum white space measurements," presented to New America Foundation BroadBand Forum, June 20, 2003.

[7] M. McHenry and D. McCloskey,

"New York City Spectrum Occupancy Measurements September 2004," available at http://www.sharedspectrum.com/inc/content/measurements/nsf/NYC_report.pdf.

[8] FCC Spectrum Policy, ET Docket No. 02-135 available at

http://www.imsasafety.org/wireless/docket02135.pdf.

[9] Cognitive Radio Technologies Proceeding (CRTP), ET Docket No. 03-108

http://www.fcc.gov/oet/cognitiveradio/.

[10] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proc. ACM Mobicom*, pp. 73–87, Cologne, Germany, August 2005.

[11] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proc. ACM Mobicom*, pp. 58–72, Cologne, Germany, August 2005.

[12] J. Zhao, H. Zheng, and G. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *Proc. IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 259–268, Baltimore, MD, Nov. 8–11, 2005.

[13] D. Ugarte and A.B. McDonald, "On the capacity of dynamic spectrum access enabled networks," in *Proc. IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 630–633, Baltimore, MD, Nov. 8–11, 2005.

[14] C. Xin, B. Xie, and C.-C. Shen, "A novel layered graph model for topology formation and routing in dynamic spectrum access networks," in *Proc. IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 308–317, Baltimore, MD, Nov. 8–11, 2005.

[15] M.E. Steenstrup, "Opportunistic use of radio-frequency spectrum: A network perspective," in *Proc. IEEE DySPAN*, pp. 638–641, Nov. 8–11, 2005, Baltimore, MD.

[16] Y. Shi and Y.T. Hou, "Optimal power control for multi-hop software defined radio networks," in *Proc. IEEE Infocom*, Anchorage, AL, May 6–12, 2007.

[17] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness, W.H. Freeman and Company*, pp. 245–248, New York, NY, 1979.

[18] B. Radunovic and J.-Y. Le Boudec, "Optimal power control, scheduling and routing in UWB networks," *IEEE Journal on Selected Areas in Communications*, 22(7): 1252–1270, Sept. 2004.

[19] P. Kyasanur and N.H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Proc. IEEE Wireless Communications and Networking Conference*, pp. 2051–2056, New Orleans, LA, March 13–17, 2005.

[20] P. Kyasanur and N.H. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," in *Proc. ACM Mobicom*, pp. 43–57, Cologne, Germany, August, 2005.

[21] BARON Global Optimization Software, http://archimedes.scs.uiuc.edu/baron/baron.html.

[22] Network simulator (OPNET), OPNET Technologies Inc., http://www.opnet.com.

[23] The Network Simulator: ns-2, http://www.isi.edu/nsnam/ns/.

[24] Andrea Goldsmith, Wireless Communications, Cambridge University Press, 2005, pp. 371.

[25] T. Lin, S.F. Midkiff, and J.S. Park, "A dynamic topology switch for the emulation of wireless mobile ad hoc networks," in *Proc. IEEE Conference on Local Computer Networks (LCN)*, Tampa, FL, Nov. 2002, pp. 791-798.

[26] *LINDO API, The Premier Optimaization Engine*, LINDO Systems INC., July, 2003.

[27] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849-866, Nov. 1998.

[28] H.263 source code:

http://eeweb.poly.edu/ yao/VideobookSampleData/doc/download/download_main_nav.htm.

[29] MPEG-4: http://www.mpeg4.net/.

[30] VP6: http://www.vp6-board.com/.

[31] H.261: http://www.h261.com/.

[32] A.N. Netravali and B.G. Haskell. *Digital Pictures: Representation, Compression, and Standards*. Plenum Press, New York, NY, 1995.

[33] M. Rabbani and P.W. Jones. Digital Image Compression Techniques, *SPIE Optical Engineering Press*, Vol TT7, Bellvue, Washington, 1991.

[34] PSNR: http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

[35] X. Wang, "Design and implementation of an emulation testbed for video communication in Ad Hoc networks," Master Thesis, Virginia Tech, Jan. 2005.

Appendix A

Layered Greedy Algorithm

In addition to SF algorithm, we also implement Layered Greedy Algorithm (LGA) in the testbed and compare it to SF. LGA is a natural approach to address complex cross-layer optimization problem, which first fixes the routing and then determines the scheduling in a greedy manner. LGA for CR network consists of the following two steps.

Step 1: Determining Routes

The objective of the LGA is exactly the same as what we discussed in Section 2.3, and the CR network modeling is basically the same as what we have done in Section 2.2 except for the routing part. We consider the same scheduling constraints in LGA.

The algorithm determines the path by Minimum Hop Routing (MHR). It will first calculate the MHR path for each session by a breath-first-search approach. For each active user session (from its source node toward the corresponding destination), we may have a set of paths containing

the same minimum number of hops. We do an exhaustive search for all the combinations of MHR paths for the active sessions. The possible capacity on link (i, j) in sub-band (m, k) is $c_{ij}^{mk} = W^{(m)} \log_2(1 + \frac{g_{ij}Q}{\eta})$. If one link is shared by multiple sessions in a certain combination, we allocate the link capacity to each session proportionally. Among all the paths, the one with the maximum possible scaling factor is selected.

Step 2: Solving the Scheduling Problem

In the scheduling of LGA, the frequency band is equally divided. Initially, we set all links' scaling factor as 0. In each iteration, we select the bottleneck link which has the least scaling factor. We then try to assign sub-bands to such bottleneck link. The same scheduling constraints are considered in LGA scheduling. We count how many times each sub-band has been used in the paths. We then choose the least used sub-band and assign this sub-band if it satisfies all the scheduling constraints. The scheduling terminates if no more sub-bands can be assigned into the bottleneck link.

Layered greedy algorithm

Find MHR paths with maximum possible scaling factor.

1. Use breath first search to find all MHR paths for each session.
2. If there are multiple MHR paths for the sessions, we choose the path that offers the maximum possible scaling factor.

Scheduling

1. Initially, all links' scaling factor are set to 0.
2. For the bottleneck link $i \rightarrow j$ with the least scaling factor, we assign the least used available sub-band (m, k) to node *i*, i.e., set $x_{ij}^{mk} = 1$ and update the scaling factor, where sub-band (m, k) is available if it satisfies all the scheduling constraints.
3. If no more sub-bands can be assigned to the bottleneck link, LGA terminates.

Figure A.1: Layered Greedy Algorithm.

Appendix B

Vita

Tong Liu received his B.E. in Information Engineering from NanKai University, Tianjin, China in July 2002. In 2005, he joined Virginia Tech Department of Computer Science as a Master's degree student. He was a Graduate Teaching Assistant (GTA) from August 2005 to May 2006. During May 2006 to July 2007, he was a Graduate Research Assistant (GRA) and Webmaster of Wireless@VT Research Center.