8-2018

A Multiple-frequency Transmission Scheduling Scheme for Mesh Networks that Employ Spectrum Sharing

Taylor Maier
Clemson University, tmaier@g.clemson.edu

A Multiple-frequency Transmission Scheduling Scheme for Mesh Networks that Employ Spectrum Sharing

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Taylor Maier
August 2018

Accepted by:
Dr. Harlan Russell, Committee Chair
Dr. Carl Baum
Dr. Daniel Noneaker
Dr. Jacob Sorber

Abstract

The reallocation of underutilized spectrum becomes increasingly important as available frequency bands continue to diminish. The Citizens Broadband Radio Service (CBRS) presents spectrum-sharing opportunities, but the interference protection it requires for legacy users can create network performance issues for newer users that look to share the spectrum. To combat these issues, we introduce bridge nodes into the network - dedicated traffic-forwarding nodes that help ease the connectivity issues and bottlenecking caused by legacy node protection protocols. In this thesis, we present the augmented connected dominating set (ACDS) algorithm for selecting bridge nodes from a list of bridge node candidates within a network. Simulation results display the effectiveness of this selection scheme in terms of end-to-end success rate, end-to-end throughput, and end-to-end delay in a variety of network topology scenarios. We show the effectiveness of our algorithm by comparing it to simpler bridge node selection schemes.

Dedication

This work is dedicated to my mother - who ignited a spark, kindled the flame, and continues to teach me that all things thrive with love; to my father - whose righteous morality serves as a beacon that will guide me throughout my life; and to my brother - my first friend, my favorite instructor, and my lifelong rival.

Acknowledgments

First, I thank my advisor and committee chair, Dr. Harlan B. Russell, for his guidance throughout this project. Across my entire collegiate career, he has been my favorite professor, and I sincerely believe that any student in our department that graduates without ever having taken a class from Dr. Russell has missed out on something special.

I wish to thank Dr. Carl Baum, Dr. Daniel Noneaker, and Dr. Jacob Sorber for serving on my committee and providing valuable insight.

Clemson University is acknowledged for generous allotment of compute time on Palmetto cluster.

I would also like to thank my close friend and roommate Matthew Bauerlin for his encouragement and advice throughout my graduate studies. Further, I thank my friends Austin Anderson and Blake Gronowski for their reassurances in the waning hours of this project.

Finally, I would like to give a special thanks to all of my friends for their support, both directly and indirectly.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

At present, frequency allocation is scarce. With the ever-growing prevalence of wireless communications in the modern world, there is an ever-increasing urgency to uncover and re-purpose underutilized spectrum allocations. Naturally occurring from this situation is the idea of *spectrum sharing*, a process by which multiple systems share the same frequency band.

One example of such a spectrum sharing scenario is presented in the FCC's description of the Citizens Broadband Radio Service (CBRS) [1]. The CBRS is slated to exist within the 3.5 GHz band. This band was originally reserved for legacy systems, such as off-coast radar, but was found to be underutilized. The general idea is to allow new users to operate uninhibited within the band so long as legacy systems are unhindered by their presence. When legacy systems require access to the band, new users must back-off and give priority to the legacy users.

The CBRS is designed with a three-tiered model. This model includes Incumbent Users (IUs), Priority Access Users (PAUs), and General Authorized Access Users (GAAUs) [2]. The IUs include the legacy systems that require protection from newer users - the PAUs and the GAAUs. As per the CBRS guidelines, IUs are given higher priority to the band than PAUs, and PAUs are given higher priority to the band than GAAUs.

To enforce the priority system set forth by the CBRS guidelines, a centralized controller called the *spectrum access system* (SAS) is implemented. The SAS arbitrates between IUs, PAUs, and GAAUS, forcing lower-priority users to relinquish the band when higher-priority users require it. The logistics of the SAS has been a popular area of study [3], [4], [5]. Although PAUs and GAAUs will likely be able to report self-identification information to the SAS, the nature of IUs

may inhibit them from disclosing such information. Consider a legacy military system in which providing information about position could threaten operations security (OPSEC). To allow the SAS to gather information about IUs without breaching OPSEC, a network of sensors known as the environmental sensing capability (ESC) is put into place. These sensors allow the SAS to detect the presence of IUs without requiring direct information from them. Choosing an appropriate number of sensors, determining the proper locations for the sensors, and managing how the sensing data is utilized by the SAS are further points of investigation for the CBRS [6], [7], [8].

The 3.5 GHz band contains several different frequency channels, and higher-priority users may only require access to a subset of those channels instead of to the entire band. When this situation occurs, the SAS forces lower-priority users to back out of just those applicable channels, such that the IUs, PAUs, and GAAUs can operate within the band simultaneously on separate frequencies. As such, the arrival of IUs into the network creates *exclusion zones* (EZs) for the PAUs and GAAUs - zones in which certain subsets of frequencies are restricted. The size of these exclusion zones are based around the operating requirements of the IUs. In general, the IUs are assumed to contain instruments that are highly sensitive to interference. In order for the IUs to function properly, the SAS must guarantee that the interfering power at an IU's receiver is sufficiently low. To achieve this effect, the SAS forces lower-priority users that are likely to cause unacceptable interference at an IU to relinquish contentious frequency bands. Depending on the topology of the network and the location of the EZs, this situation can cause disconnections and bottlenecking.

We are particularly interested in a version of the CBRS that employs a *mesh network*. A mesh network contains a large number of *backbone nodes* alongside nodes that do not participate in the backbone. Backbone nodes gather traffic from non-backbone nodes and relay this traffic to neighboring backbone nodes. The process of multi-hopping through each of the backbone nodes distributes the traffic throughout the network to its intended destination. The connectivity issues caused by EZs are amplified in the mesh network scenario. We propose a new method by which to enhance connectivity and to improve network performance within the mesh network once EZs are introduced.

We consider a multi-frequency mesh network in which certain areas of the network may be unable to utilize a subset of possible frequencies - these exclusion zones create a situation in which neighboring nodes in the network become unable to communicate using their current frequency. In such a mesh network, it is not the case that all nodes can act as base stations; thus, the network

is required to support multi-hop relaying. To connect two zones of the network that have become disconnected, we can find nodes that are able to operate on frequencies accessible by both zones - these nodes are said to be *bridge node candidates* since they are able to bridge traffic across the barrier created by the EZ. We propose a new mechanism to select bridge nodes from the list of possible candidates in such a network. This mechanism is based on the essential connected dominating set (ECDS) algorithm, and it provides improved network throughput compared to simpler schemes, such as selecting the minimum number of bridge nodes to ensure network connectivity or selecting every node that is eligible to be a bridge.

The rest of this paper is organized as follows: Chapter 2 describes the system model, including the scheduling algorithm and transmission model. Chapter 3 discusses different strategies for selecting bridge nodes, including the new algorithm that we are presenting. Chapter 4 presents simulation results. Chapter 5 concludes the paper and suggests avenues for future research.

Chapter 2

System Models

2.1 Overview

We model our system using CBRS as a base. In the remainder of this paper, we will refer to the IUs as *primary users* (PUs) and to the PAUs and GAAUs as *secondary users* (SUs). A mesh of nodes comprise our network and are governed by a centralized controller that acts as a stand-in for the SAS. These nodes represent the backbone for the secondary users. As such, these nodes have access to a set of frequencies that may be utilized when transmitting data to nearby nodes, but certain frequencies may become unavailable when a primary user enters the network. The introduction of such a primary user creates an *exclusion zone* (EZ) - an area in which certain frequencies may not be accessed by the secondary users.

For our considerations, each of the secondary users functions with two transceivers that can be tuned to non-overlapping frequency bands. The transceivers operate simultaneously without interfering with each other. Under normal operation, each of the nodes serves *clients*, and, as such, each node dedicates one of its transceivers to communicating with clients. Its other transceiver is used for forwarding traffic to other nodes in the network. When a transceiver is being used to communicate with other nodes, it may operate on any frequency that is not restricted by the existence of nearby primary users.

When a primary user introduces an exclusion zone, nodes on the boundary of the zone experience connection issues with nearby nodes. Nodes within the zone must switch to an unrestricted frequency, and nodes straddling the perimeter of the EZ may not be able to communicate with each

other effectively. Our utilized solution to this problem is to convert some of the nodes outside of the exclusion zone into *bridge nodes*. Bridge nodes do not devote an transceiver to listening to clients. (When a node becomes a bridge node, its clients may be picked up by neighboring nodes.) Instead, they utilize both of their transceivers to forward traffic throughout the network. As such, a bridge node outside of an exclusion zone can tune one of its transceivers to an available frequency used within the EZ and tune its other transceiver to a frequency being used outside of the EZ.

As an example, consider a scenario in which the network contains only two possible frequencies: frequency 1 and frequency 2. An example network is shown in Figure 2.1. Consider that a primary user is introduced to the network that restricts frequency 1 within a certain range, as in Figure 2.2. Note that in Figure 2.2, the dashed blue lines represent the links that are affected by the exclusion zone. The introduction of this EZ partitions the area into two sections - one that contains only unrestricted nodes and one that contains only restricted nodes. Notice also that the section containing unrestricted nodes is further split into two pieces. The nodes operating outside of the exclusion zone can still use both frequencies, but nodes on the border must use frequency 2 when communicating with nodes inside of the zone. We can improve the efficiency of the communication from one side of the zone border to the other by introducing bridge nodes. Since the nodes operating outside the EZ can still access both frequencies, we can choose a subset of these nodes to become bridge nodes, as in Figure 2.3. These dedicated traffic-forwarding nodes assist in the process of efficient communication across the zone barrier. Due to the assumed existence of a centralized controller (such as the SAS), we can use a centralized algorithm to select the nodes that will become bridge nodes.

2.2 Lyui's Algorithm

In the network, node transmissions are scheduled using Lyui's algorithm [9]. Under Lyui's algorithm, each node is assigned a *color number* that signifies when it is scheduled to transmit. Each node's color number is dependent upon its *1-neighborhood* and *2-neighborhood*. For a particular node N, its set of *1-neighbors* is comprised of any nodes that share a link with N. Its set of *2-neighbors* is comprised of all nodes that are not themselves 1-neighbors of N but are a 1-neighbor of any of N's 1-neighbors. Node N selects a color number such that it does not share its color number with any node within its 1-neighborhood or 2-neighborhood. Once all nodes are colored in this way, it is

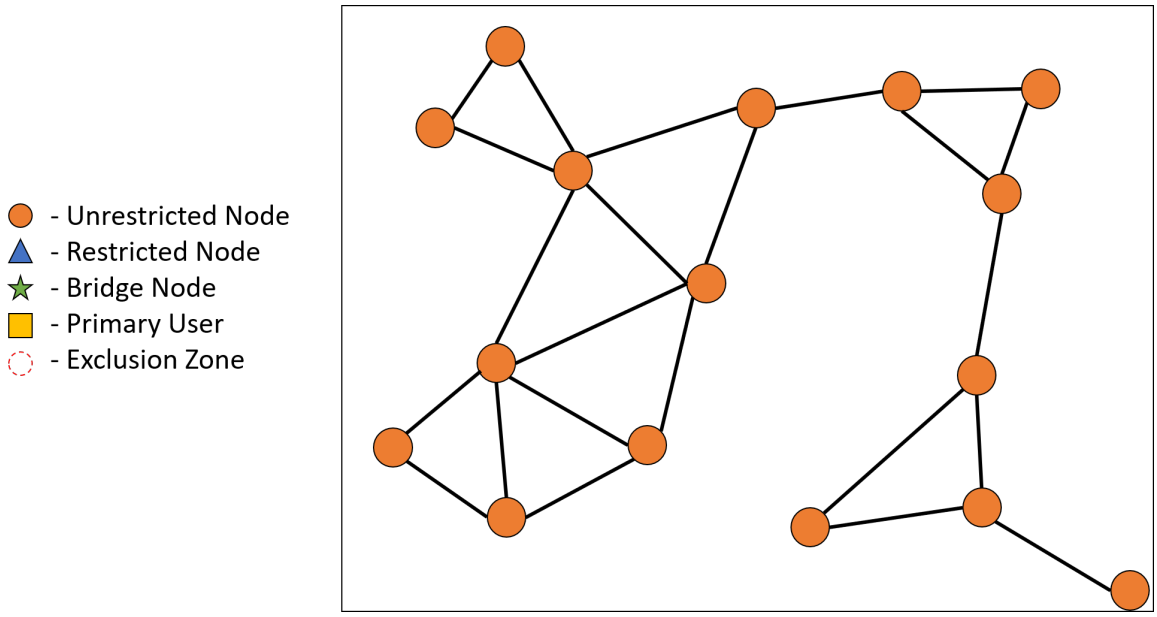

Figure 2.1: Example Network - Before Introduction of Exclusion Zone

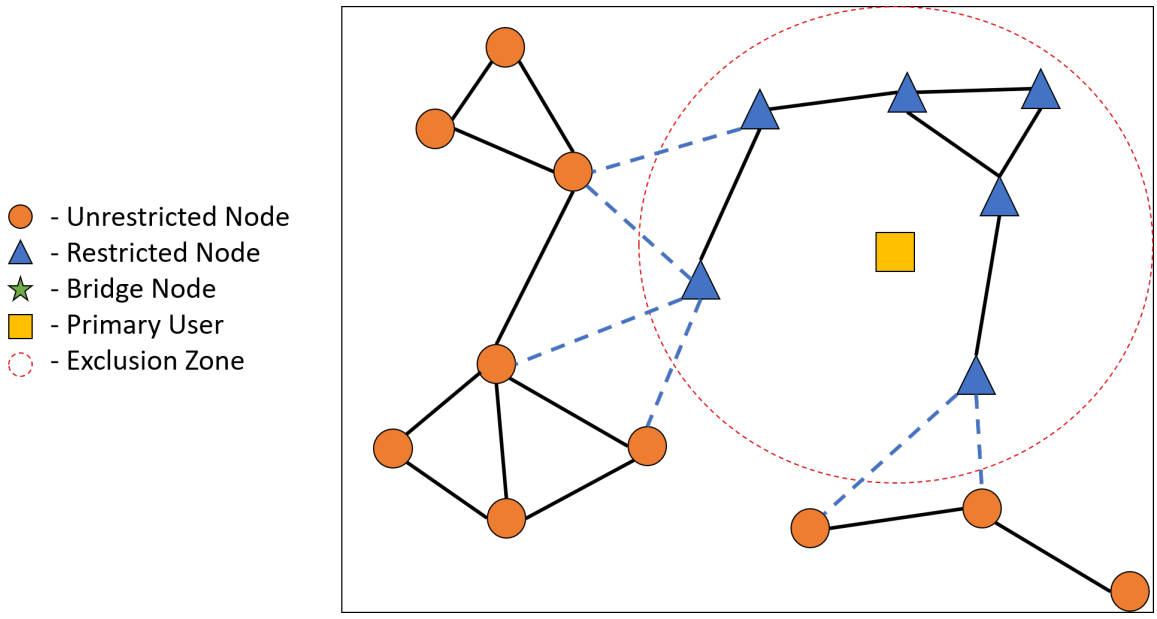

Figure 2.2: Example Network - After Introduction of Exclusion Zone

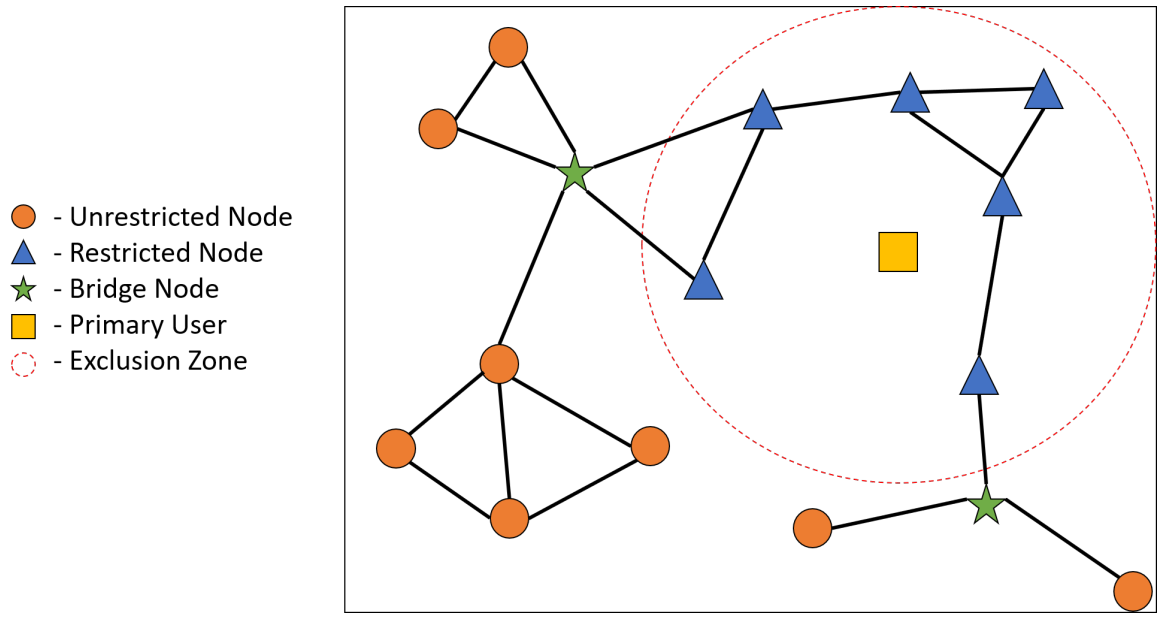

Figure 2.3: Example Network - After Selection of Bridge Nodes

guaranteed that a node will never transmit at the same time as any of its 1- or 2-neighbors.

Define the function $p(x)$ as the minimum power of 2 that is greater than or equal to x. For a transmission being scheduled for time t, a node N with color number c is eligible to transmit if it meets the following criterion:

$$t \bmod p(c) = c \bmod p(c) \tag{2.1}$$

Under Lyui's algorithm, an eligible node is only selected for transmission if its color number is the largest out of all other eligible nodes within its 1- and 2-neighborhoods. Further discussion of the details of Lyui's algorithm can be found in [10].

2.3 Transmission Model

We employ the following model for direct-sequence spread-spectrum modulation to represent the communication between two nodes in the network. Transmissions between nodes are only considered successful if the *signal to noise plus interference ratio* (SNIR) is measured to be above a certain threshold β. Similar to the model presented in [11], we define the SNIR for a transmission

from node i to node j as the following:

$$\frac{P_r(i)NT_c}{N_0 + \sum_{\forall n \neq i} P_r(n)T_c} \tag{2.2}$$

In this equation, $P_r(x)$ represents the power that the receiving node receives from a node x, N represents the spreading factor, T_c represents the chip duration, and N_0 represents the ambient noise at the receiving node. Any transmission that has a SNIR beneath the threshold β is considered unacceptable and is discarded. Our network model does not include any retransmission protocols or packet acknowledgments. As such, if a transmission is unsuccessful, the packet is lost. Successful transmissions do not produce additional acknowledgment traffic. The objective of Lyui's algorithm is to schedule transmissions so that the multiple-access interference is limited and unsuccessful transmissions are rare.

The received power $P_r(x)$ from a node x can be calculated using the following equation:

$$P_r(x) = P_t\Big(\frac{\lambda}{4\pi d_{x,j}}\Big)^{\alpha} \tag{2.3}$$

In this equation, P_t represents the transmission power, λ represents the wavelength, $d_{x,j}$ represents the distance between node x and node j, and α represents the path loss exponent.

Further analysis of this transmission model can be found in Appendix A.

2.4 Detectable Range and Communicable Neighbors

Lyui's algorithm was originally designed for a graph-based communication model that does not account for the SNIR. Since our transmission model is based around SNIR, we use a particular application of Lyui's algorithm to fit our network. As discussed in Section 2.2, a node N considers its set of 1-neighbors as the set of all nodes with which it shares a link. For our consideration, a node N considers any nodes within its *detectable range* to be 1-neighbors. In this context, the term *neighbor range* may also be used to refer to the detectable range. Node N's detectable range is defined as the maximum distance at which node N could receive a transmission from another node X in the absence of any interference. In general, if each node in the network transmits with the same transmission power, the detectable range for each node in the network can be manipulated by changing the transmission power. In our model, we specifically set the transmission power to force

a suitable detectable range.

Notice that even if node X is a 1-neighbor of node N, node X and node N may still be unable to communicate. For example, if node X is at the extreme edge of node N's detectable range, then the presence of *any* interference will cause a transmission between these two nodes to fail. As a result, we introduce a new concept to the network called *communicable neighbors*. If node Y is considered to be a communicable neighbor of node N, then node N is able to receive transmissions reliably from node Y even in the presence of reasonable interference. Put more precisely, transmissions scheduled by Lyui's algorithm between nodes N and Y will have a received SNIR greater than β with high probability.

Although detectable range is a deterministic quantity based on system parameters, each node's set of communicable neighbors will differ based on the level of interference that the node is likely to encounter when receiving a transmission. To reduce the complexity of the problem of each node identifying its own set of communicable neighbors, we instead perform preliminary studies that examine the reliability of a node's neighbors that are within detectable range based on the length of the links to those neighbors. Through experimentation using our system's parameters, we are able to identify a *communicable range* for nodes. Links between nodes that are longer than the communicable range do not provide reliable enough transmissions to guarantee high received SNIRs. As such, a node considers the set of all 1-neighbors that fall within this communicable range to be its set of communicable neighbors. The value of the communicable range is influenced strongly by the multiple-access interference control imposed by Lyui's transmission scheduling algorithm. Note that unlike detectable range, which can be controlled strictly through transmission power, communicable range cannot be directly controlled through system parameters, and must be set based on the results of preliminary experimentation.

When a node N selects a color number during setup for Lyui's algorithm, it knows the color numbers of all nodes within its detectable range; that is, node N knows the color number of all nodes within its 1-neighborhood. Furthermore, node N knows the color numbers of all of the 1-neighbors of its 1-neighbors; that is, node N knows the color numbers of all nodes within its 2-neighborhood. This list of 2-neighborhood color numbers is based on the detectable range only. Note that the nodes within detectable range will include both communicable nodes and non-communicable nodes. In contrast, the routing protocols within the network will choose *only* communicable nodes when constructing forwarding tables and when making routing decisions. Nodes that are detectable but

not communicable are assumed to have links that are too unreliable for forwarding traffic.

Chapter 3

Maintaining Connectivity Through Bridge Selection

3.1 Selection Schemes

For our investigations, we assume that the shared spectrum system operates on two frequencies, termed as *frequency 1* and *frequency 2*. Nodes in the network *prefer* to use frequency 1, and they will always do so unless frequency 1 is unavailable. Note that the set of frequencies that nodes use to communicate with clients is assumed to be mutually exclusive of this set of frequencies that nodes use to forward traffic to other nodes in the network. We are not concerned with these client frequencies in the present investigation. Instead, we focus solely on frequency 1 and frequency 2.

Each node begins as a frequency 1 node. Once exclusion zones are introduced into the network, nodes within these zones lose access to frequency 1 and instead become frequency 2 nodes. Frequency 1 nodes and frequency 2 nodes operate on two different transmission schedules, since frequency 1 transmissions do not interfere with frequency 2 transmissions. As such, Lyui's transmission scheduling algorithm is run separately for the set of nodes on each frequency.

Since nodes on frequency 1 cannot communicate with nodes on frequency 2, the network fractures and becomes disconnected as a result of the exclusion zones. To regain network connectivity, we can choose certain frequency 1 nodes to become bridge nodes. In the context of this paper, *network connectivity* is defined as the state in which any node in the network can reach any other

node in the network through multi-hop relaying on available frequencies.

At first glance, it may seem that the introduction of just one EZ will split the network into just two pieces, but it is possible that the introduction of an EZ will fracture a non-EZ portion of the network. For an example, refer back to Figures 2.1 and 2.2. Here, the introduction of one exclusion zone splits the network into three fragments.

After exclusion zones are added to the network, possible *bridge node candidates* can be identified. For a node to be a bridge candidate, it must first be outside of all exclusion zones in the network. Additionally, it must be able to communicate with at least one node that is within a zone and with at least one node that is not within a zone. Once all bridge candidates are identified, a subset of the bridge candidates is selected according to an algorithm called a *bridge node selection scheme*. Note that by the previous definition, only frequency 1 nodes can be bridge node candidates. The main goal of any bridge node selection scheme is to restore network connectivity. In order to guarantee that network connectivity is regained, it is sufficient that the bridge node selection algorithm connects adjacent EZ and non-EZ fragments with at least one bridge node, as in Figure 2.3.

For the work in this paper, we investigate the effectiveness of six different bridge node selection schemes. The first scheme, named *min*, selects a minimal number of nodes while still restoring full network connectivity. For each pair of adjacent EZ and non-EZ network fragments, exactly one node is selected to become a bridge node, as in Figure 2.3. The node that is selected is done so at random with a uniform distribution among the candidates. Selecting nodes in this way ensures that the network is connected. The second scheme, *max*, selects every possible bridge candidate.

The third scheme, *dom*, creates a *dominating set* out of the list of bridge candidates. In graph theory, a dominating set refers to a subset of nodes within a graph that have the following property: every node within the graph is either a member of the dominating set or is connected to a member of the dominating set by exactly one link [12]. In this selection scheme, a dominating set is constructed out of the list of bridge node candidates via a greedy algorithm, such that each bridge node candidate either becomes a selected bridge node or is a 1-neighbor of a selected bridge node. Put more simply, for a candidate to be selected as a bridge, it must have no 1-neighbors that have already been selected to be a bridge. We modify this dom scheme slightly to make the fourth scheme, *relaxed dom*. This scheme relaxes the requirements for a bridge candidate to be selected by

the greedy algorithm. Candidates can be selected so long as they have at most one 1-neighbor that has already been selected, as opposed to the dom scheme, where candidates may only be selected if they have no 1-neighbors that have already been selected. Further, we modify this relaxed dom scheme slightly to make the fifth scheme, *relax n*. This scheme operates the same as relaxed dom, but the number of selected bridge node 1-neighbors that a node can have and still be selected as part of the dominating set changes with n. Common values for n are 1, 2, and 3. Note that relax 0 is the same as dom, and relax 1 is the same as relaxed dom. The value of n is set at runtime based on the following equation:

$$n = \left\lfloor \frac{c}{16} \right\rfloor + 1 \tag{3.1}$$

Here, c represents the total number of bridge node candidates. Both the relaxed dom and relax n schemes introduce redundant nodes into the dominating set produced by the dom scheme. These redundant nodes can serve to reduce bottlenecking along the border of an exclusion zone.

The final scheme, *augmented connected dominating set*, or *ACDS*, is the new algorithm that we are proposing that is based on the essential connected dominating set algorithm.

3.2 Augmented Connected Dominating Set

The bridge node selection algorithm that we propose is based upon the idea of a *connected dominating set* (CDS). In graph theory, a CDS is a dominating set in which each member of the dominating set shares a link with at least one other member of the dominating set and the graph is connected [12]. For our purposes, when considering a connected dominating set as a network itself, it has the property of network connectivity defined in Section 3.1. The process of selecting the minimal CDS is an NP-hard problem, and the *essential connected dominating set* (ECDS) algorithm creates a CDS that serves as an approximation of the minimal CDS. In the ECDS algorithm, each node has an election heuristic. For our algorithm, we use its count of 1-neighbors.

We expand on the ECDS algorithm described in [13]. The algorithm presented in this citation uses ECDS for distributing routing information throughout a network, typically with nodes only knowing 1-hop and 2-hop information. In contrast to this setup, our implementation utilizes a centralized algorithm that operates with full network topology information. We implement a centralized version due to the assumed existence of a centralized controller, such as the SAS in

CBRS.

In the original ECDS algorithm, any node could serve as a dominant node. In our ACDS algorithm, however, only bridge node candidates are eligible to become dominant nodes. As such, many of a dominant node candidate's neighbors will not be eligible to be included in the CDS because they are not themselves bridge node candidates. We refer to our algorithm as *augmented* because it creates a CDS within the set of bridge nodes by choosing those bridge nodes that connect to the most other nodes - whether those other nodes are themselves bridge node candidates is irrelevant. In this way, the creation of the CDS within the network of only bridge nodes is augmented by the existence of non-bridge nodes. This distinction causes the algorithm to differ slightly, and the full algorithm is cataloged in the following list of steps:

1. Each bridge node candidate creates a list of its neighbors. This list of neighbors is different than the neighbor lists used in Lyui's algorithm because it will contain **any** node that is within range of this bridge node candidate, regardless of frequency considerations.

2. Consider a particular bridge node candidate T. Node T's nodal degree is the count of neighbors in its neighbor list. Node T compares its nodal degree to any of its neighbors that are also bridge node candidates. Node T remembers which of these nodes has the maximum nodal degree.

3. If Node T has the maximum nodal degree as calculated in step (2), Node T is chosen as a dominant node, and the algorithm concludes for this node. If Node T is tied with other nodes for the maximum nodal degree, Node T will consider itself as the node with maximum nodal degree.

4. If Node T fails step (3), it creates a queue and pushes the maximum nodal degree node into a queue Q. Node T also creates a list of all the nodes in the network in order to mark them as *visited*. This list of visited nodes includes **all** nodes, not just bridge node candidates. The node with the maximum nodal degree is marked as visited.

5. Pop Node X off of Queue Q. Mark all of Node X's 1-neighbors as visited. If any of Node X's 1-neighbors are themselves bridge node candidates that have a strictly higher nodal degree than Node T, push them into Q.

6. Repeat step (5) until Q is empty. If **any** of Node T's 1-neighbors are still *unvisited*, then Node T is selected as a dominant node. Otherwise, it is not selected.

After the algorithm completes, the created set of dominant bridge node candidates becomes the set of selected bridge nodes.

A notable difference between our ACDS algorithm and the original ECDS algorithm is how ties for maximal nodal degree are handled, as detailed in Step 4 above. In the original algorithm, ties for maximum nodal degree are broken by a node's ID number - nodes with a higher ID have priority and win ties. In our algorithm, the node of interest (Node T) has the priority to win ties. This method of tie-breaking is done intentionally to introduce redundant nodes into the set of selected bridge nodes that would otherwise go unselected by the original algorithm. As discussed in Section 3.1, the inclusion of redundant nodes can help to reduce bottlenecks.

Chapter 4

Results

4.1 Simulation Parameters and Overview

For the results presented in this paper, the values of the utilized parameters are given in Table 4.1. The parameters α, β, λ, N, N_0, and T_c are used in the calculations for SNIR during transmission simulations.

Parameter	*Value*
α	3.5
β	8.0
λ	0.125
N	8
N_0	4.0e-21
T_c	2.9e-7
Density	500 x 500 or 750 x 750
NUMTRIALS	1000000
NUMNODES	100
NUMCLIENTS	1000000
COMMRANGE	160
NEIGHBORRANGE	200
CLIENTRANGE	160

Table 4.1: Simulation Parameters

A network topology is created by randomly distributing 100 nodes within a square area with a uniform distribution on both x-position and y-position. Two sizes were chosen for the area: 500m by 500m and 750m by 750m. These sizes have respective densities of 1 node per 500 m^2 and 1 node per 750 m^2. For the 500 by 500 case, before any exclusion zones are added, nodes have an average of about 33.4 1-neighbors. For the 750 by 750 case, nodes have an average of about 16.9 1-neighbors. For Lyui's algorithm, nodes use a detectable range of 200 meters to identify neighbors. However, nodes are only considered to be communicable neighbors if they are within the communicable range of 160 meters of each other. For each of the two densities, 50 different network topologies are generated and simulated, and the results are averaged.

Within a particular network topology, four different scenarios can be employed to create *exclusion zones* within the network. The first scenario, called *halves*, partitions the network into two equal sections, with one of the sections acting as the exclusion zone. The second scenario, called *circular regions*, places three circular exclusion zones randomly within the network. These zones are placed with a uniform distribution and have radius 160, the same as the communication range of a node. The third scenario, called *stripe*, partitions the network into three equal sections across the x-coordinate, creating three equal vertical bars. In stripe, the middle bar acts an exclusion zone. The final strategy, called *inverted stripe*, partitions the network in exactly the same way as stripe; however, the two outer bars act as exclusion zones instead.

After exclusion zones are added, it is possible for a frequency 1 node to end up completely orphaned. This situation could happen in Figure 2.2, for example, if the bottom-right corner of the

network contained just one node instead of three. When this situation occurs, the orphaned node cannot be selected as a bridge node because it cannot reach both a node inside an exclusion zone and a node outside of all EZs. To handle this rare problem, we change orphaned frequency 1 nodes into frequency 2 nodes. Since frequency 1 nodes are technically able to use both frequencies (as they are not restricted by an exclusion zone), this switch is allowable.

Before the simulation of packets begins, the nodes are placed, and exclusion zones are added to the network. Afterwards, all possible bridge candidates are identified, and a subset of the candidates are marked as bridge nodes based on the selection scheme. Next, the network is split logically into two separate graphs based on frequency. Bridge nodes are included in both graphs. Then, Lyui's algorithm is run on each of the two graphs separately. As discussed in Section 2.2, each node in each graph will select a color number. Non-bridge nodes will end this phase with one color number. Bridge nodes are assigned two different color numbers, as they will participate in the scheduling algorithm of both graphs simultaneously. Each node also constructs its next-hop tables using Djikstra's algorithm. During Djikstra's algorithm, a node is considered to have links only with communicable neighbors that share its frequency availabilities. Furthermore, link weights in the network are either 1 (if two nodes share a valid link) or 0 (if two nodes do not share a valid link).

Nodes generate packets following a Bernoulli distribution. At each time step within the simulation, a node i will produce a packet with probability P_i. Each node is initialized with the same base packet generation rate $P_i = P$, and each simulation is run with various base packet generation rates. However, before the simulation of packet generation begins, nodes adjust their packet generation rates to account for the bridges being removed (since bridges do not generate packets). A Monte Carlo experiment is run that involves points being randomly selected within the rectangular area; these points represent the simulated *clients* that would be generating the traffic handled by the nodes. These clients are not a 1-to-1 representation of actual clients (i.e., we are not expecting to fit 1,000,000 clients into a 500 by 500 area), rather these simulated clients serve as an approximation of traffic density. Each client associates itself with its closest non-bridge node. After the Monte Carlo experiment concludes, a node i weights its generation rate P_i according to the percentage of clients that associated with it, as in (4.1). Bridge nodes do not handle clients and therefore do not generate any packets; instead, their clients associate with adjacent non-bridge nodes.

$$P_i = n * P * \left(\frac{c_i}{c}\right) \tag{4.1}$$

In this equation, P_i represents the new generation rate for node i, n represents the total number of nodes, P represents the base packet generation rate, c_i represents the number of clients that associated with node i, and c represents the total number of clients.

Because bridge nodes do not service clients, part of our investigations identified if the introduction of bridge nodes created *dead zones* in the network - areas in which clients can not be reached by the nodes in the backbone. Due to the denseness of the networks utilized in this study, we did not find a meaningful reduction of coverage due to the introduction of bridge nodes. (At worst, the percentage of clients that can reach a backbone node drops from 100% to around 99%.)

After the packet generation rates are set, the setup phase is completed, and the simulation begins. Nodes generate packets based on their modified generation rates. All packets have equal length and can be transmitted within a single time slot. Generated packets are assigned a destination node with a uniform distribution among all non-bridge nodes. Nodes within the two separate graphs are scheduled to transmit packets according to Lyui's algorithm. As discussed previously, nodes may only forward traffic to their communicable neighbors. All transmissions scheduled during each time step of the simulation happen simultaneously, potentially interfering with each other. The simulation runs for 1,000,000 slot durations. For each transmission on a particular frequency, the SNIR of the transmission is calculated, as discussed in Section 2.3. If the SNIR is found to be below the threshold β, then the packet is dropped.

Each node contains two queues. One queue contains packets that the node plans to transmit on frequency 1, while the second queue contains packets that the node plans to transmit on frequency 2. At any point in time, the total number of packets in both queues cannot exceed 10. A node will drop any packet that it receives if it currently has 10 packets in its queues. Notice that a non-bridge node will only use one of the two queues (e.g., frequency 2 nodes will only use the second queue, since a frequency 2 node can only plan to transmit packets on frequency 2). Bridge nodes, however, are free to use both queues. When a bridge node receives a packet, it checks the packet for its destination and uses its routing table to decide which of the two queues to use for the packet. If the next hop for the packet is a non-bridge frequency 1 node, the packet is placed into the first queue, since that packet will be transmitted on frequency 1. If the next hop for the packet is a frequency

2 node, the packet is placed into the second queue. In our simulation, if the packet's next hop is a bridge node, the bridge node chooses to transmit the packet on whichever frequency it received the packet on, regardless of scheduling availabilities. That is, if the packet was received on frequency 1, it will be transmitted to the next bridge node on frequency 1 as well. When a bridge node is selected by Lyui's transmission scheduling algorithm, it pops the front packet off of the applicable queue. (Recall from Section 3.1 that two separate instances of Lyui's algorithm are run simultaneously - one for each frequency being used in the network.)

4.2 Simulation Results

In this section, we use the following performance metrics to evaluate each selection scheme: *success rate*, *throughput*, and *average delay*. Success rate is defined as the total number of packets that reach their destinations divided by the total number of packets generated. Throughput is defined as the total number of packets that reach their destinations divided by the total number of time slots of the simulation. The average delay is defined as the average number of time slots required for a packet to complete the path from its source to its destination.

When examining the data gathered from the simulation, we first compare the results to the two simplest selection schemes: choosing the minimum number of bridge nodes (min) and choosing the maximum number of bridge nodes (max). We also compare the results of the more complicated selection schemes to determine which algorithm is the most robust.

We begin by examining the 500 by 500 network density. The results of the halves scenario are shown in Figures 4.1 - 4.3. From the throughput graph, our algorithm (ACDS) out-performs all of the other selection schemes. The min scheme displays the worst performance out of all of the schemes, and the dominating set scheme (dom) performs worse than simply choosing every possible bridge node (max). The relaxed dominating set (relaxed dom) and relax n schemes contend with our algorithm. In Figures 4.4-4.6, the circular regions results are shown. Again, ACDS demonstrates a higher throughput than the other algorithms. Dom fails to surpass max, and relaxed dom and relax n offer little improvement over max. Figures 4.7-4.9 display the stripe results. Here, our algorithm is the only one to offer any improvement over the max scheme. In the inverted stripe results shown in Figures 4.10-4.12, relaxed dom, relax n, and our algorithm all offer considerable improvement over the max scheme; however, our algorithm provides the most throughput in the range of 90%-100%

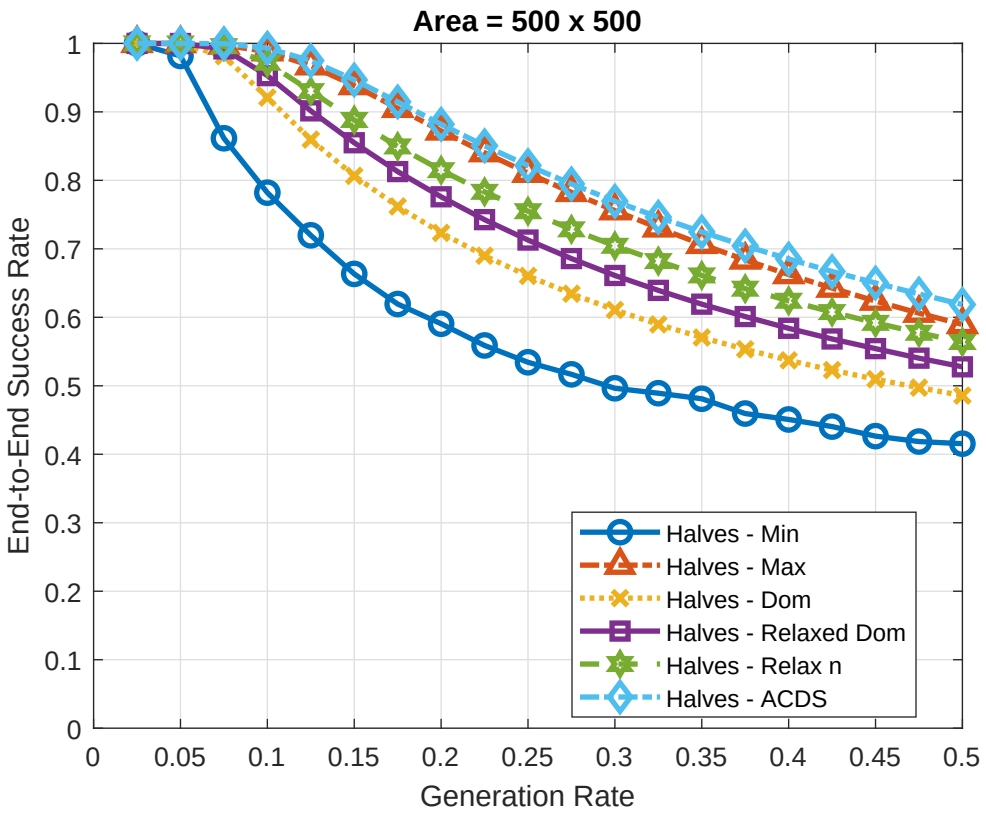

Figure 4.1: Knockout Style: Halves - Success Rate

success rate. In all four of these knockout cases, our algorithm provides the most throughput in the range of 90%-100% success rate.

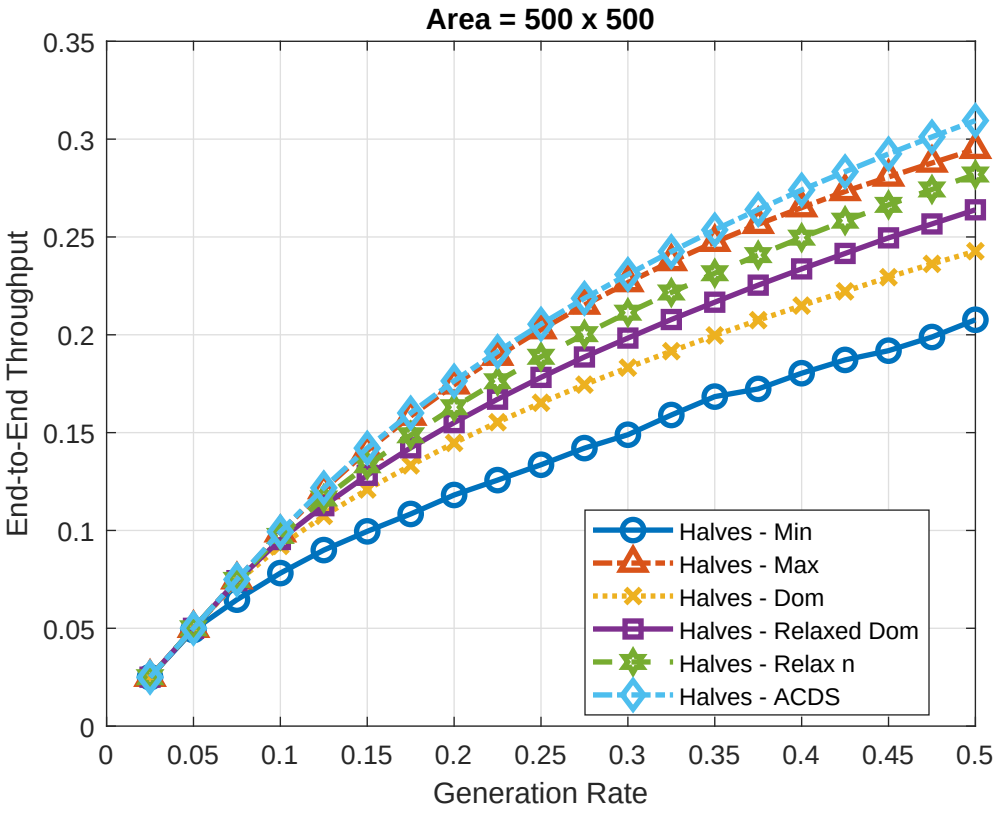

Figure 4.2: Knockout Style: Halves - Throughput

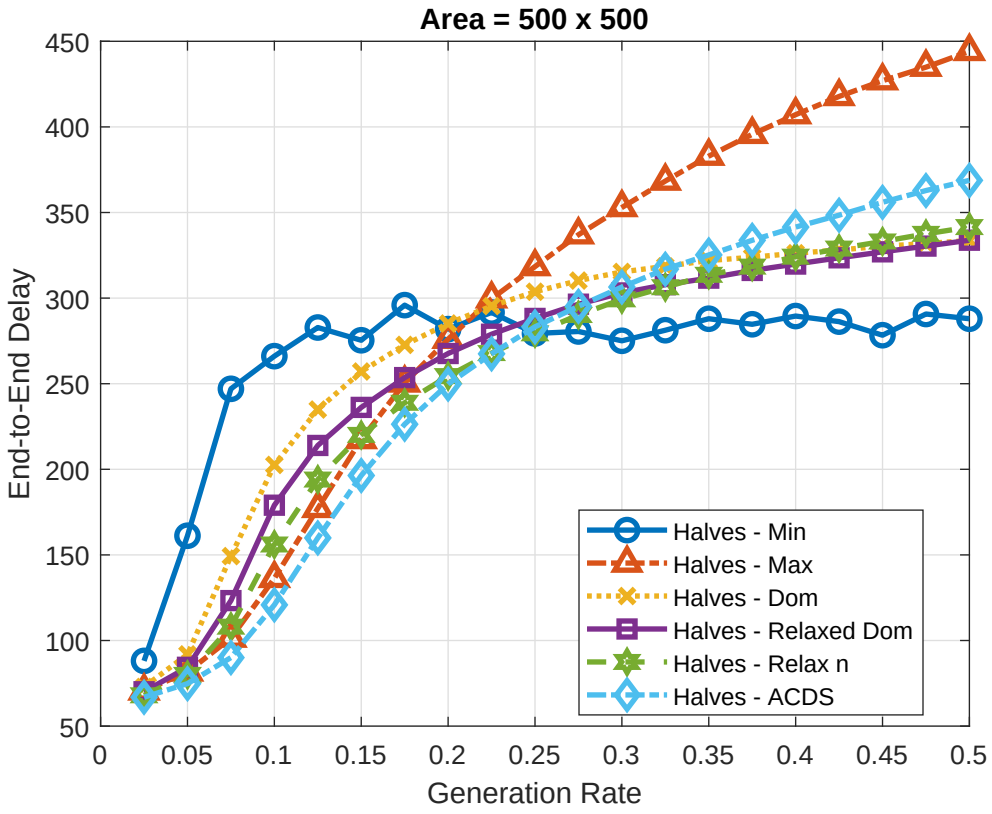

Figure 4.3: Knockout Style: Halves - Delay

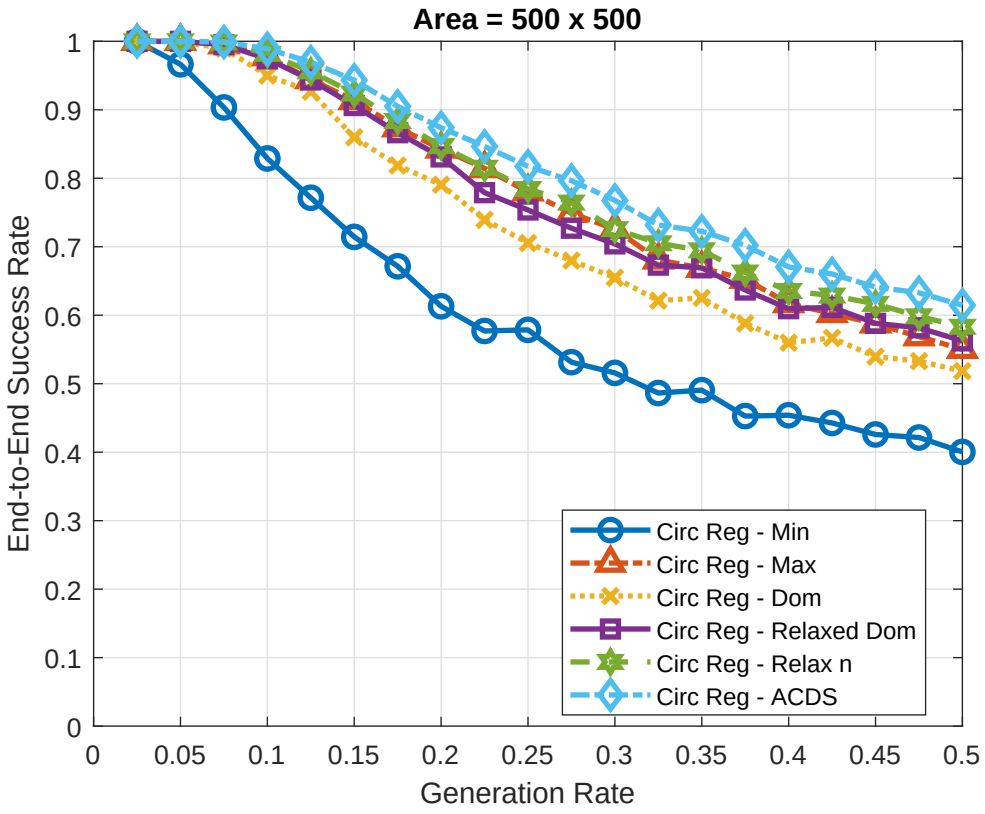

Figure 4.4: Knockout Style: Circular Regions - Success Rate

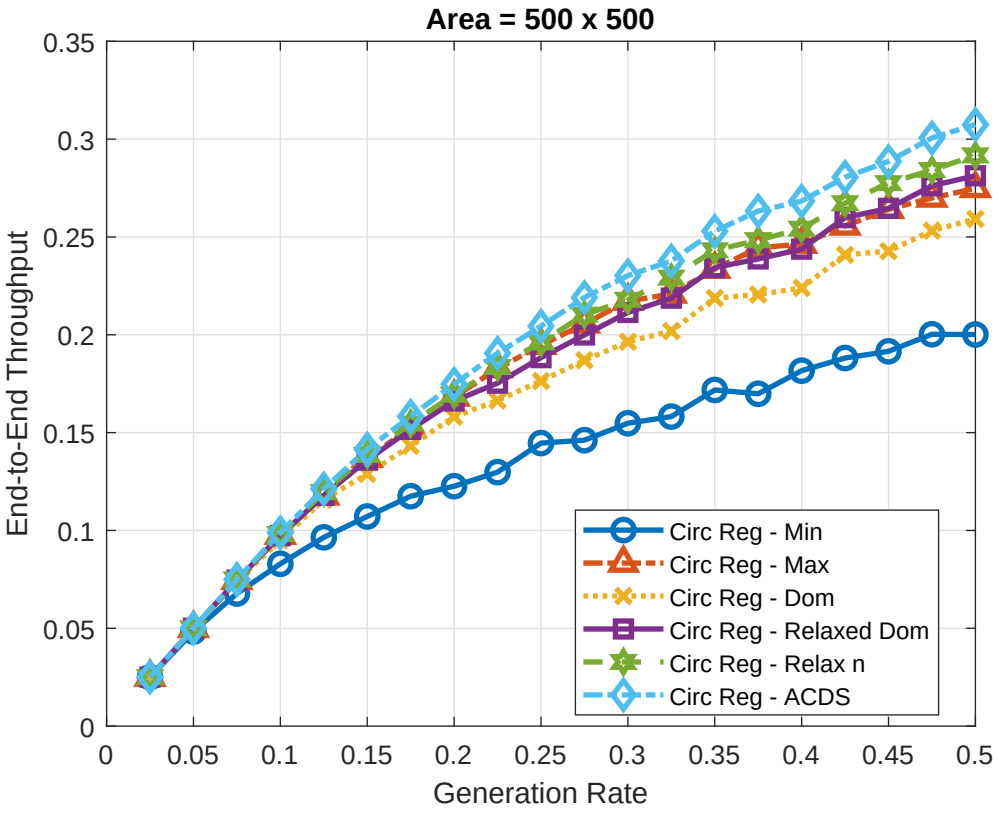

Figure 4.5: Knockout Style: Circular Regions - Throughput

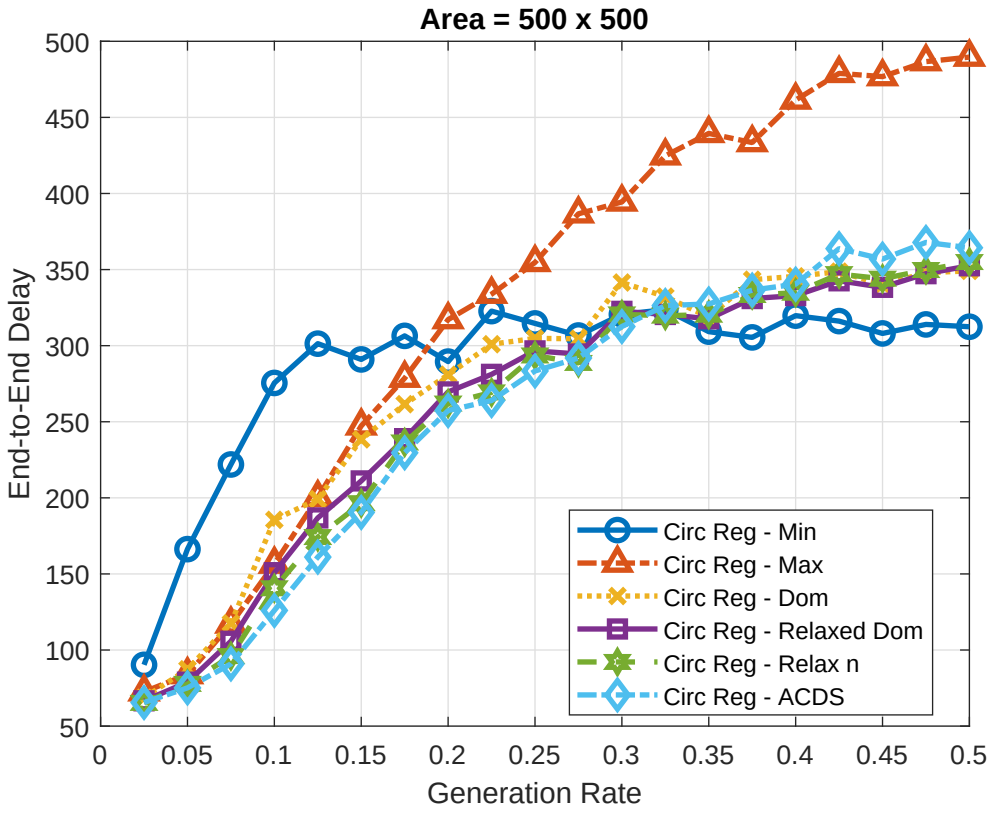

Figure 4.6: Knockout Style: Circular Regions - Delay

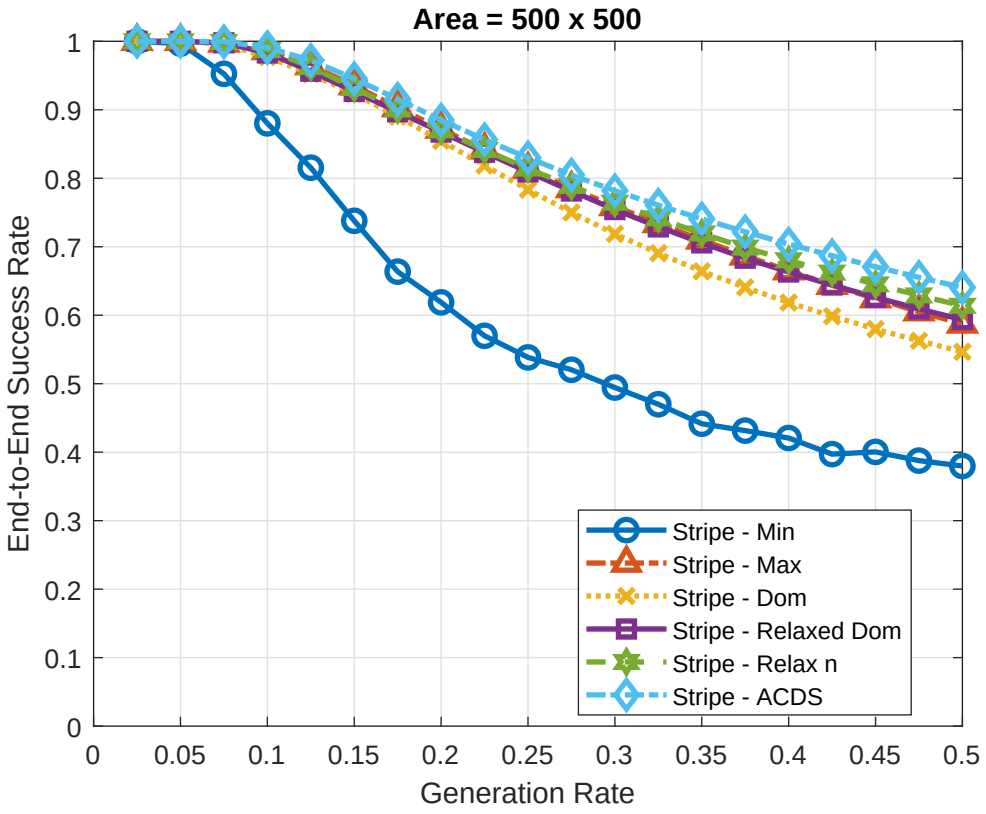

Figure 4.7: Knockout Style: Stripe - Success Rate

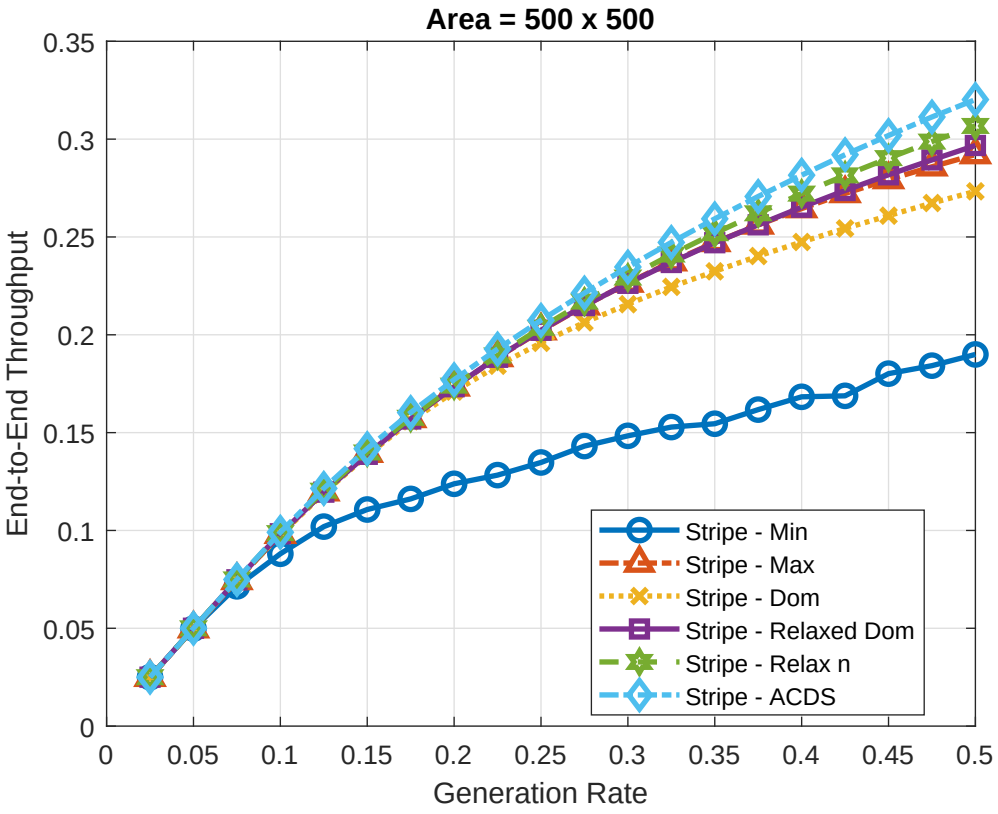

Figure 4.8: Knockout Style: Stripe - Throughput

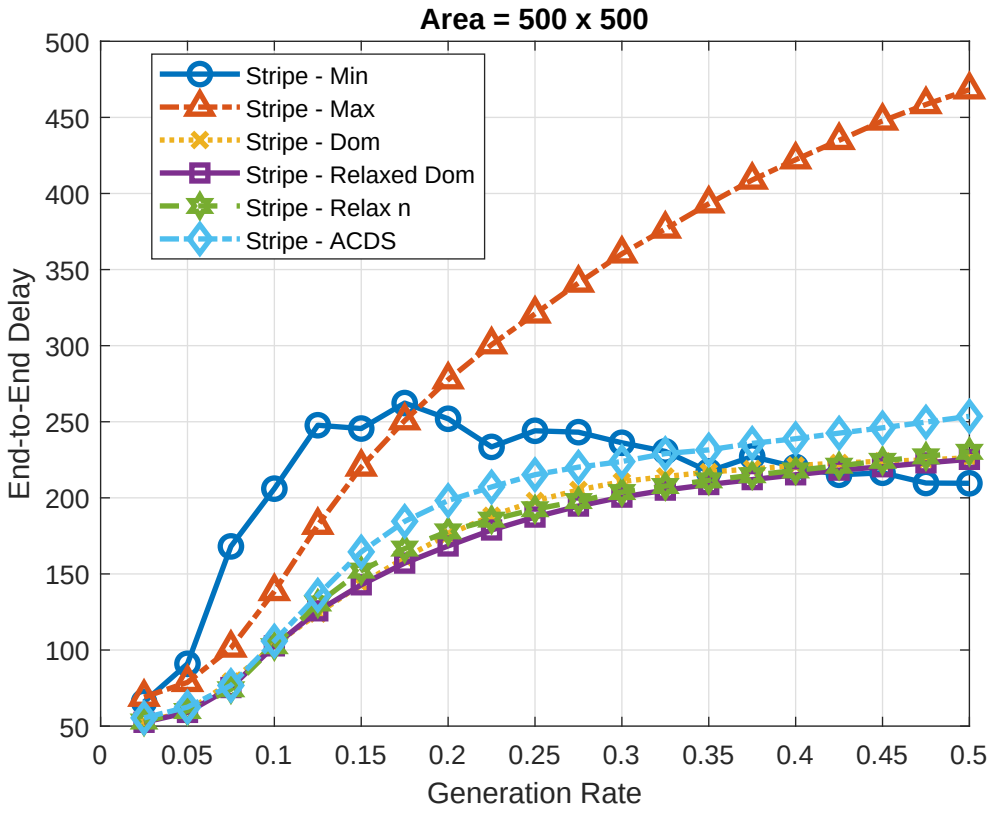

Figure 4.9: Knockout Style: Stripe - Delay

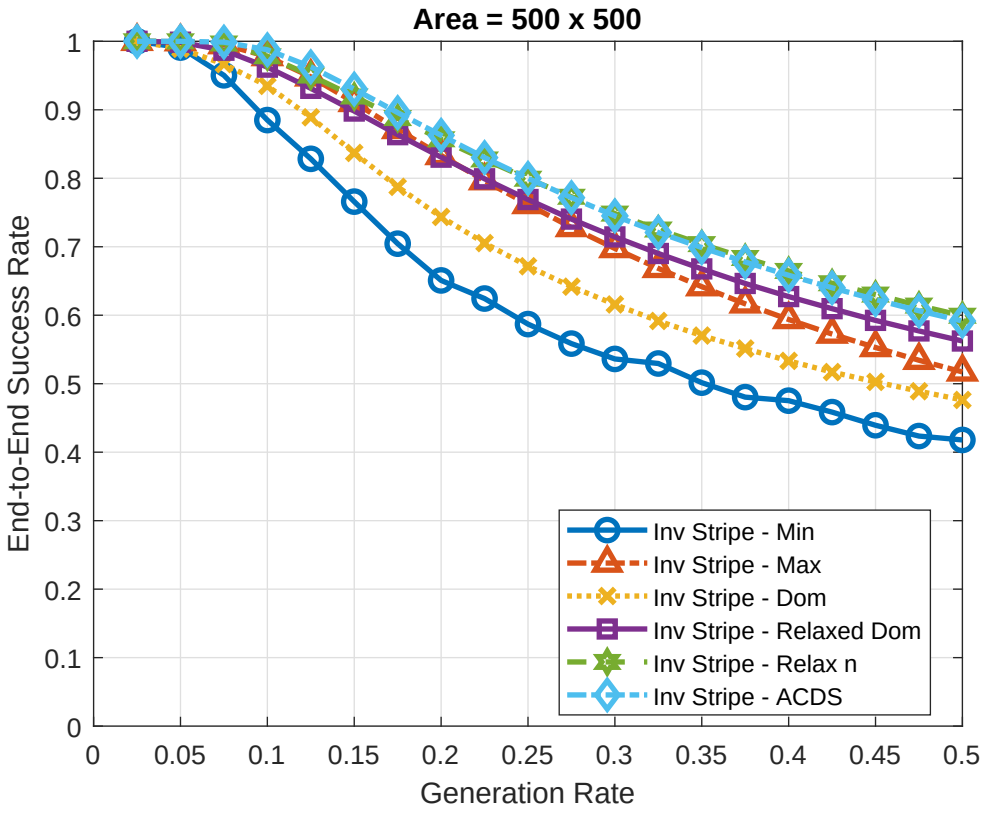

Figure 4.10: Knockout Style: Inverted Stripe - Success Rate

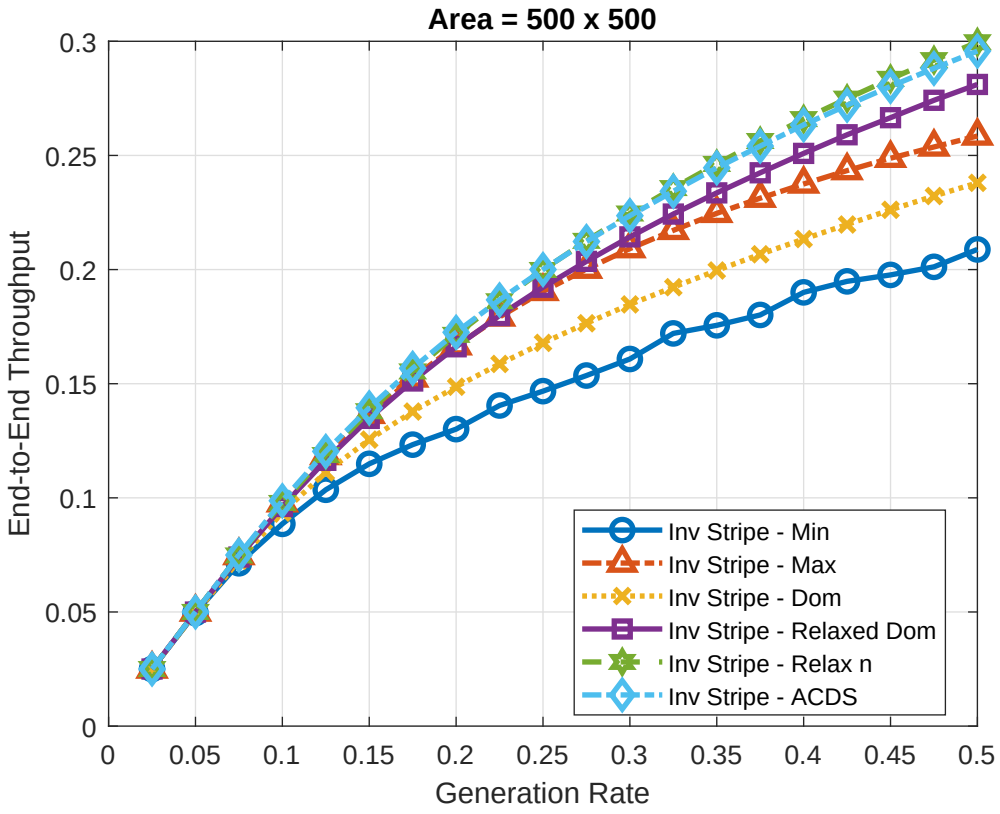

Figure 4.11: Knockout Style: Inverted Stripe - Throughput

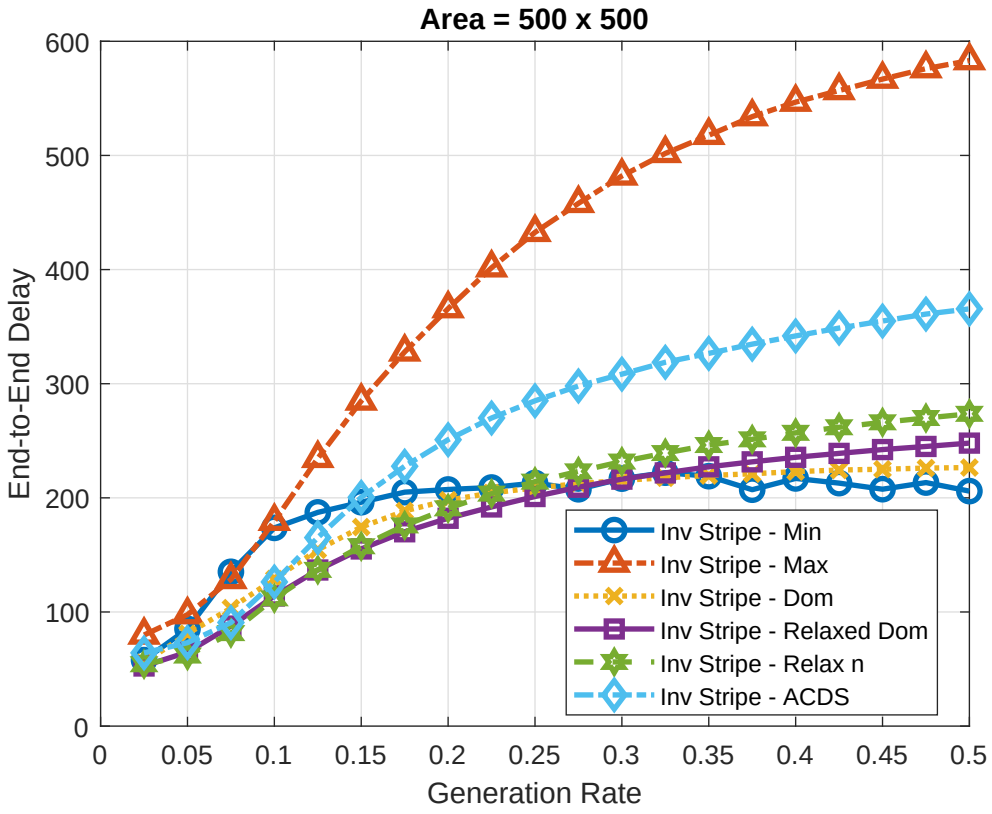

Figure 4.12: Knockout Style: Inverted Stripe - Delay

Scheme	*Selected*	*Candidates*	*Percentage*
Min	1	27.3	3.66
Max	27.3	27.3	100.00
Dom	3	27.3	10.99
Relaxed Dom	5.5	27.3	20.15
Relax n	8.8	27.3	32.23
ACDS	13.9	27.3	50.92

Table 4.2: Percentage of Selected Bridge Candidates - 500 by 500 - Halves

Next, consider the investigations that utilize the 750 by 750 density. Figures 4.13-4.15 display the halves results for this density. Similar to the 500 by 500 case, the min scheme performs poorly in all knockout scenarios. For the halves scenario, max, relaxed dom, relax n, and ACDS all produce similar results, with ACDS having a marginal advantage. Figures 4.16-4.18, the circular regions results, demonstrate similar output, with dom also contending. Figures 4.19-4.21 display the stripe results, in which relax dom, relax n, and ACDS all demonstrate marginal improvements over max. Figures 4.19-4.21 display the inverted stripe results. Here, relax n and ACDS show higher throughput than relaxed dom and max.

In all cases, ACDS has the highest success rate and throughput out of all of the selection schemes. When selecting bridge nodes to forward traffic across the edge of the exclusion zone, it is first important to select enough nodes to prevent bottlenecking along the border of the zone. As is obvious from the min results, choosing a minimal number of bridges results in unacceptably low performance. Choosing every candidate to be a bridge node (as in max) will prevent these bottlenecks, but this scheme may end up choosing too many redundant bridge nodes. It is clear from the results that creating a dominating set among the bridge nodes (dom) does not choose a high enough proportion of the bridge node candidates to prevent bottlenecking; dom performs worse than max in every case. Relaxed dom, relax n, and ACDS choose a larger percentage of bridge candidates than the dom scheme does. Table 4.1 lists the number of candidates selected by each scheme in the 500 by 500 halves scenario. The relaxed dom and relax n selection schemes choose candidates in a pure greedy sense - like the max case, they may end up choosing redundant bridges.

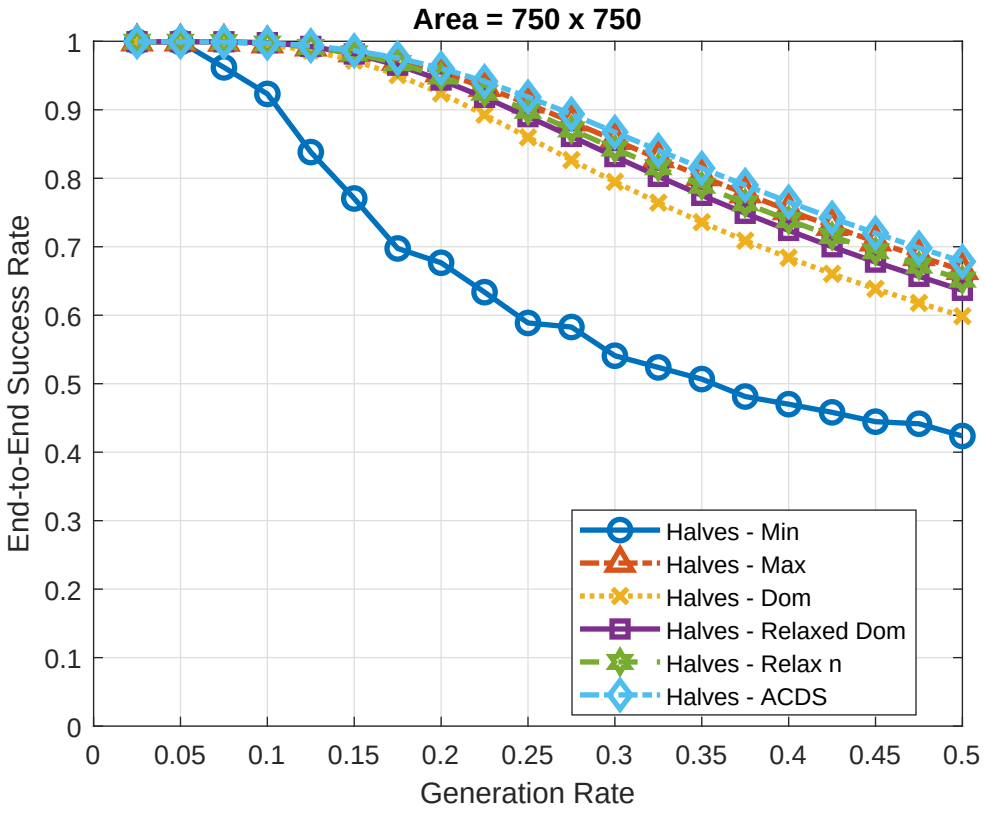

Figure 4.13: Knockout Style: Halves - Success Rate

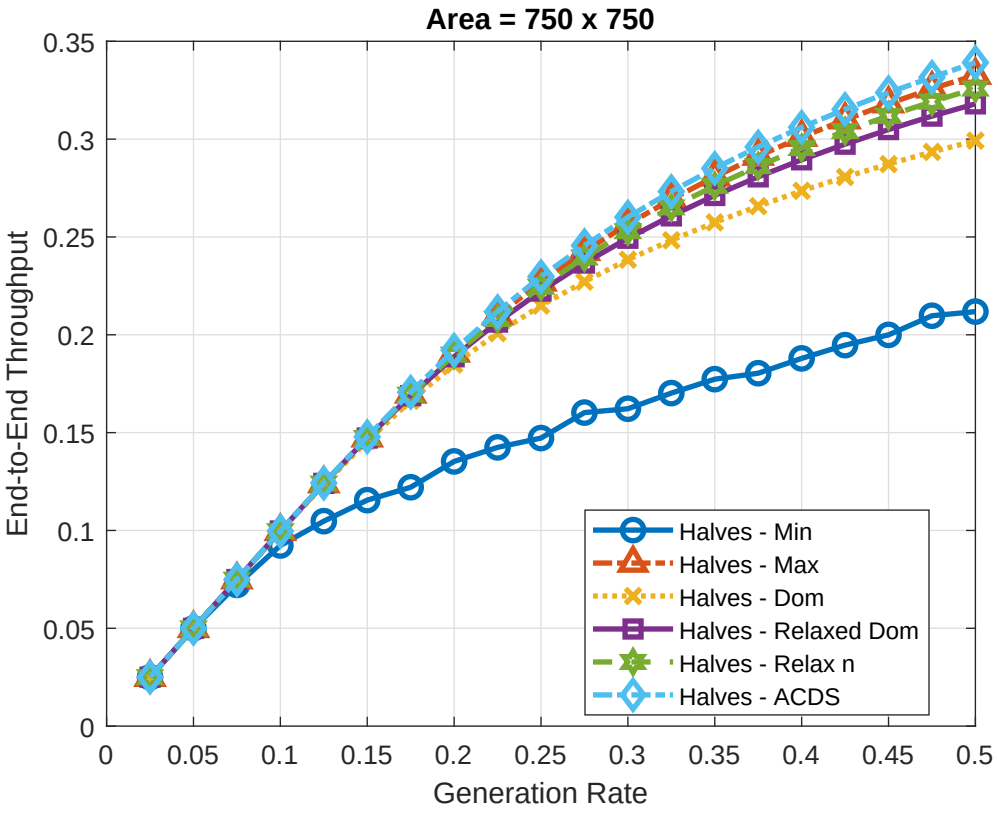

Figure 4.14: Knockout Style: Halves - Throughput

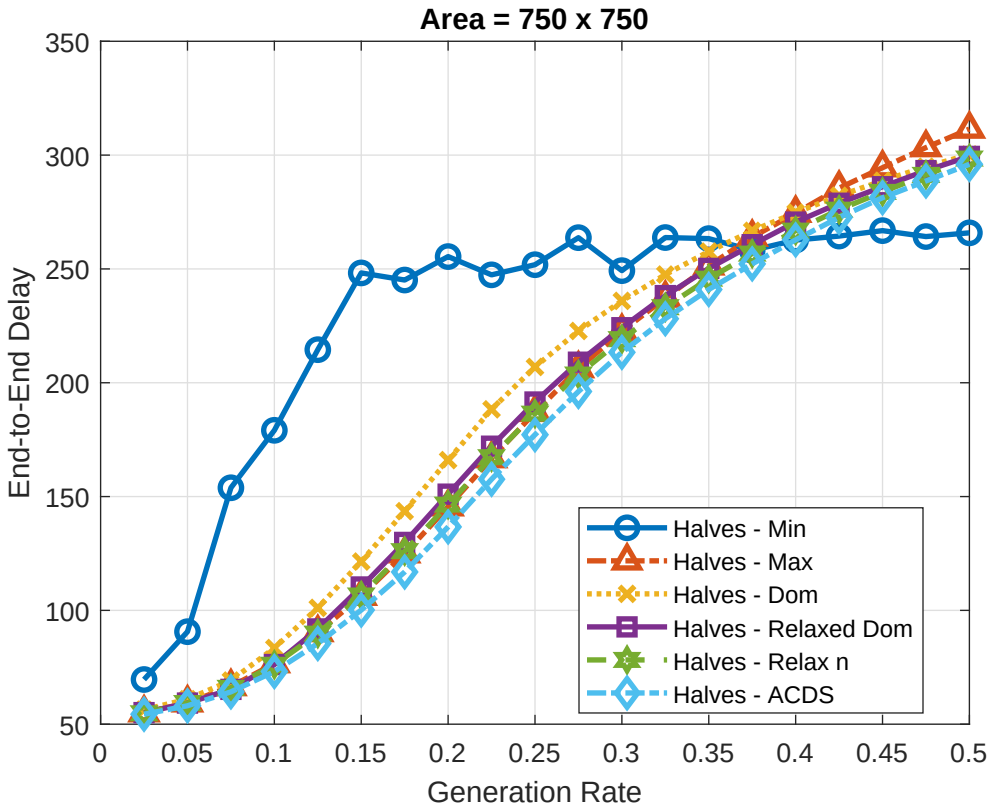

Figure 4.15: Knockout Style: Halves - Delay

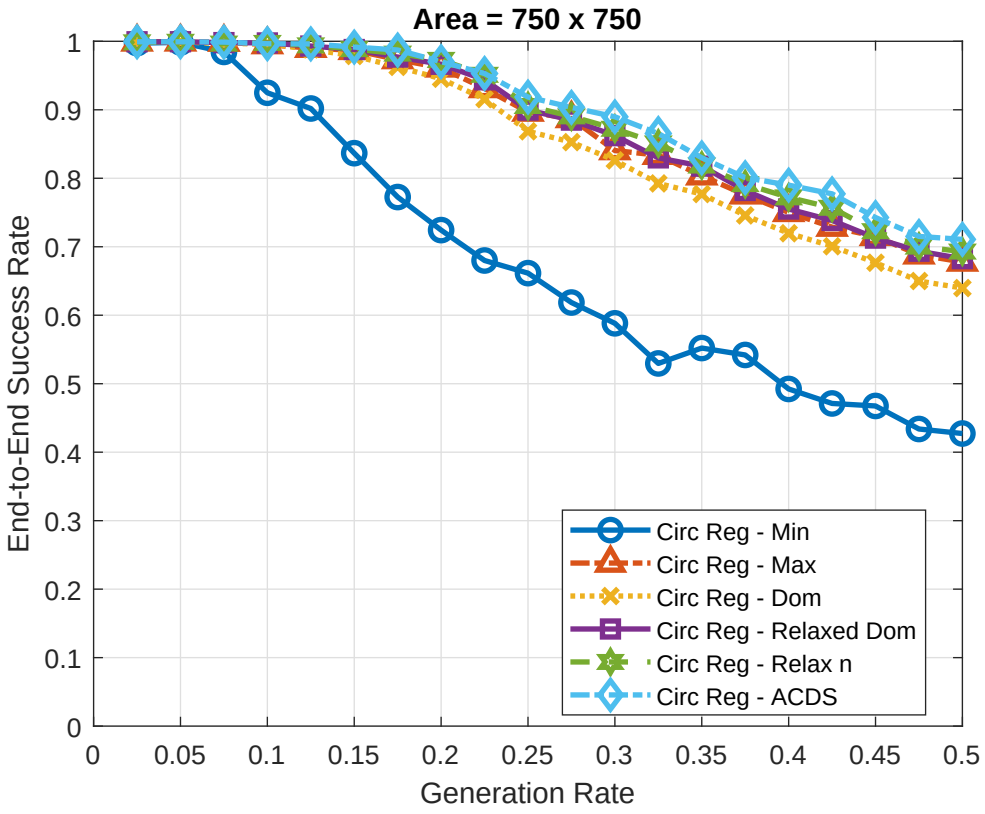

Figure 4.16: Knockout Style: Circular Regions - Success Rate

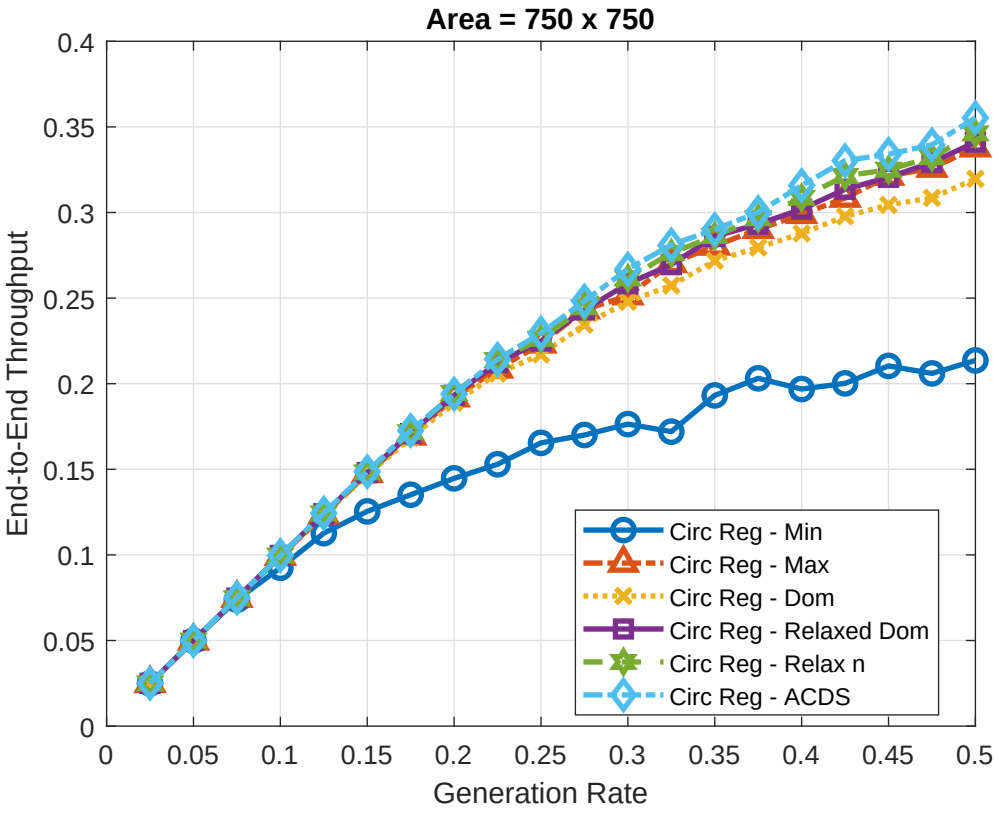

Figure 4.17: Knockout Style: Circular Regions - Throughput

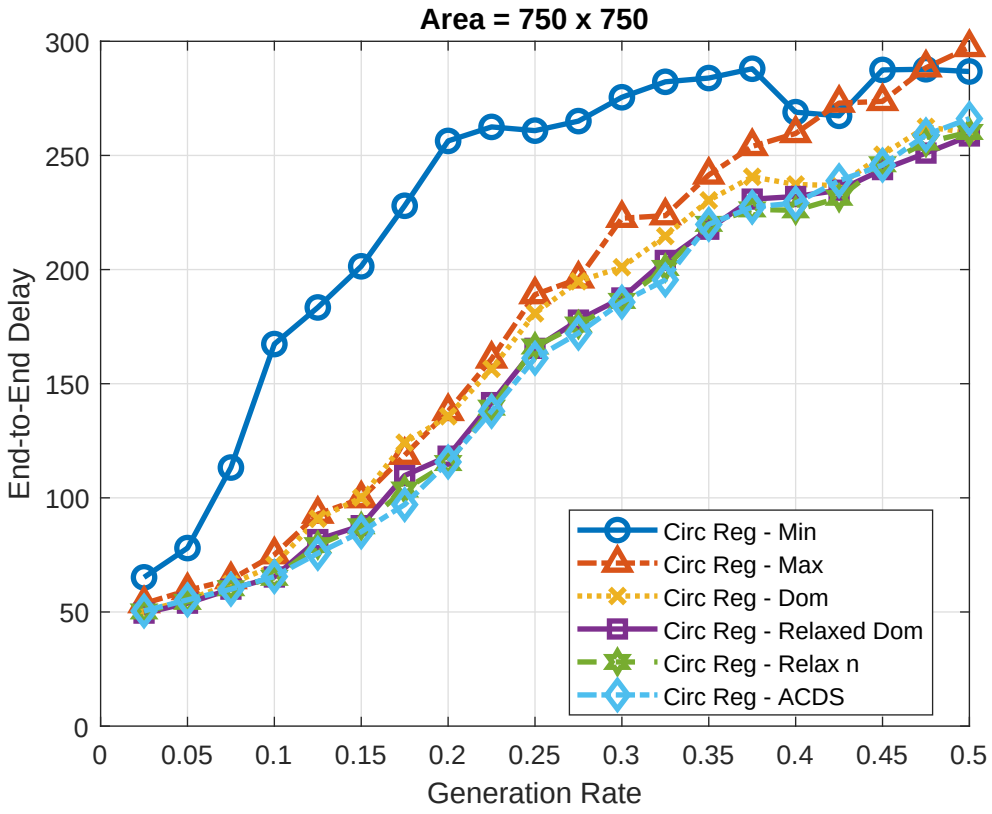

Figure 4.18: Knockout Style: Circular Regions - Delay

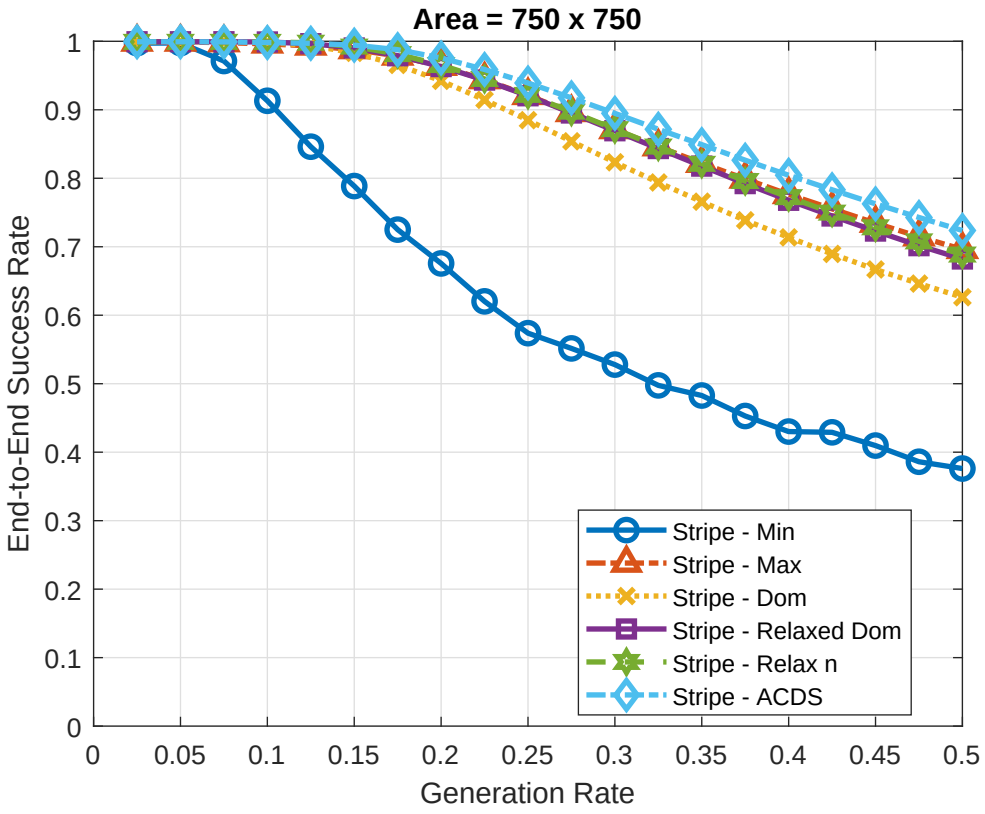

Figure 4.19: Knockout Style: Stripe - Success Rate

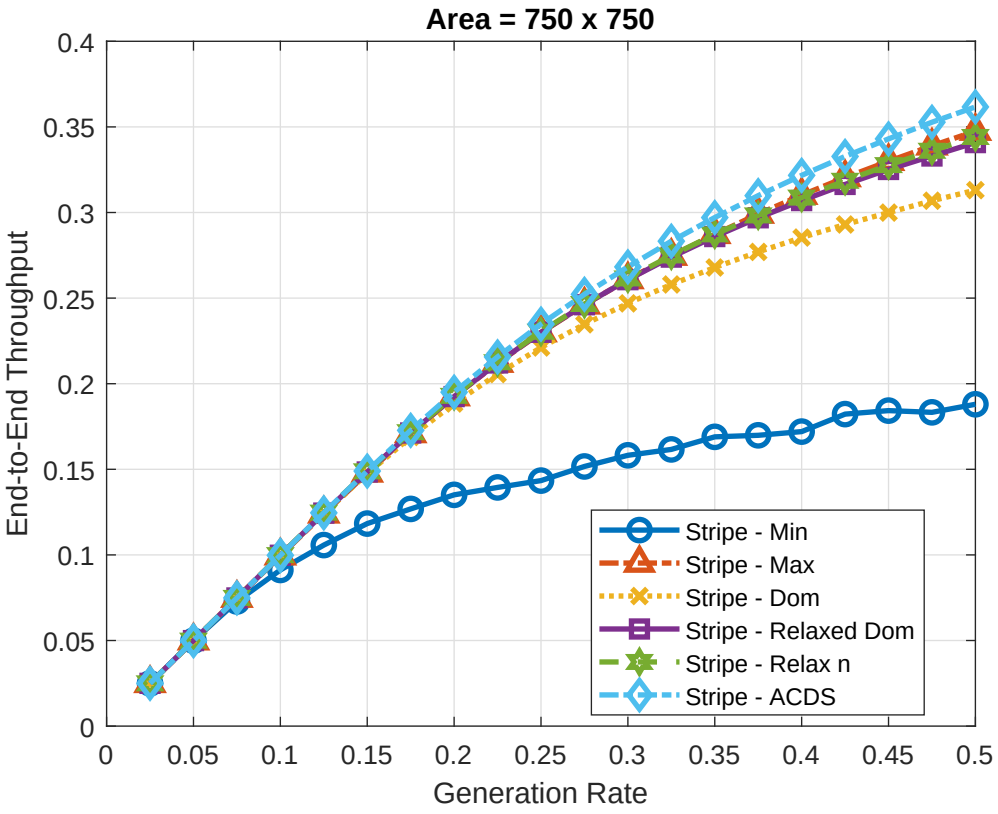

Figure 4.20: Knockout Style: Stripe - Throughput

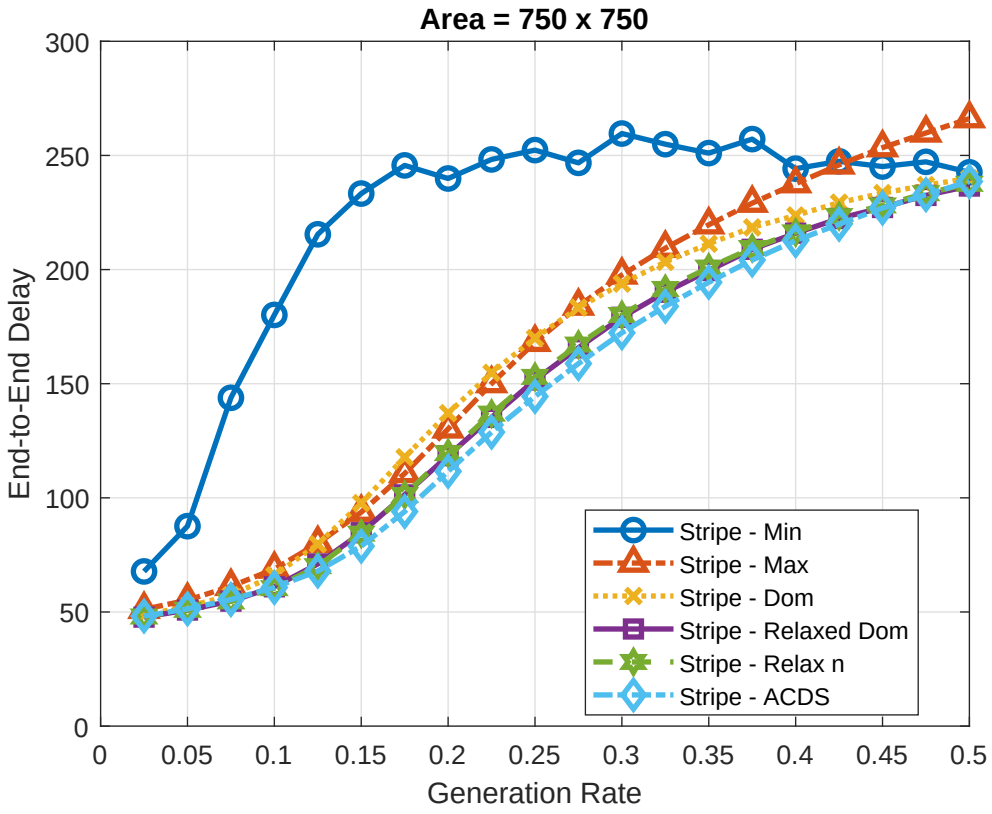

Figure 4.21: Knockout Style: Stripe - Delay

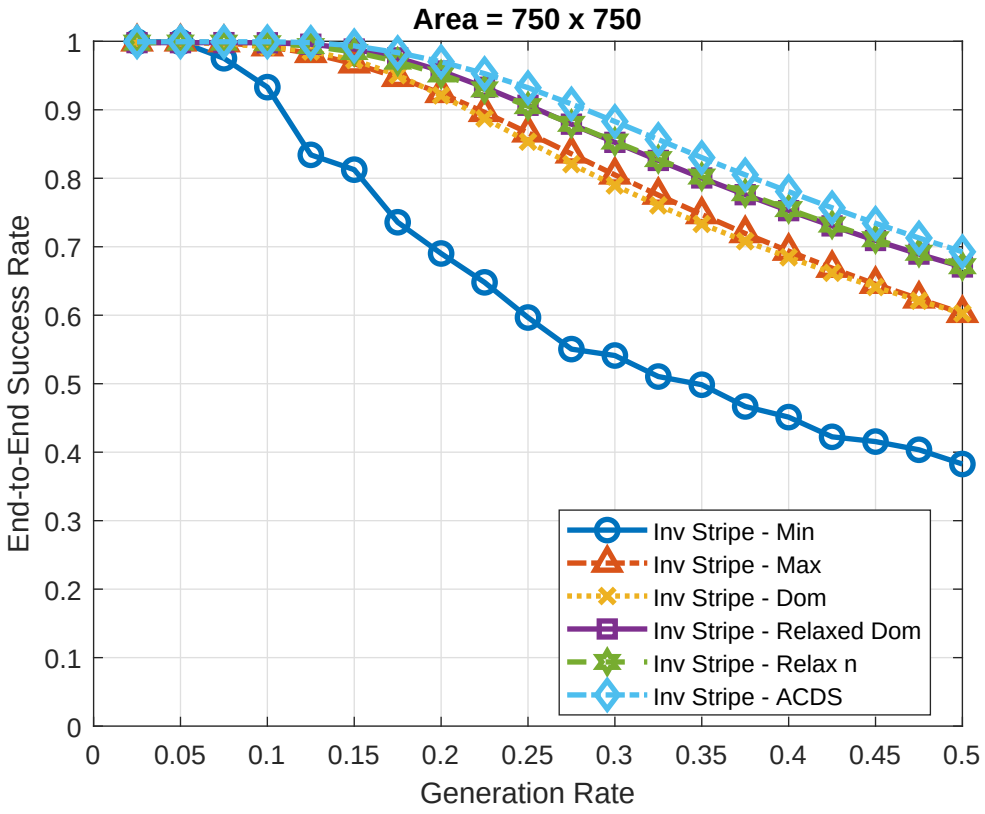

Figure 4.22: Knockout Style: Inverted Stripe - Success Rate

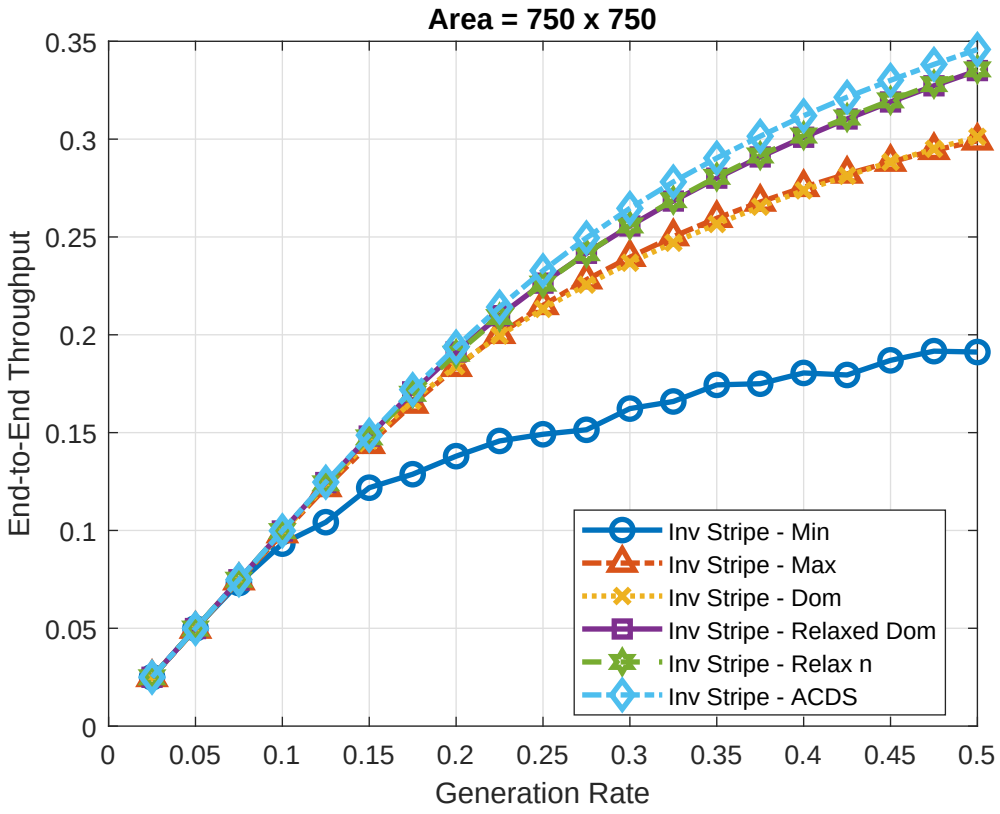

Figure 4.23: Knockout Style: Inverted Stripe - Throughput

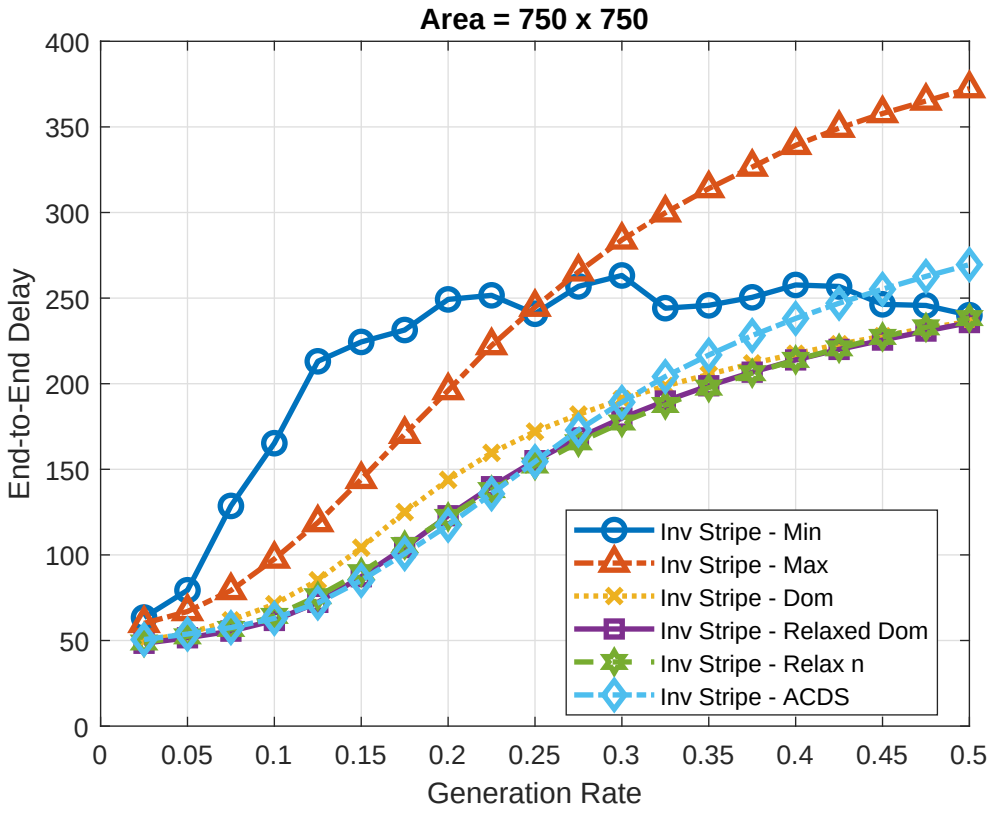

Figure 4.24: Knockout Style: Inverted Stripe - Delay

Chapter 5

Conclusions and Discussion

Although the Citizens Broadband Radio Service establishes a framework in which underutilized spectrum can be shared with new users, it also presents the problem of protecting incumbent users from interference created by these new users. The introduction of the spectrum access system allows arbitration between different tiers of users. The enforcement of access priorities by the SAS creates exclusion zones in the network for lower priority users, as these users must back out of frequency bands that higher priority incumbent users require. The introduction of exclusion zones into a network weakens network performance for these users by threatening network connectivity and by creating bottlenecks.

We examine a strategy to restore network performance by converting some nodes into bridge nodes - dedicated traffic-forwarding nodes that specialize in transferring traffic into and out of exclusion zones. We analyze the effects of these bridge nodes in a mesh network implementation of CBRS in which nodes operate on exactly two frequencies and with exactly two transceivers. Transmissions are scheduled using Lyui's transmission scheduling algorithm, and a signal-to-noise plus interference ratio is calculated to determine whether transmissions are successful.

Our main contribution is the introduction of the augmented connected dominating set algorithm for selecting bridge nodes from a list of bridge node candidates. We compare our algorithm to other, simpler algorithms for selecting bridge nodes. The ACDS algorithm provides consistently higher throughput than all other selection schemes, regardless of the shape or location of introduced exclusion zones. Furthermore, algorithms that contend with the ACDS algorithm display weak performance in at least one exclusion zone scenario.

Future investigations into this research problem would examine more complicated simulation designs.

Currently, the simulation weights each node's packet generation rate based on how many clients they are likely to serve (see equation (4.1)). However, packet destinations are not weighted in the same way. (Packet destinations are selected from the set of all non-bridge nodes with a uniform distribution.) The results may change if this destination weighting is implemented similarly to the generation rate weighting.

A different extension would be to allow for nodes to utilize more than just two frequencies - forcing the selection algorithm to allocate bridge nodes to pairwise frequency borders.

Another extension would be to allow the nodes that are operating on frequency 1 to utilize frequency 2 when possible. Currently, the nodes on frequency 1 are capable of accessing frequency 2, but only use both frequencies if they are selected to be a bridge node. Since frequency 2 is available to these nodes, throughput will likely increase if these nodes can be scheduled to use frequency 2 alongside frequency 1.

More straightforward extensions include experimenting with new frequency knockout styles and shapes and testing new bridge node selection schemes.

Appendices

Appendix A SNIR Model Considerations

For this section, we consider the SNIR of a transmission sent from node i to node j. As described in Section 2.3, this SNIR is calculated as:

$$SNIR = \frac{P_r(i)NT_c}{N_0 + \sum_{\forall k \neq i} P_r(k)T_c} \geq \beta \tag{1}$$

Furthermore, the power received at node j from a node i is calculated as:

$$P_r(i) = P_t\Big(\frac{\lambda}{4\pi d_{i,j}}\Big)^{\alpha} \tag{2}$$

An important implication of (2) is that, for certain values of $d_{i,j}$, the received power $P_r(i)$ will be higher than the transmission power P_t. We term the smallest value of $d_{i,j}$ such that $P_r(i) < P_t$ as d_{min}. For our system parameters (Table 4.1), d_{min} is calculated to be 9.95 mm. Within our simulation, the probability that two nodes are within 9.95 mm of each other is less than 0.001%. We can approximate this probability as:

$$P = 1 - \Big(1 - \Big(N * \frac{\pi * (d_{min})^2}{X * Y}\Big)\Big)^N \tag{3}$$

In equation (3), P represents the probability we are calculating, N represents the total number of nodes, and X and Y represent the dimensions of the rectangular area. Note that this equation is an overestimate and thus provides an upper bound. Because this probability is sufficiently low, the path loss equation models far field behavior.

As discussed in Section 2.4, we can set the transmission power in order to force the detectable range to be a suitable value. To do so, we calculate the required transmission power based on system parameters. In following calculations, the variable R refers to the value of the detectable range that we have chosen for our system. We set the transmission power by the following calculation:

$$P_t = \Big(\frac{4\pi R}{\lambda}\Big)^{\alpha} \frac{\beta N_0}{T_c N} \tag{4}$$

Using our set of system parameters (Table 4.1), P_t is set to be 0.011 W.

We can substitute (4) into (2) and (2) into (1):

$$SNIR = \frac{\left(\frac{4\pi R}{\lambda}\right)^{\alpha} \frac{\beta N_0}{T_c N} \left(\frac{\lambda}{4\pi d_{i,j}}\right)^{\alpha} N T_c}{N_0 + \sum_{\forall k \neq i} \left(\frac{4\pi R}{\lambda}\right)^{\alpha} \frac{\beta N_0}{T_c N} \left(\frac{\lambda}{4\pi d_{k,j}}\right)^{\alpha} T_c} \geq \beta \tag{5}$$

By reducing (5), we arrive at a final formula for SNIR in our model that is simpler to compute than (1). Notice that (6) does not depend on λ, T_c, or N_0. These variables cancel out of the equation due to the transmission power's dependence on these variables (as in (4)).

$$SNIR = \frac{\left(\frac{R}{d_{i,j}}\right)^{\alpha}}{1 + \sum_{\forall k \neq i} \left(\frac{R}{d_{k,j}}\right)^{\alpha} \frac{\beta}{N}} \geq 1 \tag{6}$$

Bibliography

[1] FCC, "FCC 15-47: in the matter of amendment of the commission's rules with regard to commercial operations in the 3550-3650 MHz Band; GN Docket No. 12-354," pp. 1–187, 2015.

[2] S. Bhattarai, J.-M. Park, B. Gao, K. Bian, and W. Lehr, "An overview of dynamic spectrum sharing: ongoing initiatives, challenges, and a roadmap for future research," *IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 2, pp. 110–128, June 2016.

[3] A. Lackpour, A. Rosenwinkel, J. R. Guerci, A. Mody, and D. Ryan, "Design and analysis of an information exchange-based radar/communications spectrum sharing system (RCS3)," *IEEE Radar Conference*, 2016.

[4] M. M. Sohul, M. Yao, T. Yang, and J. H. Reed, "Spectrum access system for the citizen broadband radio service," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 18–25, July 2015.

[5] S. Yrjölä, M. Matinmikko, M. Mustonen, and P. Ahokangas, "Analysis of dynamic capabilities for spectrum sharing in the citizens broadband radio service," *Analog Integrated Circuits and Signal Processing*, 2017.

[6] W. C. Headley, V. G. Chavali, and C. R. C. M. Da Silva, "Exploiting radio correlation and reliability information in collaborative spectrum sensing," *IEEE Communications Letters*, vol. 15, no. 8, pp. 825–827, 2011.

[7] S. Joshi, K.-B.-S. Manosha, M. Jokinen, and T. Hänninen, "ESC sensor nodes placement and location for moving incumbent protection in CBRS," *Wireless Innovation Forum Conference on Wireless Communication Technology and Software Defined Radio*, March 2016.

[8] D. G. Kuester, R. T. Jacobs, Y. Ma, and J. B. Coder, "Demultiplexing spectrum-sharing field sources with distributed field probes," *IEEE International Symposium on Electromagnetic Compatibility*, pp. 336–341, Sept 2016.

[9] W.-P. Lyui, "Design of a new operational structure for mobile radio networks," Ph.D. dissertation, Clemson University, Clemson, SC, August 1991.

[10] J. L. Hammond and H. B. Russell, "Properties of a transmission assignment algorithm for multiple-hop packet radio networks," *IEEE Trans. on Wireless Communications*, vol. 3, no. 4, pp. 1048–1052, July 2004.

[11] B. Wolf and H. B. Russell, "Immediate neighbor scheduling (INS): An adaptive protocol for mobile ad hoc networks using direct-sequence spread-spectrum modulation," *Ad Hoc Networks*, vol. 9, no. 3, pp. 453–467, May 2011.

[12] S. T. Hedetniemi and R. C. Laskar, "Bibliography on domination in graphs and some basic definitions of domination parameters," *Discrete Mathematics*, vol. 86, pp. 257–277, 1990.

[13] R. Ogier and P. Spagnolo, "Mobile ad hoc network (MANET) extension of OSPF using connected dominating set (CDS) flooding," Internet Requests for Comments, RFC Editor, RFC 5614, August 2009.