IMPROVING PERFORMANCE THROUGH TOPOLOGY MANAGEMENT AND WIRELESS SCHEDULING IN MILITARY MULTI-HOP RADIO NETWORKS

by

ZACHARY S. BUNTING

BACHELOR OF SCIENCE, U.S. NAVAL ACADEMY (2011)

Submitted to the Sloan School of Management in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2013

This work is sponsored by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract # FA8721-05-C-0002. Opinions, interpretations, recommendations and conclusions are those of the author and are not necessarily endorsed by the United States Government.

Signature of Author .....
Sloan School of Management
May 17, 2013

Certified by .....
Eytan Modiano
Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Patrick Jaillet
Dugald C. Jackson Professor, Department of Electrical Engineering and Computer Science
Co-director, Operations Research Center

Improving Performance through Topology Management and Wireless Scheduling in Military Multi-hop Radio Networks

by

Zachary S. Bunting

Submitted to the Sloan School of Management
on May 10, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Operations Research

Abstract

We investigate two distinct problems in military radio networking. In the first problem, we study a mobile airborne multi-hop wireless network. The mobility of the nodes leads to dynamic link capacities requiring changes to the topology by adding and removing links. Changes are intended to minimize maximum link load. Mixed integer linear programming is used to periodically find topological modifications resulting in optimal performance. To reduce computation and the rate of changes to the topology, we design and employ heuristic algorithms. We present several such algorithms of differing levels of complexity, and model performance using each. A comparison of the results of each method is given.

In the second problem, we study a ground multi-hop wireless network. Scalability is an issue for such ground tactical radio networks, as increasing numbers of nodes and flows compete for the capacity of each link. The introduction of a relay node allows additional routes for traffic flows. Greater benefit is achieved by fixing the relay node at a higher elevation to allow it to broadcast to all other nodes simultaneously, thereby reducing the number of hops packets must travel. We use a combination of linear programming (LP) and novel bounds on the achievable network performance to investigate the benefits of such a relay node. We show that a relay node provides moderate improvement under an all-to-all unicast traffic model and more substantial improvement for broadcast traffic patterns.

Thesis Supervisor: Eytan Modiano
Title: Professor

Acknowledgments

I would like to first thank my advisor, Professor Eytan Modiano, and acknowledge the guidance, encouragement, knowledge, support, patience, assistance, and diligence you demonstrated over the last two years. Your efforts helped me to harness the raw skills and tools I brought into this experience and accomplish what I would not have otherwise been able to do.

I would also like to thank members of the Lincoln Laboratory staff for providing an environment in which I have been able to work side by side with such a talented research team. Specifically, I wish to thank John Kuconis for lining me up with Group 65; Matt Kercher for constantly looking to set me up for success and patiently helping me adjust to the challenges of the program; Aradhana Narula-Tam for intense help in identifying a problem and working through its solution, as well as helping me expand my vision and making requisite sacrifices to help me achieve my goals; and other individuals that helped me along the way, including Evelyn Bennett, Devanshu Mehta, Bow-Nan Cheng, Russ Hamilton, Jeff Pondick, Ed Kuczynski, and Larry Bressler.

I would like to thank my fellow ORC students for the technical help and moral support they offered me. In particular, I wish to thank my classmates, Dave Culver, John Kessler, Emily Frost, Eric Robinson, and Stephen Relyea, as well as Ben Letham and John Silberholz.

I would also like to thank the U.S. Navy, for affording me the time and opportunity to pursue an advanced degree these past two years.

Finally, I would like to thank my family. Dad and Mom, you have always encouraged me to expand my horizons and expressed confidence in me even when I felt it was unwarranted. My dear wife Jenny, your countless sacrifices to allow me to finish this project have not gone unnoticed. Your unwavering support has kept me persevering at times when I did not see how I could get past roadblocks. You deserve my eternal gratitude and admiration.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

Chapter 1

Introduction

A multi-hop wireless network is a communications network consisting of many radio nodes. Some or all nodes source messages that must in turn be received by a single destination node or a set of nodes. Many messages need to be delivered to nodes to which the source cannot directly transmit, instead relying on other nodes in the network to forward the data. Some of the main problems that arise in multi-hop wireless networking are the routing, scheduling, and topology management. The routing problem is one of determining the sequence of nodes through which data must travel between the source and the destination. The scheduling problem is one of allocating bandwidth to specific connections or links so that data routed through those nodes can be transmitted. This is often performed by dividing time into slots and assigning each slot to a set of links that can all be active without interfering with one another. Broadly stated, topology management is a problem of designating direct links and assigning neighbors so that the network can perform up to certain measures, often including connectivity, throughput, and latency. The overarching goal of these problems is to maximize resource utilization to intelligently route packets through as few nodes as possible while also maximizing the volume of data transmitted.

Military wireless networks are similar in many ways to commercial wireless networks, but the scenario in which they are used and battle space demands require special functions or capabilities.

The first problem that we study is topology management for an airborne radio

network. The mobility of aircraft causes the quality of connections among them to improve or deteriorate based on a number of factors, including distance. We study methods whereby the network topology can me adapted to reflect the dynamic channel conditions.

We then study a ground radio network. Military operations require communications in locations removed from the infrastructure of wired internet and the power grid. Performance in these networks deteriorates as the number of users increases. The general problem we study is how to increase throughput in such networks, which we propose to address by adding relay nodes to increase scalability of such networks and then present methods whereby we quantify the benefit in throughput associated with relay nodes.

An aspect common to both problems is that finding optimal solutions to scheduling, routing, and topology management problems often requires mixed integer linear programming, a mathematical optimization or feasibility problem. Depending on the size of a network, the computation involved with such an approach is prohibitive, which is why we develop approaches to approximate mixed integer linear programming solutions. For the airborne radio network, we develop distributed heuristic algorithms, and for the ground radio network we present methods by which we can find upper and lower bounds on the values of optimal solutions.

In Chapter 2, we study the mobile airborne radio network problem, in which a network of aircraft cooperate as part of the same mission. In a general sense, an airborne network consists of some number, large or small, of aircraft. The aircraft may be of different varieties and may be assigned tasks specific to their design capabilities. Because all of these different aircraft share the same airspace and possibly the same overarching mission, they all must be able to communicate with each other. It is anticipated that aircraft of the same type that share the same sub-mission, designated here as a *flight group*, will need to communicate more with each other than with aircraft from other flight groups.

The communications system on which we focus is designed to prevent adversarial forces from intercepting and even detecting transmissions. Designed for this purpose,

highly directional beams are used instead of omnidirectional antennas because of the decreased likelihood that a third party might detect radio activity. Using these highly directional antennas, one node transmits to one receiver, and the signal neither can be received by nor interferes with any user other than the intended recipient. We say a *link* exists between two aircraft if they transmit directly to each other.

However, airborne networks are not stationary, leading to dynamic link capacities. Two linked aircraft might share data at high rates at one moment, only to see the rate drop as, among other factors, the distance between them increases. The dynamic nature of the link capacities motivates the need for topology management.

We work with aircraft equipped with two transceivers each. This means that each aircraft can have at most two direct links with other aircraft. The topology management problem that we study is one of link activation: Given that each aircraft can be directly linked to two other aircraft, we seek to determine what links should be active in the network so as to improve or even optimize performance according to some measure.

The quality of a link activation solution can be quantified by several measures, to include the percentage of successful data packet deliveries, average packet propagation delay, or maximum packet propagation delay. We choose to evaluate solutions according to maximum link load, as it relates to all of these measures.

We can use mixed integer linear programming to periodically update the topology, removing and adding links where necessary to reach an optimal performance level. However, as the computation involved in such a technique is likely prohibitive, we offer distributed heuristic algorithms that determine periodic updates. We develop multiple algorithms that require different levels of computation and awareness of link capacities and build scenarios on which to test them. We present the results of implementing the various methods and compare them against each other, highlighting the strengths and weaknesses of the methods. This work can be used by system developers to understand how best to approach the topology management problem given hardware specifications and mission requirements.

In Chapter 3, we study the ground tactical radio network problem. It is designed

to be deployed in areas in which there is little or no infrastructure. While the radios may be mounted on mobile vehicles, the speed of such vehicles or the relatively small distance they are anticipated to cover allows us to treat the network as static. When nodes are within a specified range of each other, they can communicate. Unlike the network in Chapter 2, the links in this network have the same capacity. If two nodes are within transmission range and communicate, the link supports a certain rate. That rate does not increase or decrease if the distance between them changes. If two nodes are too distant to communicate directly, any traffic that needs to flow between them must be routed through other nodes.

Another difference between the system studied in Chapter 3 and that studied in Chapter 2 is that these ground radios can interfere with each other. If one radio transmits in a certain direction, and another radio is near enough in terms of distance and direction, the second radio might not be able to receive from any other radio because the first radio causes interference. Wireless ad hoc networks are subject to performance-limiting interference constraints because all transmissions share the same medium. These constraints do not have nearly the same effect in wired networks because signals are isolated to a greater extent. Largely due to interference, it has been shown that the throughput of each user in multi-hop wireless networks decreases as the number of nodes sharing the channel increases under most conditions, making scalability a vital concern for such networks. Regardless of the size of the network, it is desirable to achieve the highest possible rates of traffic flow.

We propose solving the scalability problem by introducing an advantaged node. Like a cell tower, this node does not source traffic, it only relays traffic. We refer to this advantaged node as the *relay node* and all other nodes as *ground nodes.* Also similar to a cell tower, the relay node is elevated on a portable pole or tower, giving it different transmission and interference properties from the ground nodes. We allow any ground node to transmit to the relay node, regardless of distance from the ground node to the relay node. Such transmissions do not interfere with other ground nodes. When the relay node transmits, it interferes with all ground nodes so that no other transmissions may occur simultaneously. The relay node otherwise has the same

capabilities as the ground nodes in that it transmits at the same rate and cannot receive multiple simultaneous transmissions.

We test our proposed solution by comparing network performance with and without a relay node in various scenarios. The *flow rate* is the amount of traffic that can be generated and delivered for a given flow in a routing and scheduling solution. We evaluate the benefit of a relay node by maximizing the minimum flow rate when it is present and when it is absent. The complexity in finding optimal solutions motivates our use of techniques from literature, development of greedy heuristic algorithms and some other novel techniques to find bounds on performance. By comparing these bounds we can evaluate the benefit of relay nodes. The bounding techniques that we develop in Chapter 3 apply broadly to wireless systems with binary interference models.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Topology Management in Mobile Airborne Networks

2.1 Introduction

2.1.1 Background

At the genesis of military aviation, a pilot's vision was among the more important qualifications because it increased his situational awareness. Throughout the decades, the instruments on advanced military aircraft have allowed the pilot to know so much more about his surroundings and situation. But alone, a pilot is still much more limited than he is when working with a squadron toward a common mission. It is essential that pilots have the ability to communicate pertinent information from radar, weapons targeting, missile tracking, and other systems, as well as voice messages to other pilots with the same mission so as to create an even more complete image for all cooperating pilots. While strides made in modern equipment that can track enemies and manage weapons systems enhance the pilot's capabilities, the efficient communication of all the data the sensors collect is paramount. It is for this reason that communication systems must incorporate methods for maximizing the amount of data shared among the different aircraft. As important as vision was to pioneers in aviation are today's sensors and other cockpit instruments, the performance of which

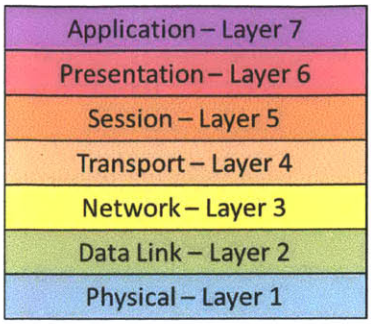

Figure 2-1: The conventional network stack. The system from which this problem derives has a layered network architecture. Topology management takes place at the network level.

depends mightily on the ability to transmit and receive data.

In a general sense, an airborne network consists of some number, large or small, of aircraft. The aircraft may be of different varieties and may be assigned tasks specific to their design capabilities. One such example is the way a U.S. Navy Carrier Air Wing (CVW) operates. A typical CVW might consist of some F/A-18C Hornets, some of which may be assigned to strike missions with others engaging in air-to-air combat. There are likely to be some E-2C Hawkeyes that specialize in tactical early warning to get a radar image of the battle space, as well as some EA-6B Prowlers that engage in electronic warfare. Additionally, some land-based P-3C Orions could be operating in the same space engaged in submarine hunting [1]. Because all of these different aircraft share the same airspace and possibly the same overarching mission, they all must be able to communicate with each other. It is anticipated that aircraft of the same type that share the same sub-mission, designated here as a *flight group*, will need to communicate more with each other than with aircraft from other flight groups.

The communications system from which this problem is derived has an architecture similar to that of the conventional network stack, which is seen in Figure 2-1. Our focus on topology management, falls under the scope of the network layer.

The communications system on which we focus is designed to prevent adversarial forces from intercepting and even detecting transmissions. Designed for this purpose,

highly directional beams are used instead of omnidirectional antennas because of the decreased likelihood that a third party might detect radio activity. Using these highly directional antennas, one node transmits to one receiver, and the signal neither can be received by nor interferes with any user other than the intended recipient. We say a *link* exists between two aircraft if they transmit directly to each other.

However, airborne networks are not stationary, leading to dynamic link capacities. Two linked aircraft might share data at high rates at one moment, only to see the rate drop as, among other factors, the distance between them increases. The dynamic nature of the link capacities motivates the need for topology management.

We work with aircraft equipped with two transceivers each. This means that each aircraft can have at most two direct links with other aircraft. The topology management problem that we study is one of link activation: Given that each aircraft can be directly linked to two other aircraft, we seek to determine what links should be active in the network so as to improve or even optimize performance according to some measure.

The quality of a link activation solution can be quantified by several measures, to include the percentage of successful data packet deliveries, average packet propagation delay, or maximum packet propagation delay. We choose to evaluate solutions according to maximum link load, as it relates to all of these measures. Link *load* or *utilization* is defined as the average rate of traffic flow across a given link divided by its capacity.

2.1.2 General Approach

The communications network is modeled as a graph of nodes and edges ($G = (N, E)$). The nodes represent individual aircraft and the edges connecting the nodes represent possible links across which data can be transferred. Because of the mobility of the aircraft and other dynamic factors that can include atmospheric disturbance, the capacity of the edges are time variant. Two nodes that are near each other without geographical barriers separating them and in clear atmospheric conditions are connected by a link with a high capacity. However, as they move relative to each

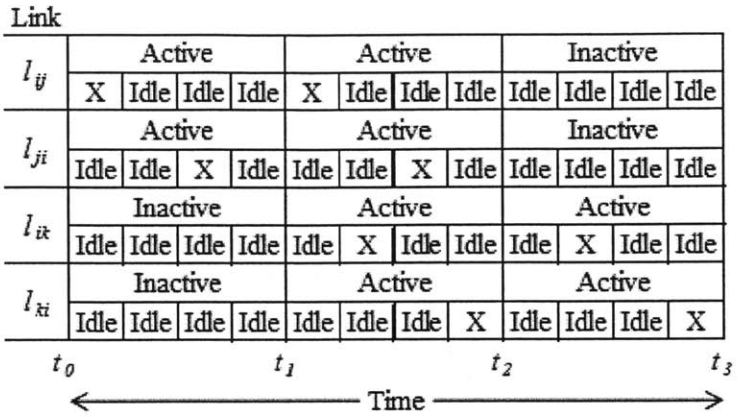

Figure 2-2: This graphic shows the difference between active and inactive. The links shown are either active or inactive for each time slot (t_0 to t_1, t_1 to t_2, t_2 to t_3). Within the time slot, transmission only occurs in one direction across a link (denoted by X) one-quarter of the time slot, and is idle the remainder of the time slot. This allows for a node to have two active incident links, and to be able to transmit and receive from both in a time slot. It is also possible for a link to be active in one direction more than in the other direction in a particular time slot, so long as the combined duration of its activity is one-half the length of the time slot.

other or as they experience other disturbances that degrade the channel condition, the capacity of the link may diminish.

In this wireless network, a link can be established between any two nodes. The topology management problem is to determine which of the potential links to activate. If a link is *active* for the period from time t_0 to time t_1, it can support some rate of traffic. There need not be continual data transmission across a link from t_0 to t_1 for it to be active during that time period. In fact, given that each node cannot be adjacent to more than two active links, there is a time division multiplexing scheme that divides time into slots and assigns which transmissions occur at which slots. Only active slots may be assigned slots. This is demonstrated in Figure 2-2.

Any valid set of activated links must satisfy the constraint that no more than two edges incident to each node can be active. Figure 2-3 shows a few valid link activation sets for a network with 6 nodes. But in addition to satisfying this minimum constraint, there are additional desirable properties for a solution. The first is that the network must be connected, or that given any two nodes in the network, there

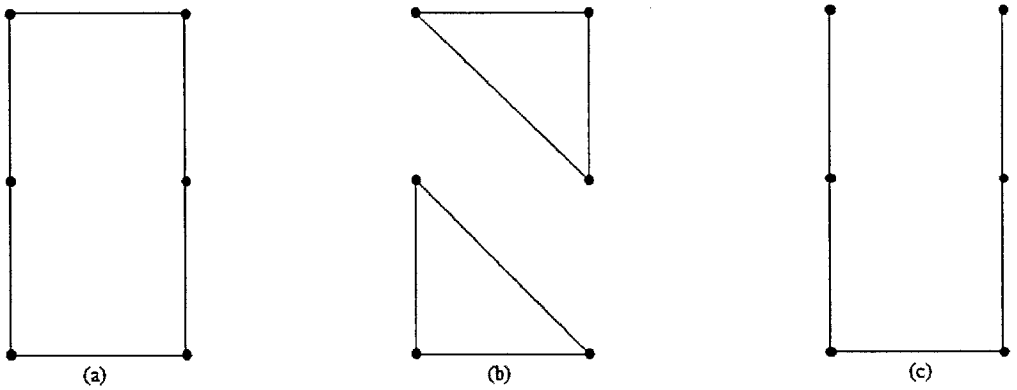

Figure 2-3: Possible results of link activation sets. All solutions are limited to two active links per node. (a) A ring. (b) A disconnected graph. For some pairs of nodes, it is impossible to exchange data. (c) While connected, there is only one path between any pair of nodes because two nodes are only adjacent to one active link.

must be a path of active links connecting the two nodes. Additionally, it is beneficial that there be two active edges incident to every node. The result is a ring topology. The alternative to a ring topology that still results in a connected graph with no more than two active links per node is a line topology. The advantages of a ring topology over a line topology are outlined below.

One benefit of a ring is that the maximum number of hops separating any two nodes is $\frac{n}{2}$, where n is the number of nodes in the graph. The routing problem is also quite simple in a ring, as there are only two routes between a source and destination: clockwise and counterclockwise around the ring. The orientations *clockwise* and *counterclockwise* refer to topological orientation. Aircraft are located in three dimensions, and the ring may not physically be circular. Nonetheless, there are two distinct orientations. For most pairs of nodes, one of these two routes contains fewer hops than the other. Furthermore, whereas a line topology is vulnerable to becoming disconnected if any one link fails, the result of a link failure or node loss in a ring topology is still a line topology, which is connected. And while an optimal ring might require some links to cover greater distances than a line, as seen in Figure 2-4, the ring has more potential paths between sources and destinations, and the additional routes alleviate some of the congestion.

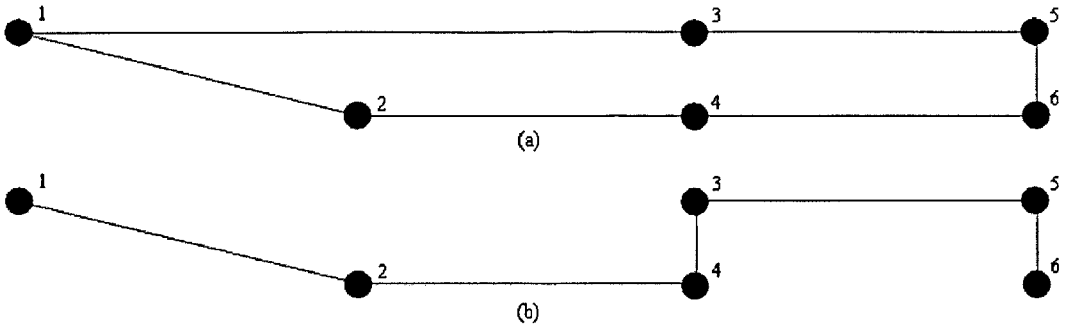

Figure 2-4: The link activation sets resulting in the ring in (a) and the line in (b) minimize the maximum link length. The two nearest nodes to node 1 are nodes 2 and 3, so in a ring there must be at least a link as long as the distance between nodes 1 and 3. The result is a longer link, which would correspond to lower link capacity. The line topology does not have such a long link, but any advantage in shorter links is likely overcome by the existence of multiple routes between source and destination nodes in the ring. In addition, the ring is a more robust structure.

The quality of a solution can be determined by the most heavily loaded link. Bottlenecking arises when the performance or capacity of the network is limited by one or a few components. When a link is heavily loaded, it causes delays for all traffic routed through that link. Traffic has a tendency to be bursty, and the variability in traffic rates leads to queuing delays. As the rate of traffic flow across a given link approaches its capacity, the variability causes delays to increase without bound [9], which is certainly an undesirable outcome.

The functional unit of a flight group and its special traffic demands impact the quality of a solution. An aircraft may have more data to share with the other aircraft in its flight group than with other aircraft, and so it is beneficial for aircraft in the same flight group to be consecutive nodes in the ring. If a node is between two nodes of the same flight group, it becomes a forward-only node for those packets that only need to be delivered to the other group. But by segregating nodes of different flight groups, a node never needs to forward traffic that is intended to stay within a separate flight group. Figure 2-5 illustrates scenarios when flight groups are segregated and when they are integrated.

At takeoff, some initial set of links are activated, as in Figure 2-6. They criteria

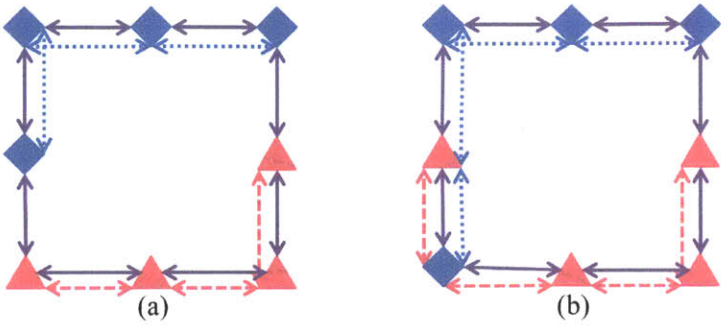

Figure 2-5: The blue diamonds and red triangles belong to two different flight groups. Red dotted arrows indicate flow of intra-group traffic for the red group, and blue dotted arrows indicate flow of intra-group traffic for the blue group. Solid purple arrows indicated inter-group traffic. The diagram in (a) shows which links support which types of traffic when all nodes in a flight group are consecutive in the ring structure. All links need to support at most the inter-group traffic and intra-group traffic for the group to which its end nodes belong. The diagram in (b) can result when flight groups are not all consecutive nodes in the ring structure. It is seen that some links must be able to support inter-group traffic and intra-group traffic for multiple groups.

by which the link activation set is determined might range from anticipated distance between aircraft to anticipated need for particular aircraft to share information.

However, as aircraft depart from the initial anticipated positions, or perhaps as weather, obstructions, and other factors affect the data rate that links can support, the initial link activation set may become suboptimal.

A remedy for the situation calls for modifying the link activation set to reflect new link capacities. One such potential modification is shown in Figure 2-8. Such

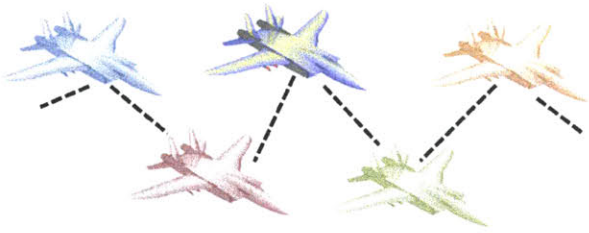

Figure 2-6: A possible initial link activation set.

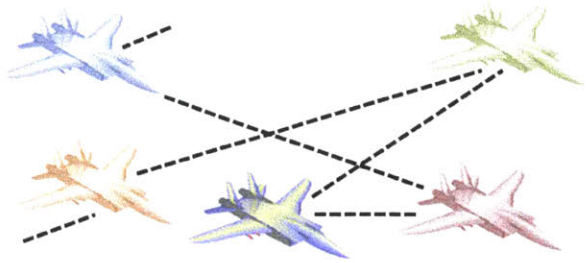

Figure 2-7: After some time elapses, the initial link activation set may no longer be optimal.

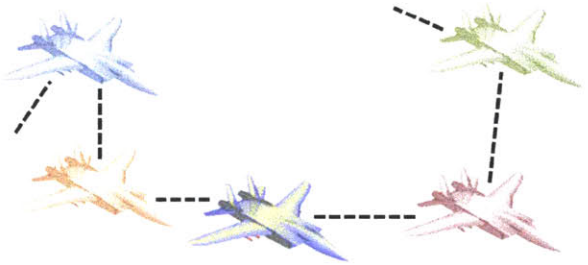

Figure 2-8: By changing which links are active, performance can improve. Compare the length of links with those in Figure 2-7.

modifications need to occur automatically, as the pilots cannot be expected to manually manage the topology or fly in such a way as to maintain the quality of a certain link activation set. The remainder of the chapter focuses on ways to identify when a new link activation set should be implemented and how it should be determined.

The remainder of this chapter is organized in the following manner. In Section 2 we discuss how this problem relates to the literature, and how other works have influenced the approaches presented in this chapter. In Section 3 we present a mixed integer linear programming (MILP) approach to topology management for this problem. We also detail the specifics of the traffic and mobility models, and then present results from implementing the MILP approach. In Section 4 we outline several heuristic algorithms to solve the same problem, and give results from implementing the various

heuristic approaches, comparing them to the results from the MILP approach. In Section 5 we summarize the findings of this chapter.

2.2 Related Work

Baker, Ephremides, and Flynn [2] presented two distributed algorithms to manage wireless multi-hop networks to form clusters and create links among the nodes. In a related work [3], Ephremides, Wieseltheir, and Baker demonstrated how distributed control in these algorithms provide the conditions for survivability. In military airborne networks, it is impractical to have a central ground-based topology controller due to aircraft mobility. But with the risk of losing an aircraft, it is risky to task just one aircraft with topology management for the entire network. While the connectivity of the networks studied by these authors is different from the airborne networks we study, we present distributed algorithms to achieve survivability and cut down on computation costs while sacrificing as little as possible in terms of performance.

While focusing on wavelength division multiplexing based packet networks across optical fibers, Narula-Tam and Modiano [5] differentiate between the *physical* topology, which refers to what physical links exist between nodes, and *logical* topology, which indicates which nodes can directly communicate without needing one or more intermediate nodes to forward communications. Although wireless networks are slightly different because there is no need for physical cables to be installed for two nodes to communicate, our problem is also logical topology management, with the same end goal of decreasing link load. The paper also describes link exchange algorithms, which are adopted in the distributed heuristic algorithms later in this chapter.

Ramaswami and Sivarajan [7] also work with optical networks and logical topology management. On the small scale, they demonstrate the effective use of mixed integer linear programming in solving the joint logical topology design and routing problem. However, since they show that this approach does not scale to larger networks because of computational intractability, they show how by splitting the problem into separate

routing and topology management problems, they can use heuristic algorithms for topology management, predicated upon which they can solve the routing problem using linear programming. We similarly decrease the problem size by assuming a routing solution and solving for a link activation set, or logical topology. We assume that all packets are routed according to shortest path. And given the ring structure of our solutions, if a packet is broadcast, it is propagated around the ring in both directions until all nodes have received the packet.

The aircraft mobility model stems from the random waypoint model developed by Johnson and Maltz [11] and later revised [12]. The revised version has become standard for mobile computing research. We use this to model the mobility in aircraft. Tailored to our specific problem, one key difference is that we are concerned only with relative mobility, that is how aircraft move in relation to other aircraft, not to fixed locations.

2.3 Mixed Integer Linear Programming Approach

Mixed integer linear programming is a common method for optimizing some objective function over a set of feasible solutions. The problem is formulated so that, given an existing link activation set and link capacities, finding an optimal solution to the MILP results in an optimal link activation set.

By allowing time to elapse after implementation of a link activation set solution, some link capacities change. In this section, we develop a MILP formulation and solve it periodically to determine changes in link activation to improve network performance.

2.3.1 Mixed Integer Linear Programming Formulation

Initially, links are configured so as to optimize the objective function in the MILP formulation below. This formulation is developed for unicast traffic. Later we explain how this formulation can be used to model broadcast traffic as well. A critical difference is that for unicast traffic, each node that receives a packet either discards

it or forwards it, except the final node. For this reason, multiple paths between two nodes may be equivalent, while in broadcast traffic, every node receives the same packet, and so two paths with the same source and terminus nodes may be different in that one allows the packet to be received by all nodes, while another may skip some intermediate nodes. While we principally study the case in which all traffic is broadcast, we can use the unicast model. The objective is to minimize the maximum link load, or the ratio of the traffic across a link to its capacity. Equation 2.1 is the objective function of the MILP.

$$\min \max_{i,j \in N} \frac{F_{ij}}{C_{ij}} \tag{2.1}$$

N is the set of all nodes, and given any pair of nodes (i, j), F_{ij} represents the total flow of traffic from node i to node j. C_{ij} is the capacity of the link from node i to node j. The equations and inequalities that follow are the constraints subject to which all feasible solutions conform.

$$\sum_{j} x_{ij} = 2, \forall i \in N \tag{2.2}$$

The binary variable x_{ij} is used to indicate whether or not the link between nodes i and j is active. It takes the value 1 to indicate that link (i, j) is active and 0 if link (i, j) is inactive. This constraint that only two links incident to each node be active comes from the fact that each aircraft is equipped with only two transceivers.

$$x_{ii} = 0, \forall i \in N \tag{2.3}$$

$$x_{ij} = x_{ji}, \forall i, j \in N \tag{2.4}$$

The constraints in Equations 2.3 and 2.4 are fairly straightforward. There are no links that originate and terminate at the same node, and if one node transmits directly to a second, the second node can also transmit directly to the first.

$$\sum_{i \in G, j \notin G} x_{ij} = 2, \forall G \in \boldsymbol{G} \tag{2.5}$$

Define $\boldsymbol{G}$ to be the set of all flight groups, and let G represent a particular flight group. The above constraint limits solutions to those in which there are only two active links between aircraft in a group and aircraft outside the group. Alternatively put, the constraint in Equation 2.5 requires nodes in the same group to be consecutive in the ring. As explained earlier, if aircraft in a group are not consecutive nodes in the ring, nodes outside the group will be required to forward traffic just intended for the group.

$$F_{ij} \leq x_{ij} C_{ij}, \forall i, j \in N \tag{2.6}$$

As defined earlier, F_{ij} and C_{ij} represent the total flow and the capacity, respectively, for the link from node i to node j. This inequality restricts solutions to those in which flow does not exceed capacity across any link.

$$F_{ij} = \sum_{s,d \in N, s \neq d} f_{ij}^{sd}, \forall i, j \in N \tag{2.7}$$

The variable f_{ij}^{sd} is the amount of flow of data originating at source node s with destination d that flows from node i to node j. Since F_{ij} is the total flow from i to j, it is the sum of all flows from i to j.

$$A_i^{sd} = \begin{cases} T^{sd} & i = s \\ -T^{sd} & i = d \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

In this equation, A_i^{sd} is the amount of supply or demand at each node of a particular flow. For flows of which a node is the source, it has some amount of surplus, T^{sd} of flow, and the opposite is true for destinations. Nodes that are neither the source nor destination of a particular flow have no surplus or demand. These are for flow balance purposes, and the values of T^{sd} are inputs.

$$\sum_j (f_{ij}^{sd} - f_{ji}^{sd}) = A_i^{sd}, \forall i, s, d \in N \tag{2.9}$$

This constraint is a flow balance equation. In combination with the previous constraint, for a particular source-destination pair, the only two nodes at which flow in does not equal flow out are the source and destination, and for a particular flow, the amount of traffic leaving the source equals the amount of traffic arriving at the destination. The remaining constraints simply set bounds on the variables.

$$x_{ij} \in \{0, 1\}, \forall i, j \in N \tag{2.10}$$

$$f_{ij}^{sd} \geq 0, \forall i, j, s, d \in N \tag{2.11}$$

One constraint that would normally be found in such a MILP formulation for topology management is one which ensures that all nodes are connected. This constraint would be redundant unless there is a partition of nodes such that each source-destination pair lies in the same proper subset of the nodes. But because in the case studied, we model traffic as all-to-all unicast, so that every source node sources traffic for every other node.

The complete MILP formulation together is:

$$\textbf{min} \max_{i,j \in N} \frac{F_{ij}}{C_{ij}} \tag{2.1}$$

Subject to:

$$\sum_j x_{ij} = 2, \forall i \in N \tag{2.2}$$

$$x_{ii} = 0, \forall i \in N \tag{2.3}$$

$$x_{ij} = x_{ji}, \forall i, j \in N \tag{2.4}$$

$$\sum_{i \in G, j \notin G} x_{ij} = 2, \forall G \in \boldsymbol{G} \tag{2.5}$$

$$F_{ij} \leq x_{ij} C_{ij}, \forall i, j \in N \tag{2.6}$$

Figure 2-9: The example problem is topology management for a network with 20 aircraft, equally divided among 4 flight groups.

$$F_{ij} = \sum_{s,d \in N, s \neq d} f_{ij}^{sd}, \forall i,j \in N \tag{2.7}$$

$$A_i^{sd} = \begin{cases} T^{sd} & i = s \\ -T^{sd} & i = d \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

$$\sum_j (f_{ij}^{sd} - f_{ji}^{sd}) = A_i^{sd}, \forall i,s,d \in N \tag{2.9}$$

$$f_{ij}^{sd} \geq 0, \forall i,j,s,d \in N \tag{2.11}$$

2.3.2 Example Problem: 20-Node Network

We use the example problem, later also referenced as the 20-node problem, to illustrate how link quality is affected by mobility and how the attempts presented later are geared toward improving performance by modifying the link activation set. This example problem includes four flight groups with five aircraft each.

Traffic Model

In a real scenario, traffic patterns can quickly become quite sophisticated and modeling can also become complex. Different aircraft platforms using the same network

likely produce traffic at different rates and the intended audience of the transmissions may change as well. Traffic also has a tendency to be bursty, with the variability in traffic rates leading to queuing delays as described earlier. The instability of the system is alarming as traffic rate approaches link capacity. But when dealing with constant deterministic traffic patterns, the queuing problems are insignificant. For the sake of simplicity, we use constant deterministic traffic patterns.

It is apparent why maximum link load is a good choice for the objective function. When the maximum link load is minimized, it allows for the greatest variability in traffic while queuing delays for all links remain at manageable levels. Just one link that is operating at or near its capacity causes a bottleneck and prevents all traffic from using that link. This, in turn results in re-routing, but increases the load on other links, potentially causing additional bottlenecks. For reliable transmission without intolerable delays under variable traffic conditions, minimizing the maximum link load is a useful objective.

The traffic rate in this problem is always set at a low enough level so that flow rate across a link never exceeds its capacity. Another aspect of the traffic model, nodes in the network are capable of producing several types of traffic. A node can send a packet to exactly one other node – unicast traffic – either inside or outside its flight group. A node can also send a packet to all other nodes in its flight group – broadcast traffic. The third and final option is also broadcast traffic, as a node can source a packet that must be received by all other nodes in the network. The volume of traffic routed through a link can be modeled with high fidelity if we limit traffic to just intra-flight or inter-flight broadcast traffic. Let α be the rate of intra-flight traffic generated by all nodes, and β be the rate of inter-flight traffic generated by all nodes. These rates represent constant streams of traffic sourced by all nodes.

Suppose a link connects two nodes in a network with n_1 nodes. Assuming the routing scheme mentioned earlier, half of the inter-group traffic it receives comes from each of the two nodes with which it is directly linked. The node needs to receive inter-group traffic at a rate of $\beta(n_1 - 1)$, and half comes from each link. This means that each link must support an inter-group traffic rate of $\frac{\beta(n_1-1)}{2}$ in each direction, or

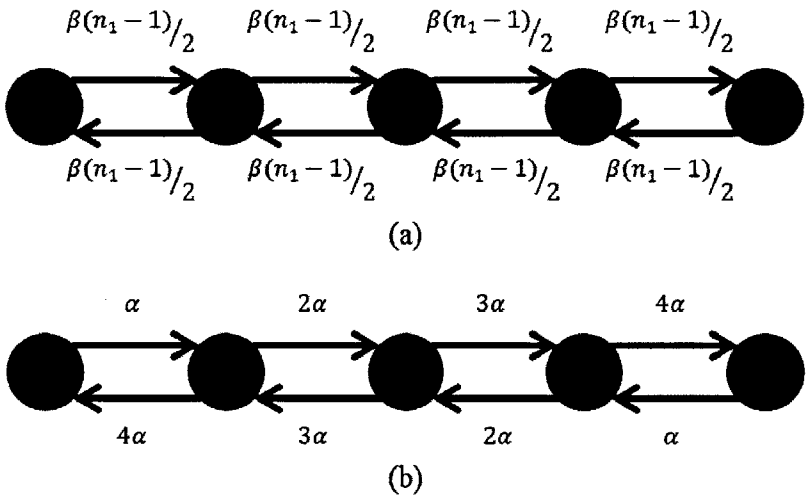

Figure 2-10: In (a), each link needs to support a total rate of $\beta(n_1 - 1)$ of inter-group traffic, where β is the rate of inter-group traffic generated per node and n_1 is the number of nodes in the network. For each inter-group packet a node generates, it is also the final recipient of one inter-group packet, explaining why the rate for each link is the same. In (b), all nodes shown are in the same group. Although not evenly balanced in both directions, all links in the group must support the same total rate of αn_2 intra-group traffic if n_2 is the number of nodes in the group and α is the rate of intra-group traffic generated per node.

$\beta(n_1 - 1)$ total. In addition, each link connecting two nodes in the same group of size n_2 must support additional intra-group traffic at a rate of αn_2. For visualization, see Figure 2-10.

As stated earlier, while this is broadcast traffic, we can accurately model the volume of traffic for each link using unicast traffic flows. Consider a 5-node network without flight groups. All broadcast traffic must be received by all nodes, and each node generates traffic at rate 1. In tandem with Figure 2-11, Table 2.1 shows the route of each flow.

We can model unicast traffic in a way that each link is loaded the same as with the broadcast model. Table 2.2 shows a shortest path routing for each unicast traffic flow. The result is that each link must support the same amount of traffic flow. By scaling the rates of unicast flow, we can use the unicast traffic model to represent

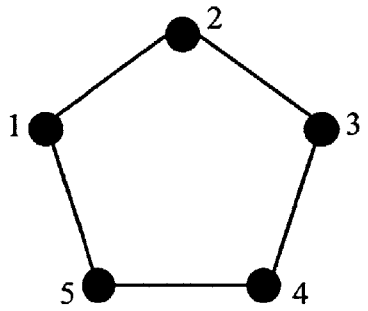

Figure 2-11: A 5-node network.

Source	Route 1	Route 2
1	(1,2),(2,3)	(1,5),(5,4)
2	(2,3),(3,4)	(2,1),(1,5)
3	(3,4),(4,5)	(3,2),(2,1)
4	(4,5),(5,1)	(4,3),(3,2)
5	(5,1),(1,2)	(5,4),(4,3)

Table 2.1: Traffic from each source node in the network in Figure 2-11 is routed in two directions until all nodes have received it. Each link is activated the same number of times (2) in each direction.

Source	Destination	Route
1	2	(1,2)
1	3	(1,2),(2,3)
1	4	(1,5),(5,4)
1	5	(1,5)
2	1	(2,1)
2	3	(2,3)
2	4	(2,3),(3,4)
2	5	(2,1),(1,5)
3	1	(3,2),(2,1)
3	2	(3,2)
3	4	(3,4)
3	5	(3,4),(4,5)
4	1	(4,5),(5,1)
4	2	(4,3),(3,2)
4	3	(4,3)
4	5	(4,5)
5	1	(5,1)
5	2	(5,1),(1,2)
5	3	(5,4),(4,3)
5	4	(5,4)

Table 2.2: Traffic for each source-destination pair in the network in Figure 2-11 with routing. Each link is activated the same number of times (3) in each direction.

the load on each link from inter-group traffic under the broadcast traffic model. In a similar fashion, we can represent the load from intra-group broadcast traffic from unicast traffic.

In the unicast traffic model, each link is activated $\frac{3}{2}$ the number of times as in the broadcast model. If we adjust the rate of each source-destination pair to $\frac{2}{3}$, each link has the same load under both models. This technique applies to represent link loads from both inter-group and intra-group broadcast traffic as unicast traffic.

The reason for this exercise in representing broadcast traffic loads with unicast traffic is to show that we can use the MILP formulation presented previously with a broadcast model, even though it specifically applies to unicast traffic. A broadcast traffic MILP requires additional constraints and resources to solve.

Physical Layer Model

The capacity of a link is the data rate that it can support. A link is bi-directional. We define its capacity as being equal to the sum of the rate at which the first node transmits to the second and the rate at which the second node transmits to the first. Disturbances, such as those caused by inclement weather, can deteriorate channel quality, but it is largely determined by the distance between two nodes. Ignoring disturbances is a reasonable modeling assumption for airborne nodes, as channel quality is more heavily impacted by distance, and any attempt to model disturbances would require additional randomness. A common model for signal attenuation in free space is that power loss is inversely proportional to the square of distance [10]. The result is that we can model link capacity between nodes i and j as $C_{ij} = \frac{\gamma}{d_{ij}^2}$ for some constant antenna gain factor γ and with d_{ij} representing the distance between nodes i and j. This measure refers to the combined capacity of flow from i to j and j to i. Capacity is thus inversely proportional to the square of the distance between the nodes. In reality, other factors may cause the rate to deviate from these values. Also, rate may be a discrete function dependent on node distance, such that the system has multiple data rate transmission settings as opposed to continuous values. However, the techniques used here would be valid in those cases as well. All that is needed at any moment is the real-time capacity of a link, whether affected by factors other than distance, or a discrete-valued function.

Mobility Model

The random waypoint model (RWM) was introduced by Johnson and Maltz [11] to assess their routing protocol for mobile ad hoc networks. It is commonly used in simulation to evaluate protocols and algorithms in mobile scenarios. As it applies to this problem, aircraft may change location and velocity over time, and random variables can be used to model these changes.

The true location of the aircraft is not important for the changes in channel conditions. Rather, it is the relative locations since channel capacity is a function of

distance between the aircraft. Presumably there is some maximum range that aircraft cannot exceed, and for that reason the aircraft are restricted to a square with side length L. Keeping a two-dimensional model allows for easier illustration. Again, this is not true location, but more like a moving frame of reference. If, for instance, all aircraft fly in the same direction at the same speed, their positions relative to each other will not change, so the location diagram would remain constant.

The model used does not represent with high fidelity actual flight velocities or formations, but by introducing randomness in the manner about to be described, a topology management scheme that performs well according to the RWM should also perform well in actual scenarios. Because flight groups are presumed to stay together, a center point is determined for each group around which the planes in that group are located. Initialization is performed by independently placing the center of each group at random in the xy-space as a uniform two-dimensional random variable. Each node is then placed randomly around its respective group center as follows. First, an angular direction Θ_i is chosen randomly on $[0, 2\pi)$ for each node, and then the distance D_i from the group center is an exponential random variable. If the distance D_i along the trajectory Θ_i would place the node outside the boundary of the grid, the node is placed where the trajectory intersects the boundary. After the initial placement, a time step Δt passes, and the relative locations of the nodes change. The group centers move at a random angle with the same distribution as Θ. The distance from the previous location is also an exponential random variable. Nodes are randomly placed around the group center as during the initialization step. Between a given time and the time instance Δt later, the nodes are assumed to travel in a straight line and at constant speed between the two points.

For the example problem, the mean distance from a group center to a node in its group is 4 miles and the mean distance the group center travels in a time interval Δt is 5 miles. The boundaries form a square with $L = 100$ miles. These numbers do not correspond to specific flight speeds or time intervals between updates, but can be easily altered to reflect true values. As will be presented later, reconfiguration can happen after each time interval Δt. For an operator to decide how frequently to

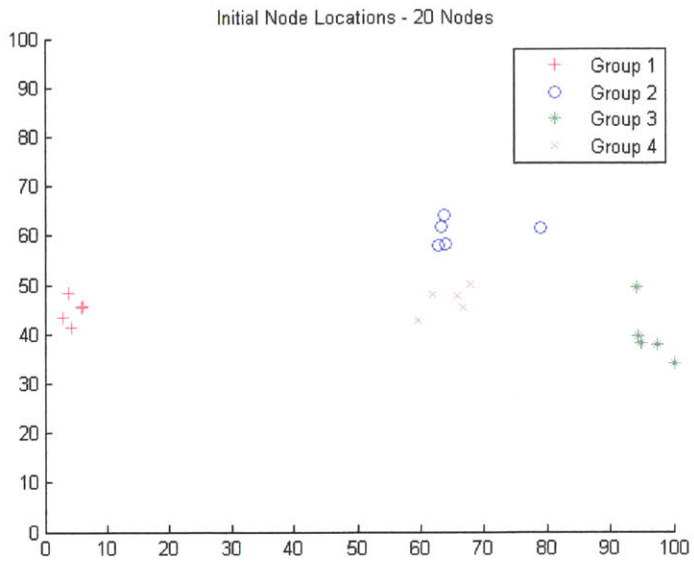

Figure 2-12: Initial node placement of the 20 nodes in the example problem. Each group of 5 aircraft is distinguishable by a unique color and mark.

reconfigure the links, the speed of the aircraft is a key parameter. For the remainder of this chapter, the relative speed and time intervals are held constant to allow for easier comparison across scenarios.

Implementation

Next, Figure 2-12 indicates the initial placement of the nodes in this 20-node problem. The nodes in each group have a distinct color and mark.

Before solving the optimization problem to determine the initial link activation set, it is necessary to specify some of the parameters. First, for the traffic pattern, recall that α and β represent the intra-group and inter-group traffic generation rates, respectively. And given the physical layer model, the link capacity for any node pair is given as $C_{ij} = \frac{\gamma}{d_{ij}^2}$. Assuming nodes share more information with their flight group than with the network at large, we set values of $\alpha = 10$ and $\beta = 1$.

We wish to guarantee that there is always a feasible solution to avoid the scenario in which a link is overloaded and packets are deleted before delivery when setting

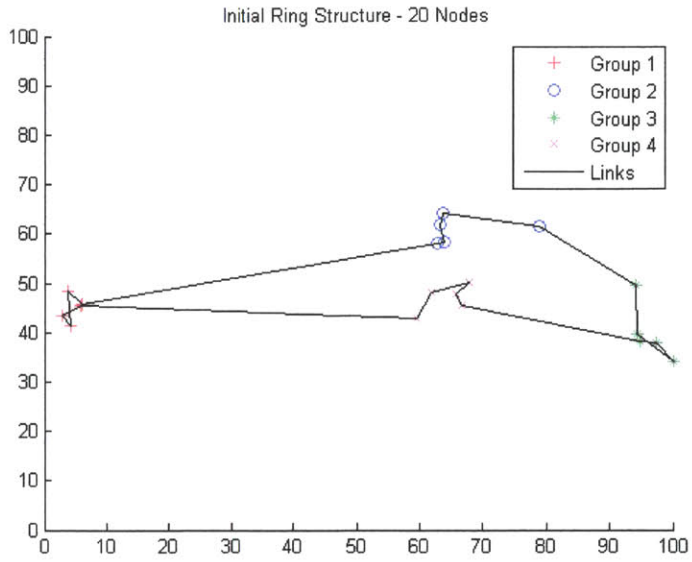

Figure 2-13: Solving the MILP formulation to optimality given the initial node locations results in this solution.

parameters. We have shown that for links connecting nodes of different flight groups, a data rate of 19 must be supported for the 20-node problem. Given that the greatest distance between any two connected nodes cannot be greater than being in opposite corners of the square region, the distance cannot exceed $100\sqrt{2}$. As $C_{ij} = \frac{\gamma}{d_{ij}^2}$, it is necessary that $\frac{\gamma}{(100\sqrt{2})^2} \geq 19$, or $\gamma \geq 380,000$. We let $\gamma = 380,000$.

Links connecting nodes in the same group must support a higher rate of traffic: $19 + 5 * 10 = 69$. For the value of γ just specified, no two linked nodes in the same group can be separated by more than 74.2 miles. The probability of such an event is less than the probability that two nodes are a combined 74.2 units from the flight group center, so it is certainly bounded above by 2×10^{-7}, where this probability is derived from the Erlang distribution. This is such a rare event that it can be disregarded.

Initially solving the MILP to optimality results in the link activation set that can be seen in Figure 2-13.

Under the current mobility model, after a time period Δt elapses, the node locations change. After one such time period, without modifying the link activation set

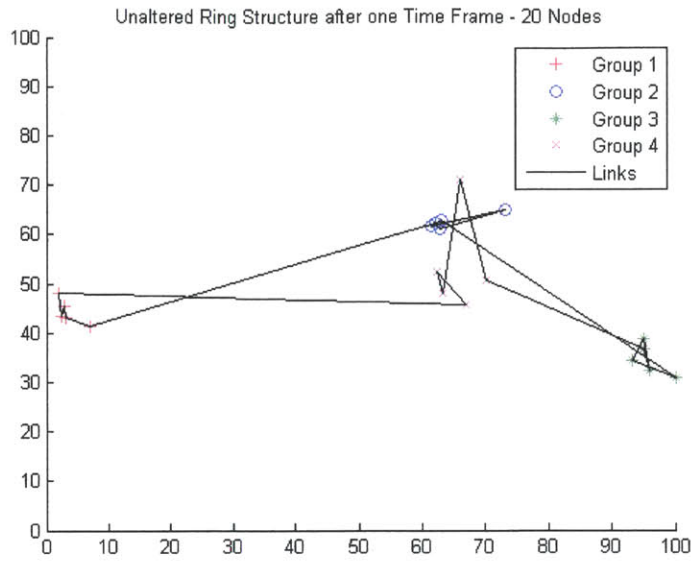

Figure 2-14: Nodes move relative to one another. This graph displays the locations of nodes after Δt time elapses from the initial configuration in Figure 2-13.

from the initial solution, the network can change to the image seen in Figure 2-14.

Even after a time period of only Δt elapses, much can change under this mobility model. And after subsequent time intervals elapse, it is possible that the initial link activation set results in even worse performance. Figure 2-15 shows the same scenario after 20 time frames of duration Δt elapse.

One response to this problem is to periodically re-optimize the link activation set according to the MILP formulation using updated channel condition information. Figure 2-16 illustrates the new link activation set resulting in a minimized maximum link load after 20 time frames of Δt have elapsed.

One observation relates to the fact that changing the link activation comes at a cost in reality. Two nodes that are many hops away from each other under an existing link activation set may not know the exact, real-time location of each other, which presents an obstacle to forming new links. Then there is some time delay in moving the antennas so that a new link can be formed. Even under the assumption that antennas can change direction instantaneously and every node has real-time information of the exact location of all other nodes, there is still additional loss. The

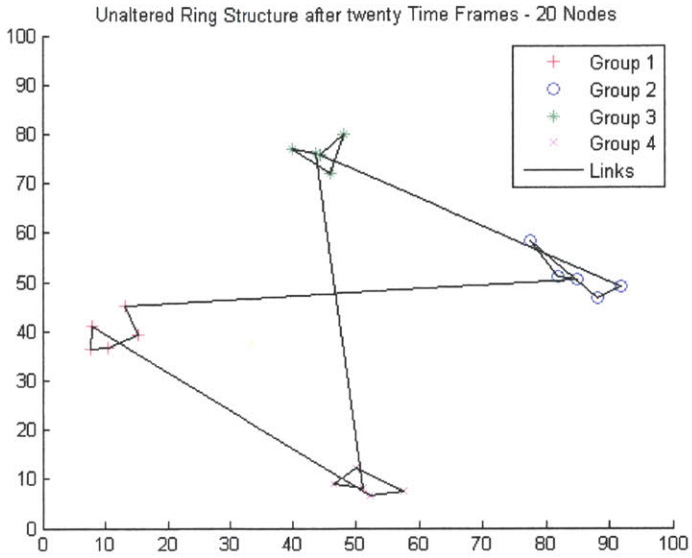

Figure 2-15: When left unaltered, a link activation set that at one time was optimal, can lead to poor performance. This shows what can result from leaving the initial link activation set from Figure 2-13 in place even after a time period of $20\Delta t$ elapses. Notice that there are some links spanning greater distances than necessary. This problem can be avoided by periodic updates to the link activation set.

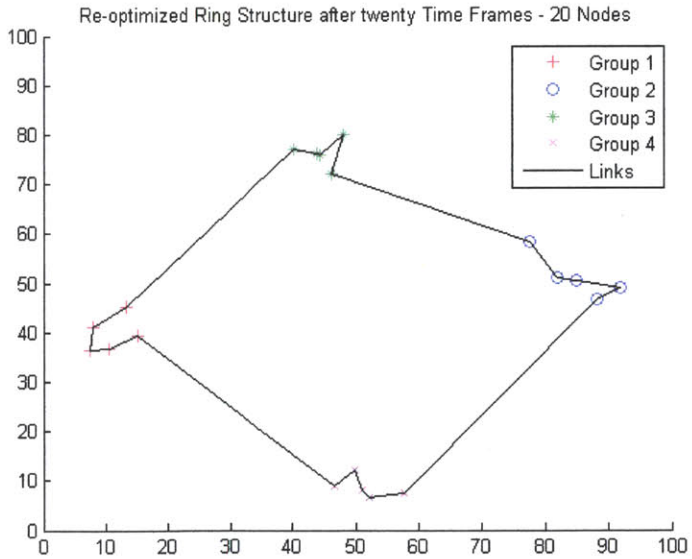

Figure 2-16: After a time period of $20\Delta t$ from initialization, the link activation set is modified to minimize the maximum link load.

information that is being relayed at the time of reconfiguration must find new routes to reach all intended destinations, which may sometimes involve being re-directed through nodes that have already received and relayed it, causing inefficiencies in the routing. In order to model this, there can be a penalty associated with each change in topology. Depending on the value of the penalty, it is possible to disincentivize changes to the link activation set to the point that they reach a desirable frequency.

As all of the traffic in this example is unicast, there is the possibility of changing to broadcast traffic. This seems like it would be the majority of traffic in the real scenario, but in determining methods for reconfiguration, there is not much difference in comparing unicast with broadcast traffic here. The objective either way is to minimize a ratio of demand to capacity, and one can easily observe what would happen with broadcast traffic by tuning parameters α, β, and γ.

2.3.3 Larger Scale Problems

The 20-node problem is illustrative in that is shows how to optimize the link activation set periodically according to the MILP formulation. It is interesting to see what happens when the number of nodes increases, and for that reason, while maintaining the number of aircraft per group at 5, we increase the number of groups to 8, 12, and then 16 for a total of 40, 60, and 80 aircraft, respectively. The locations of the nodes are determined in the same way as in the 20-node problem. Initially, each group has a center that is randomly located, and this center moves subsequently according to the RWM so that at each interval its location is some randomly generated distance from its previous location in a random direction. At every time instance, each node is randomly located around its respective group center. The random variables governing location are the same as in the example problem: group centers travel at a mean distance of 5 miles per time frame Δt, and nodes in a group have an average distance of 4 miles from the group center.

We assume the same traffic pattern as in the 20-node problem: each node generates intra-group broadcast traffic at rate α and inter-group broadcast traffic at rate β. There is still no penalty to re-optimize the logical topology. We observe the effects

of changing the frequency of reconfiguration of the active links. In one trial, the link activation set is reconfigured according to an optimal solution to the MILP formulation at each interval Δt. Another trial waits until an interval $5\Delta t$ elapses before reconfiguration, again using the MILP formulation to find an optimal link activation set. The same is also done with intervals of duration $10\Delta t$, and in the last trial, no changes are made to the original link activation set. All trials have a duration of $100\Delta t$.

Results Using MILP Formulation

Because there are twice as many total nodes in the network with 40 nodes but the same flight group size, cutting β in half while keeping α constant results in comparable total traffic demands with the link loads found in the 20-node problem. This allows for easier comparison between the cases with different numbers of nodes, while preserving the condition that there is no set of node locations that can lead to link capacity being exceeded.

Figure 2-17 shows the result from four trials of the 40-node problem. In the first, the link activation set is adjusted after each time frame of duration Δt according to the MILP solution. In the second trial, the MILP solution is used only once every five frames and in the third trial only once every ten frames. In the fourth trial, the original link activation set is unmodified throughout. The time axis begins with 0, indicating the initial configuration, and runs through 100, which represents the network after 100 time frames elapse. The maximum link load after each time frame is displayed.

Figures 2-18 and 2-19 also show instances of the same problem when there are 60 and 80 nodes. In these cases, β is divided by a factor of 3 and 4, respectively, from its original value of 0.01 in the 20-node problem. As with 40 nodes, α is held constant and the guarantee of a feasible solution is still valid.

In addition to the results from a single trial at each reconfiguration frequency that are presented in the figures above, Figures 2-20, 2-21, and 2-22 show averages of maximum link load after each frame over 25 trials for each frequency of reconfiguration.

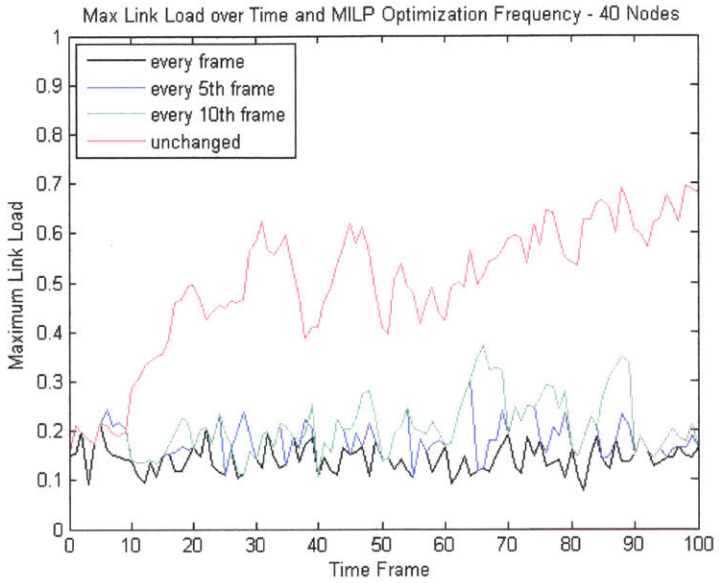

Figure 2-17: Progression of maximum link load in trials for networks with 40 nodes and differing reconfiguration frequencies.

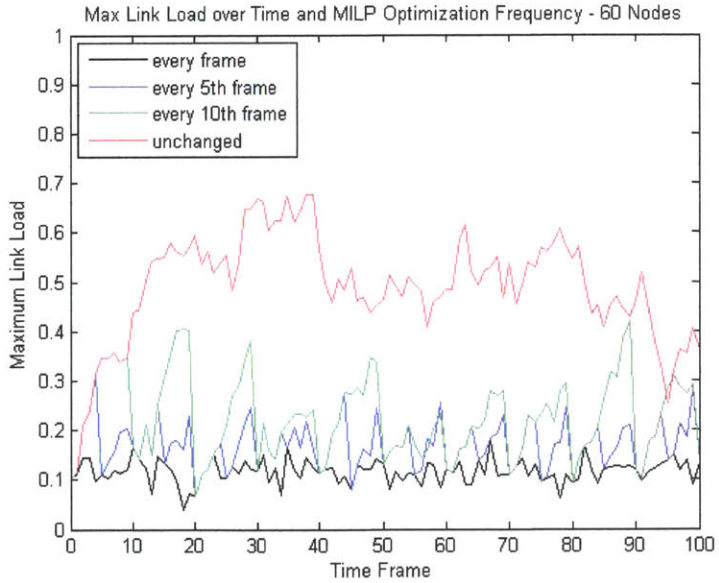

Figure 2-18: Progression of maximum link load in trials for networks with 60 nodes and differing reconfiguration frequencies.

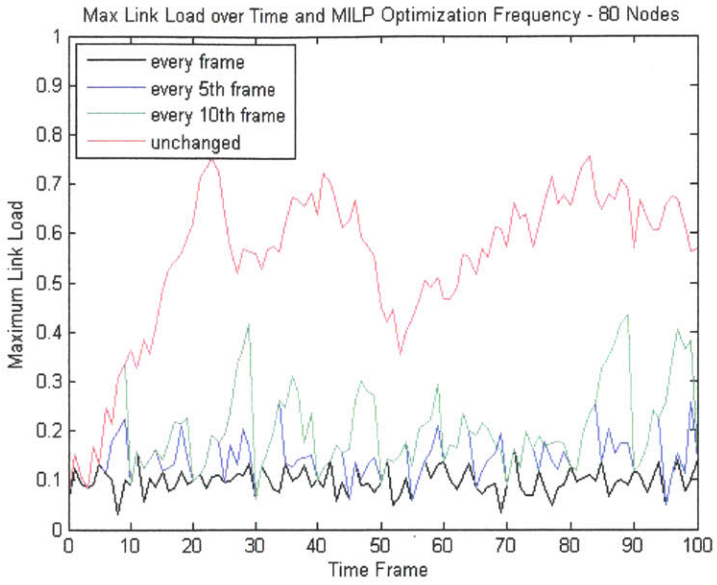

Figure 2-19: Progression of maximum link load in trials for networks with 80 nodes and differing reconfiguration frequencies.

The averaged results give better insight into the value or benefit of reconfiguration frequency.

The general trend is that as a link activation set remains unchanged, performance diminishes until the link activation set is updated to reflect changes in link qualities. Although it is computationally intensive to continually change the link activation set after every time frame, the need for frequent updates is undeniable.

Reconfiguration Penalties

Thus far the model has assumed that any reconfiguration of the link activation set is without cost. An old link activation set can be discarded to discover a new one that leads to the lowest maximum link load. In reality, there is a cost to re-route the traffic in transmission at the time of reconfiguration, in addition to any time delays necessary to establish new links. Whatever costs are involved in swapping nodes, they can be represented by ψ_{ij}, which is a penalty for activating link (i, j) when it was not in the previous link activation set. In the MILP formulation, we can introduce a new

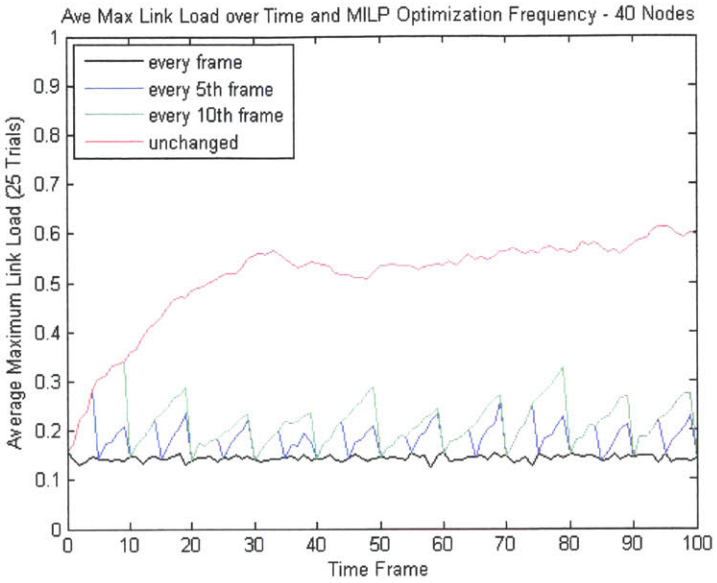

Figure 2-20: Maximum link load is given for each of the four reconfiguration frequencies, averaged over 25 trials, for a network with 40 nodes.

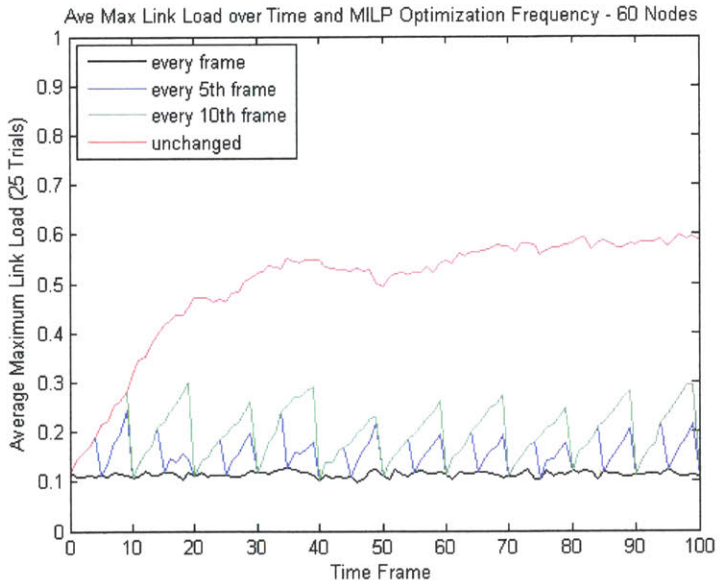

Figure 2-21: Maximum link load is given for each of the four reconfiguration frequencies, averaged over 25 trials, for a network with 60 nodes.

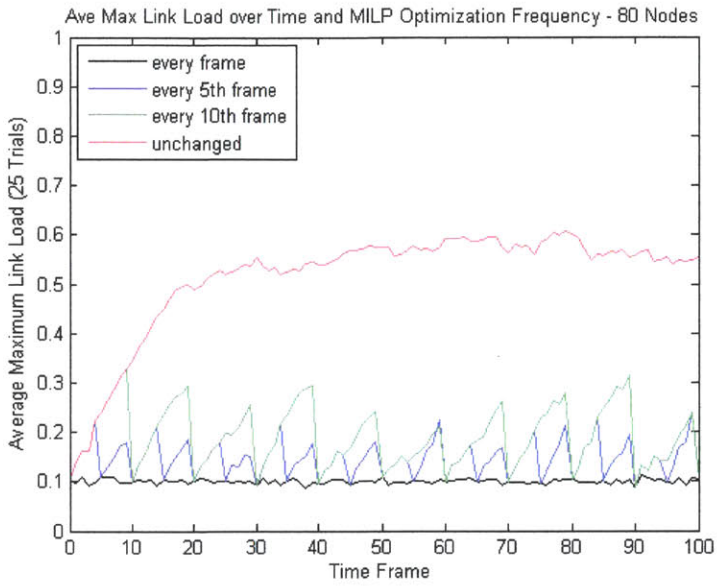

Figure 2-22: Maximum link load is given for each of the four reconfiguration frequencies, averaged over 25 trials, for a network with 80 nodes.

binary variable, w_{ij}, which satisfies the inequality $w_{ij} \geq x'_{ij} - x_{ij}$. In this case, x'_{ij} is the value of x_{ij} in the previous MILP solution. The objective function in Equation 2.1 can be replaced by:

$$\min \left(\max_{i,j \in N} \frac{F_{ij}}{C_{ij}} + \sum_{i,j \in N} \phi_{ij} w_{ij} \right) \tag{2.12}$$

A similar penalty can also be enforced with removing a link, but it suffices to penalize only additions since additions and deletions must occur in equal number.

To determine the value at which to set the penalty, it is useful to consult the trial results when reconfiguration takes place at different frequencies. For 40-node networks, the average maximum link load is 0.1442 when MILP optimization occurs at each time frame. Contrasted with MILP optimization every 10 frames, in which the average maximum link load is 0.2707, we find that by not changing the link activation sets, the maximum link load increases on average by 0.0281 per time frame, at least for solutions that are close to optimal to begin with. This average increase in maximum link load leads us to investigate values of $\psi \leq 0.0281$, because at that penalty level,

ψ	40 Nodes	60 Nodes	80 Nodes
0	29.89	47.33	62.39
0.0001	3.8901	4.3503	4.6928
0.0025	2.5182	3.2594	3.4976
0.005	1.973	2.3081	2.5847
0.01	1.5729	1.9472	2.0873
0.02	1.0873	1.4038	1.5627
0.03	0.3899	0.4492	0.4825

Table 2.3: Average number of new links at each reconfiguration for networks with 40, 60, and 80 nodes and different penalty values for new links.

reconfigurations should become rare. It is expected that the number of changes to the link activation set should decrease as ψ increases. We now use the MILP formulation at each time frame, with the new objective function in Equation 2.12, and compare the number of new links formed per time frame for different values of ψ. These averages are presented in Table 2.3.

The most striking result is the immense drop-off in the number of new links when a very small penalty is introduced as opposed to allowing links to be created without cost. If the previous solution is not considered, as is the case with the penalty-free model, there are combinatorially many solutions with the same objective function value. The one selected is not chosen based on similarity to the previous solution. By introducing a very small penalty, out of the many equivalent solutions in terms of maximum link load, those with fewer changes to the link activation set are given preference. Otherwise, most of the changes to the link activation set have no impact in the maximum link load at all. Without a penalty, an optimal link activation set is determined solely on the basis of what minimizes the maximum link load without consideration for previously established links.

The graph in Figure 2-23 shows how the maximum link load changes over time in a trial of the 80-node problem when different penalty values are incurred.

The study on the effect of different values of penalty functions sheds great insight into how many changes to the link activation set are actually needed in order to obtain optimal or near-optimal results. The graph details how larger penalties result

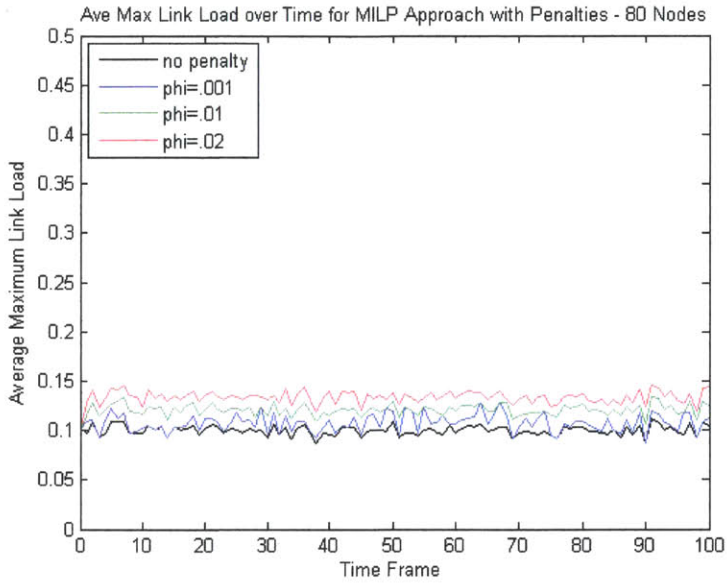

Figure 2-23: For a network with 80 nodes, the progression of maximum link load over time as different penalty values are assessed for changes to the link activation set.

in significantly higher values of the maximum link load compared to the penalty-free solution. On the other hand, there is very slight increase in the maximum link load when there is a small penalty assessed. This shows that with just one or two changes to the link activation set after each time frame, we can effectively lower the maximum link load to near-optimal levels without inflicting major setbacks, which are the result of major overhauls to the link activation set. This revelation motivates the next section, as we present heuristic algorithms that are intended to target the small number of changes that need to be made to keep the network performing at near-optimal levels.

2.4 Heuristic Algorithms

The impracticality of using mixed integer linear programming at each time instance, or even only periodically due to computational complexity gives rise to the need for different methods of determining how to update the link activation set as link capacities change. As with the MILP approach, our objective in these heuristic

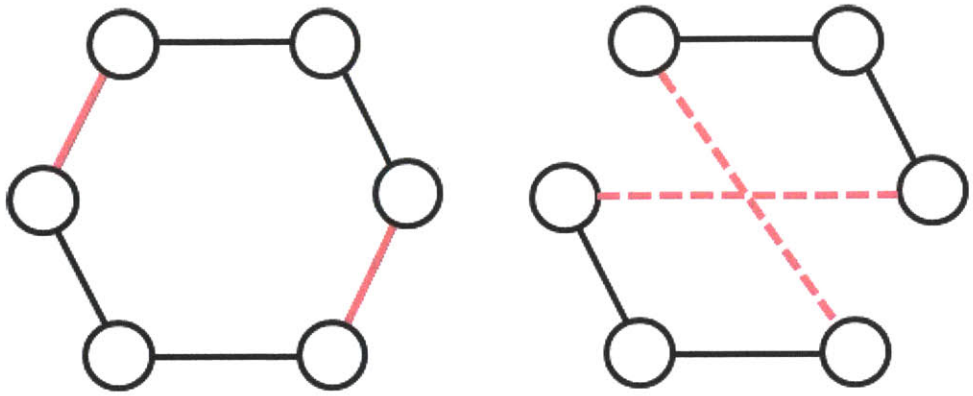

Figure 2-24: A 2-link exchange. The original link activation set is to the left. The two solid links in red are those about to be removed. The image on the right shows the modified link activation set, as dotted red links have been added to replace the two solid red links from the left.

algorithms is to minimize maximum link load. In each of the methods below, whenever a change to the link activation set takes place, it is done to decrease the maximum link load over the domain of potential solutions considered.

It was seen in the previous section that it is generally sufficient to allow for only a small number of new links in order to obtain near-optimal network performance. This is one of the driving factors motivating the development of the procedures that are explained below.

2.4.1 Local Link Exchange

Randomized Local 2-Link Exchange Algorithm

A *link exchange* is when two or more links are removed from the activation set and new links are activated so that the activation set forms a ring once again. A 2-link exchange is when two links are removed and the affected four nodes form two new links in their place, maintaining the ring structure. An example of a 2-link exchange is shown in Figure 2-24, in which the image on the left shows the configuration before the link exchange and the image on the right shows the updated version with a new link activation set. Link exchange operations are also featured in [5].

In the local link exchange heuristic, some node is marked as the control node

at each time frame according to a random process. A neighborhood consisting of some number of nearest neighbors is considered eligible for link exchange operations. Nearest neighbors are determined on the basis of number of hops from the control node in the current configuration. Out of the links in the eligible neighborhood, a 2-link exchange that makes the most improvement in maximum link load for the neighborhood is performed. A time interval Δt then passes before another node is marked at random and the neighborhood of the same size nearest the new control node is eligible for 2-link exchange again. The size of the neighborhood is subject to change, and as it increases to include more nodes in the network, this heuristic algorithm behaves more like a global two-link exchange algorithm, in which any 2-link exchange is considered eligible.

We consider three sizes of neighborhoods. The first includes the $\frac{1}{4}$ of the nodes in the network, the second includes $\frac{1}{2}$, and the third includes $\frac{3}{4}$.

The results of the randomized local 2-link exchange algorithm with varying neighborhood size compared to the baseline of solving the MILP optimization problem at every frame are given in Figure 2-25 for the 40-node network.

There are a few drawbacks to this method. The first is that for smaller neighborhood sizes, there is a large probability that the most heavily loaded link will not even be in the considered neighborhood, meaning no improvement can be made in the network-wide maximum link load. The other drawback is that there is little flexibility in terms of what link exchanges are available. Even when the most heavily loaded link is eligible for exchange, there could be another nearly equally loaded link that cannot be improved. By only allowing one link exchange and limiting which exchanges are eligible, the performance is not great.

The obvious advantages of this method are the reduction in number of new links formed and the small number of solutions that need to be considered is small. Given any two active links, there is only one way to exchange them to maintain the ring. And with a neighborhood of size v, there are at most $\frac{v^2-v}{2}$ pairs of links. This method is fast computationally, but has many restrictions. One obvious improvement would be to mark a control node near the most heavily loaded link instead of selecting it at

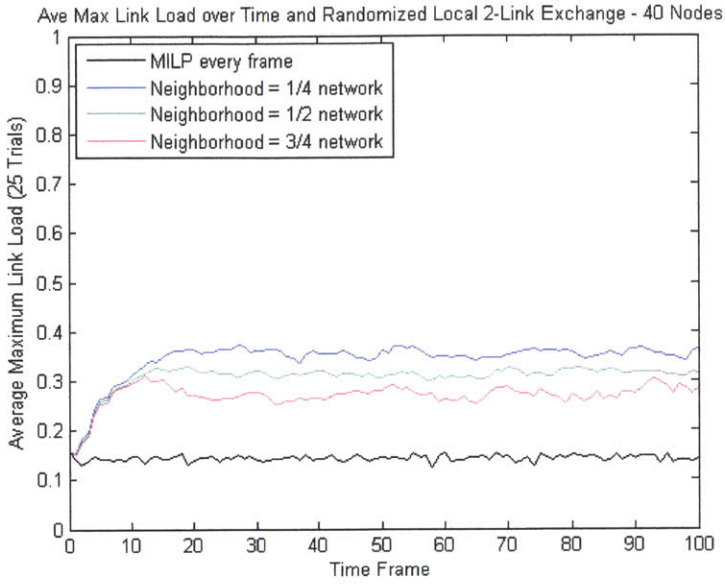

Figure 2-25: Maximum link load using the randomized local 2-link exchange algorithm for a 40-node network with varying neighborhood size compared to that obtained by optimizing according to the MILP formulation at every frame.

random. This idea drives the next algorithm.

Load-Based Local 2-Link Exchange Algorithm

Using the randomized local 2-link exchange algorithm, the probability that the most heavily loaded link is even eligible to be removed from the link activation set is only equal to the fraction of links in the control nodes neighborhood. However, if the control node after each time slot is immediately adjacent to the most heavily loaded link, then every time it will be eligible to be removed from the link activation set. The mechanics of this algorithm are otherwise identical with the randomized version. Out of all potential 2-link exchanges in some neighborhood of the control node, one is chosen that minimizes maximum link load for that neighborhood. This is repeated after each time frame of duration Δt elapses.

For the network with 40 nodes, results using this algorithm are compared with results from the randomized version in Figures 2-26 and 2-27.

Clearly, there is much improvement when the control node is selected based on

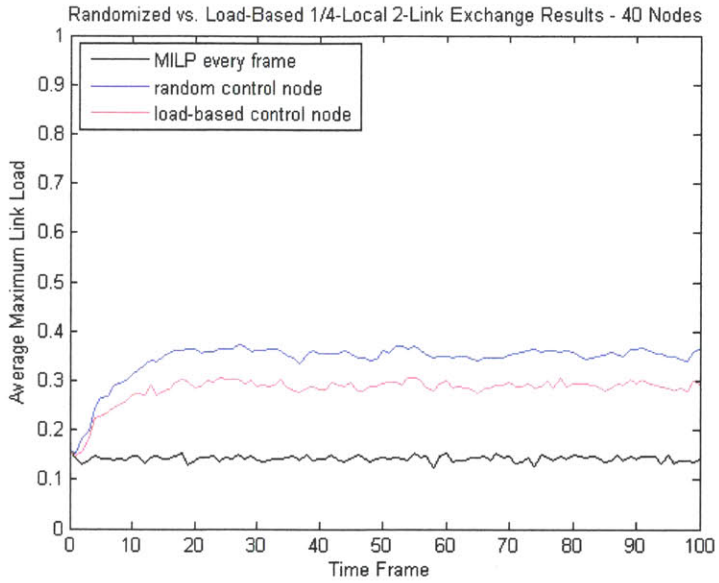

Figure 2-26: The maximum link load is compared using the MILP formulation after each frame, the randomized $\frac{1}{4}$-local 2-link exchange algorithm, and the load-based $\frac{1}{4}$-local 2-link exchange algorithm for the 40-node network. Results are averaged over 25 trials.

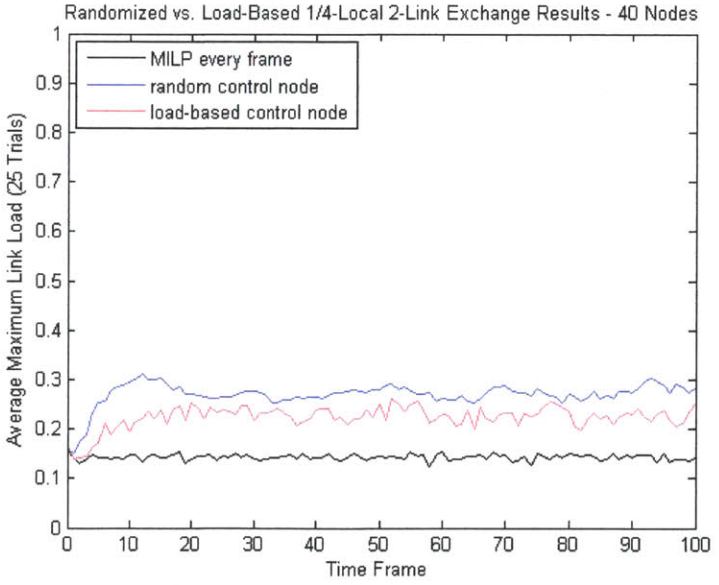

Figure 2-27: The maximum link load is compared using the MILP formulation after each frame, the randomized $\frac{3}{4}$-local 2-link exchange algorithm, and the load-based $\frac{3}{4}$-local 2-link exchange algorithm for the 40-node network. Results are averaged over 25 trials.

its proximity to the most heavily loaded link. The computation is the same as with the randomized local 2-link exchange algorithm, but the drawback still exists in not having enough flexibility to perform the number of exchanges needed to reach a performance level near that of the MILP solution.

Load-Based Local 3-Link Exchange Algorithm

One way to improve the amount of flexibility in potential new link activation sets is to allow for more expanded link exchanges. Up to this point, the link exchange algorithms have only employed 2-link exchanges. A 3-link exchange uses the same concept and is also featured in [5]: 3 links are chosen, and new links formed to reconnect the affected 6 nodes so that the result is a connected ring structure.

The only concern with the 3-link exchange algorithm as it compares to the 2-link exchange algorithm is computation, as more potential solutions exist. For a neighborhood of size v, each set of three links can be reconfigured in one of 7 ways, as seen in Figure 2-28. And there are as many as $\frac{v^3-v}{6}$ sets of 3 links in the neighborhood. But this is only a slight increase in computational complexity, at least compared to the approach that uses the MILP formulation to dictate a new link activation set after every time frame.

Since it has been demonstrated that the load-based method of selecting control nodes is superior to the random method, we apply the local 3-link exchange algorithm using the load-based method, and compare it to the 2-link exchange algorithm for the network with 40 nodes. Results are displayed in Figure 2-29.

Multiple Load-Based Local 3-Link Exchange Algorithm

The final improvement to these local link exchange algorithms is that instead of having one control node, we can designate multiple control nodes. In this algorithm, each control node acts the same as in the load-based local 3-link exchange algorithm. The difference is that we order the links based on their load, selecting a control node near the most heavily loaded link, and one near the second most heavily loaded link, and so on. Each node designates which old links to remove from the link activation

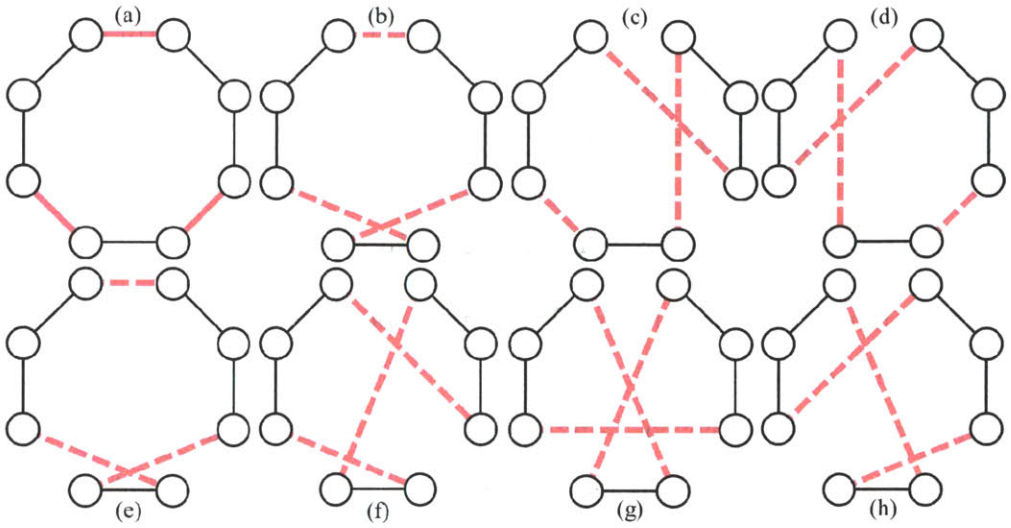

Figure 2-28: Given the original link activation set in (a), the three red links are selected for 3-link exchanges. The remaining diagrams are all valid 3-link exchanges for this set of three links.

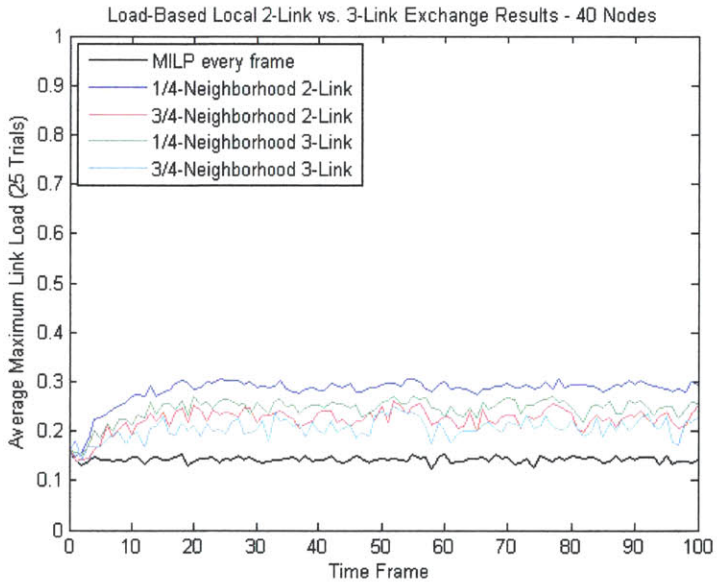

Figure 2-29: The maximum link load is compared using the MILP formulation after each frame, the load-based local 2-link exchange algorithm, and the load-based local 3-link exchange algorithm for the 40-node network for neighborhoods $\frac{1}{4}$ and $\frac{3}{4}$ the size of the network. Results are averaged over 25 trials.

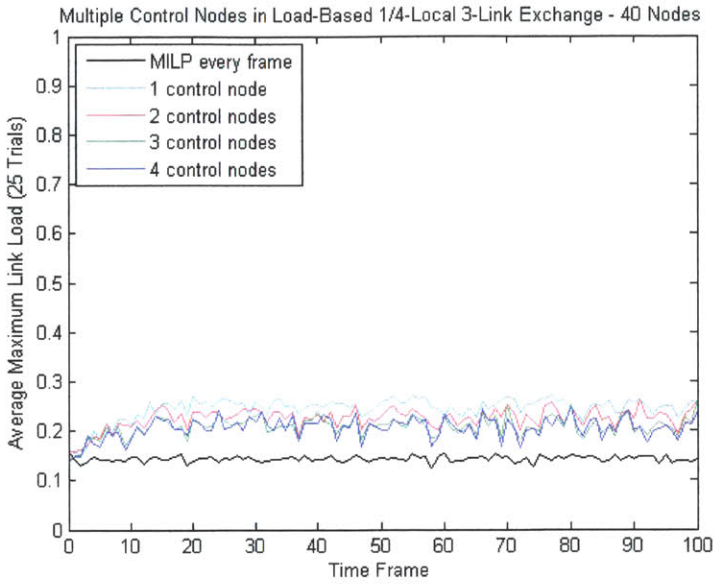

Figure 2-30: For different numbers of control nodes, the maximum link load resulting from the multiple load-based local 3-link exchange algorithm is compared with the original MILP approach and the single load-based local 3-link exchange algorithm in the 40-node network. For the local exchange algorithms, a neighborhood of size 20 is used.

set and which new ones to add to it. When conflicts arise as to the designation, a control node adjacent to a more heavily loaded link has preference. Because of this priority scheme, there is no limit to the number of control nodes. Essentially, if the link exchange operations determined by each marked node conflict, the one with higher priority is enforced. However, as the number of control nodes increases, so too does the number of new links being established at each time frame, which is one of the drawbacks of the original MILP formulation.

Figure 2-30 shows the results of the multiple load-based local 3-link exchange algorithm compared to previous methods in the 40-node network when the neighborhood size is fixed at $\frac{1}{2}$ of the network, or 20 nodes.

This appears to perform better than as the number of control nodes increases. As mentioned earlier, this can become problematic as each control node can call for up to 3 new links. Overall, out of all of the local link exchange algorithms, the multiple load-based local 3-link exchange algorithm performs the best, when compared with

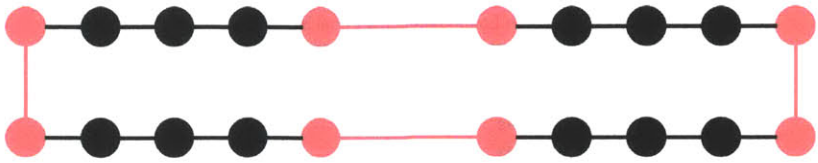

Figure 2-31: In the 20-node network, group endpoint methods reduce the number of nodes and links eligible to be removed or added. Nodes and links in black are fixed. The links in red can be removed and replaced by new links, provided there are the same number of links and the ring structure is left intact.

other algorithms using the same neighborhood size.

2.4.2 Group Endpoint Methods

One major reason it is impractical to regularly reconfigure the link activation set according to the MILP formulation is that the number of nodes makes computation prohibitive. One key to reduce computation is to limit the number of nodes to just the two nodes from each group that share a link with nodes outside the group. By ignoring links between nodes in the same group, the number of valid potential reconfigurations is dramatically reduced. After making this change, we can then choose to use the MILP formulation at each time frame, or any of the link exchange algorithms.

In the 20-node problem, instead of finding a set of 20 links to connect 20 nodes, only group endpoints are considered, of which there are 8. But by leaving links internal to a group unchanged, there are only 4 links that need to be activated. Figure 2-31 shows which nodes and links are eligible to be changed in group endpoint methods. Even the MILP approach becomes more reasonable as the number of potential solutions drops.

In the group exchange MILP approach, we use the same formulation, adding a constraint for each link connecting two nodes in the same group to be carried over to each subsequent solution. This is mathematically represented in Equation 2.13.

$$x_{ij} = x_{ij}, \forall i, j \in G, \forall G \in \boldsymbol{G} \tag{2.13}$$

And due to the reduced size of the problem, instead of local link exchange algo-

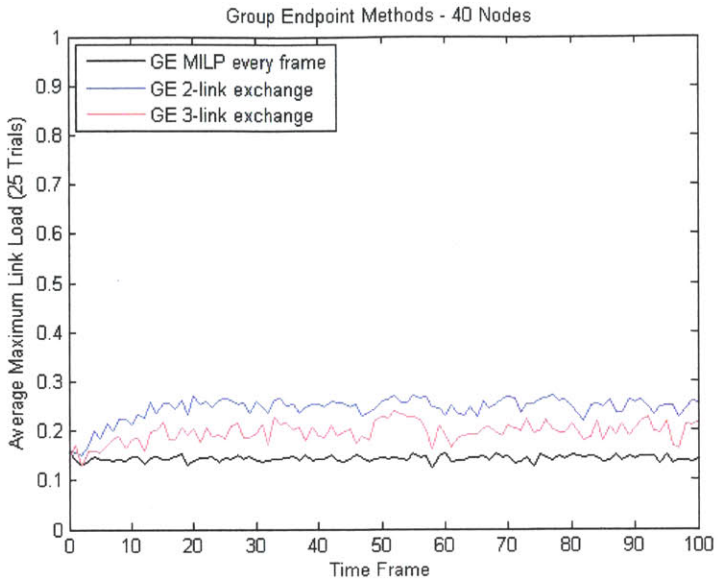

Figure 2-32: Results from applying the group endpoint versions of the original MILP, 2-link exchange algorithm, and 3-link exchange algorithm to the 40-node problem are compared. Values are averaged over 25 trials.

rithms, we only consider global link exchange algorithms with 2- and 3-link exchange operations.

In Figure 2-32, we compare results from the original MILP approach with results from the group endpoint approaches for the 40-node network.

The reason these methods are effective even when the set of eligible nodes is reduced owes to the ratio of distance between nodes in a flight group and the distance between different flight groups, as well as the ratio of intra-group traffic and inter-group traffic. The most heavily loaded links tend to be those between nodes in different flight groups because of the sometimes extreme distances they cover. For cases when more of the traffic remains within flight groups, other methods are anticipated to outperform the group endpoint methods.

2.4.3 Summary of Results

In this section, we have developed heuristic algorithms for topology management. In Table 2.4, we present the average maximum link load over 25 trials using each

Approach	Average Maximum Link Load 40 Nodes	60 Nodes	80 Nodes
MILP Every Frame	0.1442	0.1143	0.1013
MILP Every 5th Frame	0.2142	0.1934	0.1866
MILP Every 10th Frame	0.2707	0.2586	0.2593
Unchanged	0.5224	0.5952	0.6767
Randomized $\frac{1}{4}$-Local 2-Link	0.3726	0.4045	0.4214
Randomized $\frac{1}{2}$-Local 2-Link	0.3301	0.3324	0.3181
Randomized $\frac{3}{4}$-Local 2-Link	0.2895	0.2708	0.2564
Load-Based $\frac{1}{4}$-Local 2-Link	0.2891	0.2726	0.2612
Load-Based $\frac{1}{2}$-Local 2-Link	0.2635	0.2443	0.2305
Load-Based $\frac{3}{4}$-Local 2-Link	0.2425	0.2286	0.2102
Randomized $\frac{1}{4}$-Local 3-Link	0.3446	0.3755	0.3770
Randomized $\frac{1}{2}$-Local 3-Link	0.2722	0.2929	0.2748
Randomized $\frac{3}{4}$-Local 3-Link	0.2090	0.2034	0.1885
Load-Based $\frac{1}{4}$-Local 3-Link	0.2464	0.2589	0.2209
Load-Based $\frac{1}{2}$-Local 3-Link	0.2108	0.1998	0.1958
Load-Based $\frac{3}{4}$-Local 3-Link	0.1937	0.1731	0.1667
2 Controllers, Load-Based $\frac{1}{4}$-Local 3-Link	0.2237	0.2323	0.2156
3 Controllers, Load-Based $\frac{1}{4}$-Local 3-Link	0.2178	0.2068	0.1846
4 Controllers, Load-Based $\frac{1}{4}$-Local 3-Link	0.2126	0.1978	0.1699
Group Endpoint 2-Link	0.2452	0.2322	0.2266
Group Endpoint 3-Link	0.1872	0.1833	0.1484

Table 2.4: For each approach, the average value of the maximum link load over all time frames, averaged over 25 trials of each.

approach for each network size.

2.5 Summary and Conclusion

We have studied many approaches to topology management for airborne networks with flight group units and a limited number of transceivers per aircraft. Mixed integer linear programming is a computationally intensive method that results in optimal solutions. Although it is most likely impractical to use this operationally, it provides a baseline against which to compare other methods.

Distributed heuristic algorithms developed above provide adequate alternatives.

For its merits as a pure distributed algorithm in that only knowledge of links in a

Approach (Number of Nodes)	Random	Load-Based	% Improvement
$\frac{1}{4}$-Local 2-Link Exchange (40)	+0.2284	+0.1449	36.6
$\frac{1}{2}$-Local 2-Link Exchange (40)	+0.1859	+0.1193	35.8
$\frac{3}{4}$-Local 2-Link Exchange (40)	+0.1453	+0.0983	32.3
$\frac{1}{4}$-Local 2-Link Exchange (60)	+0.2902	+0.1583	45.5
$\frac{1}{2}$-Local 2-Link Exchange (60)	+0.2181	+0.1300	40.4
$\frac{3}{4}$-Local 2-Link Exchange (60)	+0.1565	+0.1143	26.9
$\frac{1}{4}$-Local 2-Link Exchange (80)	+0.3201	+0.1599	50.0
$\frac{1}{2}$-Local 2-Link Exchange (80)	+0.2168	+0.1292	40.4
$\frac{3}{4}$-Local 2-Link Exchange (80)	+0.1551	+0.1089	35.6

Table 2.5: Randomized and load-based local 2-link exchange algorithm results are compared in terms of average difference in maximum link load above MILP performance, with the improvement achieved by the load-based algorithm over the randomized version for comparable scenarios. Results are averaged over 25 trials.

local neighborhood is required, the randomized local 2-link exchange algorithm does not perform extremely well. The only difference in overhead between the randomized and load-based local link exchange algorithms is that global knowledge of the most heavily loaded link must be maintained. The benefit when using the load-based method of determining a control node is considerable. We compare in Table 2.5 the randomized and load-based versions of the algorithm. The percent improvement is calculated by determining the difference between maximum link load when using the MILP approach and the randomized local 2-link exchange algorithm, and finding what percentage of that difference is shaved off by using the load-based version.

The degree to which 3-link exchange algorithm outperforms the 2-link exchange algorithm is similarly demonstrable, and in Table 2.6, we present the improvement achieved by 3-link algorithms over their 2-link counterparts. The improvement is calculated the same way as in Table 2.5. One trend is that the improvement is more pronounced when the size of the neighborhood increases. This owes to the fact that the 3-link exchange algorithms outperform 2-link exchange algorithms when larger neighborhoods are considered, because then more link exchanges are available, In a small local neighborhood is considered, it might be common for only a 2-link exchange operation to be performed even when 3-link exchange operations are allowable.

When all factors are considered, the appropriate heuristic algorithm can be de-

Approach (Number of Nodes)	2-Link	3-Link	% Improvement
Load-Based $\frac{1}{4}$-Local Exchange (40)	+0.1449	+0.1022	29.5
Load-Based $\frac{1}{2}$-Local Exchange (40)	+0.1193	+0.0666	44.2
Load-Based $\frac{3}{4}$-Local Exchange (40)	+0.0983	+0.0495	49.6
Load-Based $\frac{1}{4}$-Local Exchange (60)	+0.1583	+0.1446	8.7
Load-Based $\frac{1}{2}$-Local Exchange (60)	+0.1300	+0.0855	34.2
Load-Based $\frac{3}{4}$-Local Exchange (60)	+0.1143	+0.0588	48.5
Load-Based $\frac{1}{4}$-Local Exchange (80)	+0.1599	+0.1196	25.2
Load-Based $\frac{1}{2}$-Local Exchange (80)	+0.1292	+0.0945	26.8
Load-Based $\frac{3}{4}$-Local Exchange (80)	+0.1089	+0.1667	39.9

Table 2.6: 2-link and 3-link load-based exchange algorithm results are compared in terms of average difference in maximum link load above MILP performance, with the improvement achieved by the 3-link algorithm over the 2-link version for comparable scenarios. Results are averaged over 25 trials.

termined best by computational capabilities, global awareness of real-time link loads, sizes of flight groups and relative difference between inter-group and intra-group traffic rates. Due to the inequality in traffic rates in this study, group endpoint methods perform extremely well, reducing the computational burden while allowing for global link exchanges to be performed. In other circumstances with very low rates of inter-group traffic, topology control might be performed best at the flight group level, in which the same principles would apply, leaving links in place that connect nodes in different groups and exchanging links only internal to a flight group. The key is to have an algorithm that focuses on links most likely to be heavily loaded.

Varying levels of performance can be achieved by adding layers of complexity to the algorithms, but each comes at a cost. Adding 3-link exchange operations increases computation. Increasing the size of the local neighborhood considered also increases computation, but also the need for awareness of load levels for links distant from the control nodes. And increasing the number of control nodes increases the number of new links at each time interval, which comes at a cost as the network forms the new links and re-routes traffic. But by balancing all of these factors, one can achieve a desired level of performance with a decreased computational burden as compared to the MILP approach.

Chapter 3

Ground Radios

3.1 Introduction

3.1.1 Background

The system to which this chapter applies is a ground tactical radio network. It is designed to be deployed in areas in which there is little or no infrastructure. While the radios may be mounted on mobile vehicles, the speed of such vehicles or the relatively small distance they are anticipated to cover allows us to treat the network as static. When nodes are within a specified range of each other, they can communicate. Unlike the network in Chapter 2, the links in this network have the same capacity. If two nodes are within transmission range and communicate, the link supports a certain rate. That rate does not increase or decrease if the distance between them changes. If two nodes are too distant to communicate directly, any traffic that needs to flow between them must be routed through other nodes.

Another difference between this system and that studied in Chapter 2 is that these ground radios can interfere with each other. If one radio transmits in a certain direction, and another radio is near enough in terms of distance and direction, the second radio might not be able to receive from any other radio because the first radio causes interference. Wireless ad hoc networks are subject to performance-limiting interference constraints because all transmissions share the same medium. These

constraints do not have nearly the same effect in wired networks because signals are isolated to a greater extent. Largely due to interference, it has been shown that the throughput of each user in multi-hop wireless networks decreases as the number of nodes sharing the channel increases under most conditions in a seminal paper by Gupta and Kumar [13], making scalability a vital concern for such networks. Regardless of the size of the network, it is desirable to achieve the highest possible rates of traffic flow.

In commercial voice and data networks, the introduction of cell towers is a major boon to performance, especially as they allow large numbers of users to connect to the network. These cell towers are essentially privileged nodes that can support higher rates of wireless traffic flow than normal users. This is because they can operate at higher power levels, being connected to the power grid. This is critical to support many wireless users. The cell towers are also connected to the high capacity wired backbone, so that instead of data being routed multiple wireless hops from source to destination, users gain direct access to the fiber optic infrastructure in a single hop to a cell tower. Traffic is then routed through the wired portion of the network to its destination, or to the cell tower nearest the ultimate wireless user.

Unfortunately, battlefield scenarios for which military-grade ground tactical radio networks are designed carry limitations. The need for mobility to transport equipment in generally remote locations off the power grid subject network components to size, weight and power (SWaP) constraints. Additionally, the lack of wired infrastructure limits communications to just wireless transmissions, which as mentioned have lower capacities due to interference. These constraints prevent the cell tower solution from answering the scalability problem in such networks.

Even so, we propose solving the scalability problem by introducing an advantaged node. Like a cell tower, this node does not source traffic, it only relays traffic. We refer to this advantaged node as the *relay node* and all other nodes as *ground nodes*. Also similar to a cell tower, the relay node is elevated on a portable pole or tower, giving it different transmission and interference properties from the ground nodes. We allow any ground node to transmit to the relay node, regardless of distance from

the ground node to the relay node. Such transmissions do not interfere with other ground nodes. When the relay node transmits, it interferes with all ground nodes so that no other transmissions may occur simultaneously. The relay node otherwise has the same capabilities as the ground nodes in that it transmits at the same rate and cannot receive multiple simultaneous transmissions.

3.1.2 General Approach

We test our proposed solution by comparing network performance with and without a relay node in various scenarios. These scenarios include ground nodes that transmit directionally, similar to systems like Harris' HNW [14] or Boeing's DNW [15]. The degree of directionality may differ, being modeled as an ultra-narrow beam as in Chapter 2, or being somewhat sectorized. Scenarios also include ground nodes that transmit omnidirectionally, similar to SRW [17] or Harris' ANW2 [16]. These scenarios also include all-to-all unicast traffic, in which each packet must travel from a single source node to a single destination node and every node sources traffic uniquely destined for every other ground node, and all-to-all broadcast traffic, in which all nodes source traffic and each packet generated by a source must be received by all other ground nodes.

A traffic *flow* consists of all data packets that share source and destination. Therefore, for unicast traffic, there is a flow associated with every source-destination pair, but for broadcast traffic, there is only one flow associated with every source node. The *flow rate* is the amount of traffic that can be generated and delivered for a given flow in a routing and scheduling solution. We evaluate the benefit of a relay node by maximizing the minimum flow rate when it is present and when it is absent. To maximize the minimum flow rate, we determine the greatest rate that all flows can simultaneously achieve. This ensures equitable allocation of resources. We also make a couple of assumptions. The first is that flow can be bifurcated: not all traffic sharing a source and destination needs to be identically routed. The other is that each link, including those between ground nodes and the relay node, has the same capacity: 1 unit of data per unit time when the link is active.

The problem of determining the greatest minimum flow rate can equivalently be solved by determining the least amount of time required to deliver 1 unit of traffic for each flow. The equivalence is based on the fact that if 1 unit of traffic for each flow can be delivered in a schedule of time length T, that schedule achieves a rate of $\frac{1}{T}$ for all flows. When it is easier to solve the greatest minimum flow rate problem, we can do so. When it is easier to solve the minimum schedule length problem we can also do so, and easily compare the results.

More frequently, we approach the problem as a minimum schedule length problem. Solving this problem to optimality requires solving the joint routing and scheduling problem, which can become quite complex. For this reason, we apply some techniques from literature, design greedy heuristic algorithms, and develop some novel techniques to find bounds on the minimum schedule length.

Our techniques assume binary or protocol interference models [13], in which two links either interfere or do not interfere with each other. This model states that for each node, i, there is a set of nodes, such that none of the nodes in the set can receive from a node other than i while node i transmits. This is different from additive or physical interference models, in which a node may be interfered with when nodes i and j are both transmitting, but the node can receive from other nodes when only one of nodes i and j is transmitting. The techniques presented can be applied to other similar problems that assume a protocol interference model.

The remainder of this chapter is outlined as follows. In Section 3.2, we describe how related work has influenced our approach and what distinguishes our problem and approach from the literature. In Section 3.3, we present techniques to find upper and lower bounds on minimum schedule length. In Section 3.4, we describe different scenarios in which a relay node might be used, and present the results of applying the procedures from Section 3.3 to evaluate the benefit of a relay node for networks of varying size. We conclude in Section 3.5 by characterizing the implications of our work.

3.2 Related Work

Much of the literature on wireless networks deals with efficient algorithms to find or approximate an optimal schedule [18] [19] [20] [21]. A major component of optimal scheduling is to determine *link activation sets*, or those sets of links that can be simultaneously activated because they do not interfere with one another [22] [23]. These have laid the groundwork for our approaches.

Badia et al. emphasize the importance of solving the routing and scheduling problem together, as separating the two problems often results in suboptimal performance [21]. They perform an exhaustive search over all routing and scheduling solutions to find a solution of shortest length. To reduce the number of potential solutions, they narrow their search to solutions with unit length time slots. That is, time is divided into equal intervals during which one unit of traffic can be transmitted across an active link. A link is either active for an entire slot, or idle for the entire slot. There are two reasons we depart from this exhaustive search approach. The first is that relaxing the unit length time slot constraint potentially allows for shorter schedules. The second drawback of this approach is that after implementing a mixed integer linear programming (MILP) formulation to solve for minimum schedule length, we determined that the approach is computationally too burdensome. It is impractical, especially as the size of the network grows, to expect to use such a formulation to solve a joint routing and scheduling problem to optimality using this approach.

An approach that avoids these concerns is presented by Jain et al. in [23]. They develop a technique for finding a lower bound on maximum throughput subject to traffic demands, based on determining the usage vector feasible region, where a *usage vector* contains the fraction of the schedule each link is active. Then a linear programming (LP) formulation solves for the optimal value of a linear function over a convex region. The formulation always results in a feasible solution, corresponding to an upper bound on minimum schedule length, and certain conditions guarantee optimality of the solution. Because this technique only apply for unicast traffic, we develop our own techniques to find upper bounds on minimum schedule length for

broadcast traffic, as well as techniques to find lower bounds for unicast and broadcast traffic.

3.3 Approaches for Finding Bounds on Minimum Schedule Length

In this section, we present techniques for finding upper and lower bounds on the minimum schedule length for delivering 1 unit of traffic for every flow for both unicast and broadcast traffic. The approaches to finding a lower bound vary based on the traffic pattern and presence of a relay node. The general principle behind finding an upper bound is to find a good feasible schedule, as the length of any feasible schedule is an upper bound on minimum schedule length. These approaches are applied later to quantify the relay node's benefit in multiple scenarios.

3.3.1 Bounding Methods for Unicast Traffic

To find an upper bound on minimum schedule length, we use the approach of Jain et al. to maximize network throughput. The resulting solution is always feasible, so the lower bound it provides on maximizing minimum flow rate corresponds to an upper bound on minimum schedule length, but under certain condition the bound is tight. We develop our own method to find a lower bound on minimum schedule length.

***Upper Bounding Method (Unicast)*:** We outline the approach of Jain et al., in which the authors maximize throughput for unicast traffic demands. By inputting unit traffic demands for all unicast traffic flows and setting the link capacity to 1 unit of flow per unit time, the greatest minimum flow rate is the inverse of minimum schedule length. This draws from the property that a schedule of length T in which 1 unit of every flow is delivered corresponds to a schedule in which all flows achieve a rate of $\frac{1}{T}$.

We first create a connectivity graph: any two nodes that can transmit to each other are connected by an arc in the connectivity graph. We then incorporate intereference

into the problem formulation by defining a conflict graph as follows: The number of nodes in the conflict graph equals the number of links in the original connectivity graph. If two links, l_1 and l_2 cannot be simultaneously active because the receiver of one is in the transmission range of the other, or because the two links share a node, then the corresponding conflict graph nodes, u_1 and u_2 are connected by an arc. The conflict graph allows us to determine activation sets, or sets of links that can be simultaneously activated because they do not interfere with one another.

A maximal independent set of nodes in the conflict graph is a set of nodes that share no arcs, but to which no nodes may be added to preserve independence. The maximal independent sets of nodes in the conflict graph correspond to maximal link activation sets in the original network. Since a usage vector is said to be schedulable if the corresponding links can be activated without conflict for the designated fraction of time.

Usage vectors and link activation sets can be represented in an L-dimensional space, where L is the number of links in the original network. The link activation set polytope is the convex hull in L-dimensional space defined by maximal independent set vectors. Any usage vector that lies in the activation set polytope is accordingly schedulable.

A linear programming formulation can then be used to maximize the objective function, which is the minimum flow rate, over the convex activation set polytope. Specifically, the objective function is:

$$\textbf{max } \alpha \tag{3.1}$$

In combination with the first constraint found below, this objective function maximizes the minimum rate of traffic coming out of source nodes.

$$\sum_{i \in \boldsymbol{R}(s)} f_{si}^{sd} \geq \alpha, \quad \forall (s,d) \tag{3.2}$$

In this formulation, f_{ij}^{sd} represents the rate of flow of data for source-destination pair (s,d) across link (i,j) and $\boldsymbol{R}(i)$ is the set of all nodes in transmission range of

i. The region over which the objective function is maximized is the region defined by the set of constraints that follow.

$$\sum_{j \in \boldsymbol{R}(i)} (f_{ij}^{sd} - f_{ji}^{sd}) = 0, \quad \forall (s,d), i \neq s,d \tag{3.3}$$

This second constraint only allows solutions in which, for every source-destination pair, the flow into and out of nodes other than the source and destination pair is always zero. This is a traditional flow balance equation.

$$\sum_{i} (f_{is}^{sd} + f_{di}^{sd}) = 0, \quad \forall (s,d) \tag{3.4}$$

The third constraint does not allow traffic to be routed to its source once it has already left, nor does it allow traffic to leave its destination once it has arrived. Without this constraint, traffic could flow in loops, being sent back to its source ad infinitum, just so that the source could in turn send the traffic out again. This would artificially inflate the objective function, showing that more original traffic leaves the source than it actually sources.

$$\sum_{(s,d)} f_{ij}^{sd} \leq \sum_{A:(i,j) \in A} \lambda_A, \quad \forall i,j \tag{3.5}$$

In this constraint, λ_A represents the fraction of time link activation set A is active. This fourth constraint limits the amount of flow across a link to the appropriate level determined by the amount of time the link is active.

$$\sum_{A \in \boldsymbol{A}} \lambda_A \leq 1 \tag{3.6}$$

In the fifth constraint, $\boldsymbol{A}$ is a collection of valid maximal link activation sets. This constraint requires that the usage vector $\boldsymbol{\lambda}$ lie in the link activation set polytope, since any usage vector that is a convex combination of link activation sets in $\boldsymbol{A}$ is schedulable. The final constraint allows for only non-negative f and λ variables:

$$f_{ij}^{sd} \geq 0, \quad \lambda_A \geq 0 \tag{3.7}$$

This LP can be solved rather quickly. Jain et al. show that when all maximal activation sets are in $\boldsymbol{A}$, this formulation is not only a lower bound on throughput, but that it solves for optimal network throughput. Performance thus hinges on the completeness of the collection of link activation sets. The problem of generating maximal link activation sets can be solved by creating a conflict graph wherein every node corresponds to a link in the original graph. An arc connects any two nodes in the conflict graph if their corresponding links interfere. A maximal link activation set is the same as a maximal independent set in the conflict graph. Moon and Moser showed the number of maximal independent sets to be bounded by $3^{n/3}$ and the Bron-Kerbosch algorithm enumerates all maximal independent sets in $O(3^{n/3})$ time [24]. For networks of the size presented here, exhaustive maximal non-interfering set enumeration is possible, but Jain et al. show that even without a complete set of maximal independent sets, the LP can result in an optimal or near-optimal solution. Compactly, the LP formulation is as follows:

$$\textbf{max } \alpha \tag{3.1}$$

Subject to:

$$\sum_{i \in \boldsymbol{R}(s)} f_{si}^{sd} \geq \alpha, \quad \forall (s,d) \tag{3.2}$$

$$\sum_{j \in \boldsymbol{R}(i)} (f_{ij}^{sd} - f_{ji}^{sd}) = 0, \quad \forall (s,d), i \neq s, d \tag{3.3}$$

$$\sum_{i} (f_{is}^{sd} + f_{di}^{sd}) = 0, \quad \forall (s,d) \tag{3.4}$$

$$\sum_{(s,d)} f_{ij}^{sd} \leq \sum_{A:(i,j) \in A} \lambda_A, \quad \forall i,j \tag{3.5}$$

$$\sum_{A \in \boldsymbol{A}} \lambda_A \leq 1 \tag{3.6}$$

$$f_{ij}^{sd} \geq 0, \quad \lambda_A \geq 0 \tag{3.7}$$

Lower Bounding Method (Unicast): We develop our own lower bounding

method. The method relies on the fact that it is sometimes desirable to route a traffic along a longer path if the links in the path interfere with fewer nodes than the links in a shortest path. We use the maximum link activation set sizes determined in the upper bounding method to assign costs to routing solutions. The cost of a routing solution can then be used to find a lower bound on all scheduling solutions associated with the routing solution. A minimum cost routing solution then leads to a lower bound on all scheduling solutions. The lower bound can be found by implementing the following procedure:

Procedure for Schedule Length Lower Bound (Unicast)

Step 1: Create a conflict graph. For each link (i, j) in the original graph, create a corresponding node in the conflict graph. Draw an arc in the conflict graph between any two nodes if the corresponding links in the original graph cannot be simultaneously activated. This can be because the receiver of one link is in the interference range of the other transmitter, or because the two links share a node.

Step 2: For each link (i, j) in the original graph, let m_{ij} be the maximum size of an independent set of which the corresponding node in the conflict graph is an element. There are a number of maximum independent set algorithms that can be used. This number, m_{ij}, is the maximum number of links (including link (i, j)) that can be active while link (i, j) is active.

Step 3: Assign a cost of $\frac{1}{m_{ij}}$ to each link (i, j) in the original graph.

Step 4: For each flow in the original graph, find the minimum cost route. The routing solution dictates the amount of time each link must be active in a corresponding scheduling solution.

Claim 1: The sum of the costs of the minimum cost routes for all source-destination pairs is a lower bound on minimum schedule length.

Proof. Consider a routing solution, $\boldsymbol{P}$, such that P_{ij} is the amount of time that link (i, j) must be active, with $S(\boldsymbol{P})$ the minimum schedule length corresponding to $\boldsymbol{P}$, and $C(\boldsymbol{P}) = \sum_{(i,j)} \frac{P_{ij}}{m_{ij}}$, the cost of the routing.

Suppose a schedule corresponding to $\boldsymbol{P}$ has length T. At any time instant $t \in [0, T]$, define $G(t)$ as the sum of the costs of the active links at t. There is some active

link, (i,j), such that no other active link at t has a cost greater than its cost, $\frac{1}{m_{ij}}$. From the definition of m_{ij}, no more than m_{ij} links are active at time t, and none has a cost greater than $\frac{1}{m_{ij}}$. Therefore $G(t) \leq 1$, $\forall t \in [0,T]$.

Since T is the length of a feasible schedule for $\boldsymbol{P}$, every link must be scheduled the amount of time dictated in $\boldsymbol{P}$, so $\int_0^T G(t)dt \geq C(\boldsymbol{P})$. $G(t) \leq 1$, so $T \geq C(\boldsymbol{P})$. Accordingly, $C(\boldsymbol{P}) \leq S(\boldsymbol{P})$, and if $\boldsymbol{P}^*$ is a minimum cost routing, then $C(\boldsymbol{P}^*) \leq C(\boldsymbol{P}) \leq S(\boldsymbol{P})$, $\forall \boldsymbol{P}$. □

3.3.2 Bounding Methods for Broadcast Traffic

The relay node is expected to provide greater benefit for broadcast traffic as compared to its benefit for unicast traffic. This is because each data packet must be received by all ground nodes, and a relay node can transmit to all ground nodes simultaneously. This is in contrast to unicast traffic, for which each data packet has only one destination, so when the relay node transmits, the signal is actively received by only one node and interferes with all others. For this reason, we develop different approaches to find bounds on minimum schedule length when there is a relay node and when there is not. To find upper bounds, we develop greedy heuristics that result in feasible routing and scheduling solutions. To find lower bounds, we present techniques based on finding the maximum number of transmissions that can occur simultaneously. As broadcast traffic is a special case of multicast, these approaches can be modified for multicast traffic.

Upper Bound without Relay (Broadcast): As any feasible routing and scheduling solution is an upper bound on minimum frame length, an upper bound can be obtained using a greedy heuristic algorithm to find a valid routing and scheduling solution. The basic idea behind our greedy heuristic algorithm is that a short schedule results by increasing the number of transmissions that occur at a time.

We apply the unit length time slot constraint in this algorithm: any transmission occurring at the beginning of a time slot continues until the end of the time slot. In any time slot, a node can only transmit data it has already received or that it generates.

We keep track of which nodes have received (or source) which packets with entries in an $n \times n$ matrix, $\boldsymbol{P}$. Initially, $P_{ij} = 0$ for $i \neq j$, and $P_{ii} = 1$ for all i. At the beginning of each time frame, we calculate a value, v_{ij} for every node i and packet with source j. If $P_{ij} = 0$, then $v_{ij} = 0$. Otherwise, $v_{ij} = \sum_{k \in \boldsymbol{R}(i)}(1 - P_{kj})$, where $\boldsymbol{R}(i)$ is the set of nodes that can receive a transmission from node i. This value is the number of nodes that can receive the packet sourced by j for the first time if node i transmits packet j. Select a node/packet pair that has maximum value, v_{ij}. Let A and B be the set of nodes actively transmitting and receiving, respectively, in a given time slot. At the beginning of a time slot these sets are empty. If (i^*, j^*) is the node/packet pair selected, then add i^* to A, and any node k to B if k is not already in A or B, is not in the interference range of any node in A other than i^*, is in $\boldsymbol{R}(i^*)$, and $P_{kj^*} = 0$. Next, if k just entered B, update $\boldsymbol{P}$ so that $P_{kj^*} = 1$. This process determines which node can transmit which packet and reach the greatest number of nodes that have yet to receive the packet.

For the same time slot, we repeat the process by determining values of v_{ij}, except that $v_{ij} = 0$ if i is in A or B, or if any node in B is in the interference range of i. Another change is that v_{ij} is now only the number of nodes in transmission range of i that have yet to receive node j and are not in interference range of any node in A for that slot. Once again, a node/packet pair with maximum value, v_{ij} is selected. This process continues until at some iteration, $v_{ij} = 0, \forall(i, j)$. Each iteration of this process determines which node can transmit which packet and reach the greatest number of nodes that have yet to receive the packet such that the transmission does not interfere with any previously selected receivers.

The entire process is repeated at the beginning of each time slot, with A and B empty. The algorithm terminates when $P_{ij} = 1$ for all pairs (i, j).

Lower Bound without Relay (Broadcast): We develop a lower bound by modifying the lower bounding approach for unicast traffic to allow for a transmission to be decoded by multiple receivers. One node can transmit, and any subset of nodes in its transmission range can receive the transmission. Depending on which subset of nodes receives, different sets of nodes may be able to transmit. This property is

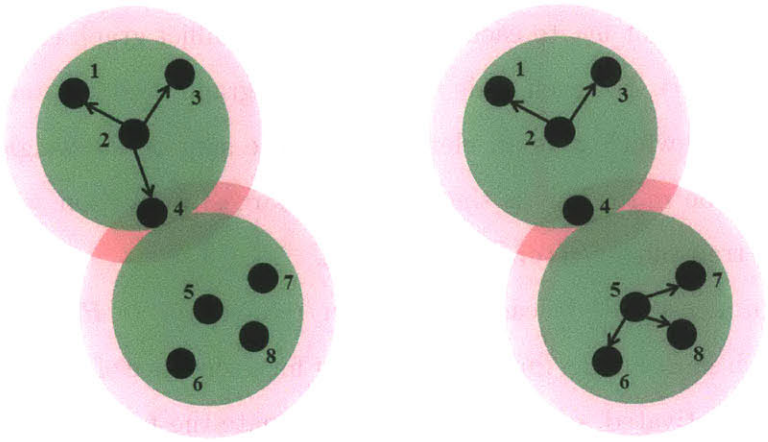

Figure 3-1: An illustration of subsets of receiving nodes. Green and red circles indicate transmission and interference ranges, respectively, for nodes 2 and 5. In (a), node 2 transmits to nodes 1, 3, and 4. Because node 4 is in the interference range of node 5, node 5 cannot transmit at all. In (b), the transmission from node 2 is no longer received by node 4. This allows node 5 to transmit to any or all nodes in its transmission range.

illustrated in Figure 3-1.

This procedure for obtaining a lower bound on schedule length is similar to that for unicast traffic, except that instead of a cost for every link, we assign a cost to every transmitter and each subset of nodes in its transmission range. The procedure below results in a lower bound:

Procedure for Broadcast Lower Bound without Relay

Step 1: For each ground node i, let $\boldsymbol{R}(i)$ be the set of all other ground nodes within transmission range of i, and let $\{\boldsymbol{R}_1(i), \boldsymbol{R}_2(i), ..., \boldsymbol{R}_n(i)\}$ be all unique non-empty subsets of $\boldsymbol{R}(i)$. Together, i with any subset of nodes within its transmission range form a *transmitter/receiver set*, $(i, \boldsymbol{R}_k(i))$ in that it is possible for i to transmit a packet and for it to be received by all nodes in $\boldsymbol{R}_k(i)$ at the same time.

Step 2: Create a conflict graph. For each transmitter/receiver set, $(i, \boldsymbol{R}_k(i))$, create a node, v_{ik} in the conflict graph. Draw an arc between any two nodes in the conflict graph if their corresponding transmitter/receiver sets cannot be activated simultaneously.

Step 3: Assign a weight w_{ik} to each node v_{ik} in the conflict graph equal to the number of nodes in $\boldsymbol{R}_k(i)$. For each node in the conflict graph, find the maximum weight independent set of which it is an element. Let the sum of the weights of such an independent set be m_{ik}. This is the size of the largest activation set to which the corresponding transmitter/receiver set belongs.

Step 4: Assign a cost, $\frac{w_{ik}}{m_{ik}}$, to each transmitter/receiver set, $(i, \boldsymbol{R}_k(i))$.

Step 5: For each source node, find minimum cost for a sequence of transmitter/receiver sets that, when activated, can completely disseminate the unit of traffic originating at the source.

Claim 2: The sum of the costs of all minimum cost sequences of transmitter/receiver sets for disseminating traffic originating at all nodes is a lower bound on minimum schedule length.

Proof. Consider a routing solution, $\boldsymbol{P}$, such that P_{ik} is the amount of time that transmission/receiver set $(i, \boldsymbol{R}_k(i))$ must be active, with $S(\boldsymbol{P})$ the minimum schedule length corresponding to $\boldsymbol{P}$, and $C(\boldsymbol{P}) = \sum_{(i,\boldsymbol{R}_k(i))} P_{ik}\frac{w_{ik}}{m_{ik}}$, the cost of the routing.

Consider a routing solution, $\boldsymbol{P}$, consisting of the sequence of transmissions to disseminate each flow's traffic, with $S(\boldsymbol{P})$ and $C(\boldsymbol{P})$ defined as before.

Suppose a schedule corresponding to $\boldsymbol{P}$ has length T. Define $G(t)$ for $t \in [0, T]$ as the sum of the costs of the active transmitter/receiver sets at t. Let the costs of the transmitter/receiver sets active at t be indexed $(\frac{w_1}{m_1}, \frac{w_2}{m_2}, ..., \frac{w_z}{m_z})$. Suppose $\tilde{m} = \min_{1\leq j\leq z} m_j$. Therefore, $G(t) = \sum_{j=1}^{z} \frac{w_j}{m_j} \leq \frac{1}{\tilde{m}} \sum_{j=1}^{z} w_j$. And the sum of the weights of active transmitter/receiver sets can be no greater than $\tilde{m}$ from the definition of m_{ik} and $\tilde{m}$, so $G(t) \leq \frac{1}{\tilde{m}} \sum_{j=1}^{z} w_j \leq 1$.

Since T is the length of a feasible schedule for $\boldsymbol{P}$, every transmitter/receiver set must be scheduled the amount of time dictated in $\boldsymbol{P}$, so $\int_0^T G(t)dt \geq C(\boldsymbol{P})$. $G(t) \leq 1$, so $T \geq C(\boldsymbol{P})$. Accordingly, $C(\boldsymbol{P}) \leq S(\boldsymbol{P})$, and if $\boldsymbol{P}^*$ is a minimum cost routing, then $C(\boldsymbol{P}^*) \leq C(\boldsymbol{P}) \leq S(\boldsymbol{P})$, $\forall \boldsymbol{P}$. □

Upper Bound with Relay (Broadcast): Again, any feasible routing and scheduling solution is an upper bound on minimum schedule length. This requires

determining how to flow 1 unit of traffic for each flow, and scheduling the link activations. Because the relay node is most beneficial when used to transmit data that would otherwise require a lot of time or link activations, we first calculate, for every flow, the minimum of time required for 1 unit of traffic to begin at its source and be received by all nodes without routing through the relay node and in the absence of any additional activity in the network. For each flow, f_i, let s_i be the time required for such delivery.

Let B be an ordered set of all flows, such that the first flow in B requires the minimum amount of time for delivery. Let A, A', and B' be initially empty ordered sets of flows. One by one, move flows from B to A, maintaining the order of flows, while $\sum_{i:f_i \in A} s_i \leq |B|$. Thus, all flows in A can be delivered without the relay node in the same amount of time required for flows in B to be transmitted from their respective source to the relay node.

Schedule the transmissions and link activations required for the first flow, f_{1^*}, in A to be received by all nodes in s_{1^*} time units. Also schedule during that time transmissions from nodes in B to the relay node when one can be simultaneously scheduled. When possible, the transmission from source to relay should be for the flow appearing last in B. Whenever 1 unit of traffic is received by the relay node, move the flow from B to B'.

During the s_{1^*} time units, also schedule any other transmissions that may simultaneously occur for other flows in A. At the end of the s_{1^*} time units, recalculate values of s for flows with transmitted data in the s_{1^*} time units, re-ordering A as necessary. Move any flow that has been received by all ground nodes from A to A'. Move the first flow in B to A whenever this can be done preserving the inequality: $\sum_{i:f_i \in A} s_i \leq |B|$.

Continue the process by activating links and routing traffic for the new first flow in A, while also simultaneously activating other links when possible, until, after t_1 time units, every flow is in A' or B'. All that remains is for the relay node to transmit data it has received in the first t_1 time units. The total schedule length is therefore $t_1 + |B'|$.

Lower Bound with Relay (Broadcast): Consider that a relay node can receive, transmit, or be idle. When it transmits, the maximum number of nodes receiving data is $n-1$. When it receives, there is some maximum number of ground nodes that can simultaneously receive data. Let that number be M. When the relay node is idle, at most $M+1$ nodes can simultaneously receive data. For all-to-all broadcast traffic, there are n flows, and each flow has $n-1$ ground node recipients. Suppose T^* is the minimum schedule length. The average number of active receivers during a schedule of length T^* is $\frac{n^2-n}{T^*}$.

The relay node can only transmit as often as it receives. Suppose α is the fraction of the schedule the relay node receives and the fraction of the schedule it transmits. The fraction of the schedule the relay node is idle is $1-2\alpha$. The greatest average number of active receivers is given by $\bar{r} = \max_{0\leq\alpha\leq 0.5} \alpha(M+n-1)+(1-\alpha)(M+1)$. Therefore $\max_{0\leq\alpha\leq 0.5} \alpha(M+n-1)+(1-\alpha)(M+1) \geq \frac{n^2-n}{T^*}$, implying that $T^* \geq \frac{n^2-n}{\bar{r}}$, a lower bound on minimum schedule length.

3.4 Application of Bounding Techniques to Example Networks

In this section, we apply the bounding techniques described in the previous section to example networks. When we know the optimal schedule length with and without a relay node, we can determine the precise benefit of the relay node. But in other cases, bound comparison gives insight into the relay node's benefit. The first scenario is a ring topology: each ground node can transmit to one of two near neighbors at a time without interfering with any other nodes. The second scenario is a regular grid mesh topology with omnidirectional transmissions: in the two-dimensional plane, a ground node can transmit to its nearest horizontal and vertical neighbors and interferes with is nearest diagonal neighbors. The last scenario is a random mesh topology: node locations are determined at random and each ground node can transmit to nodes within a fixed transmission range and interferes with nodes within a larger fixed

interference range. Examples of these topologies are illustrated in Figure 3-2.

As a way of accounting for progress in a routing and scheduling solution, we use the term *unit hop* to indicate either the amount of data transmitted or yet to be transmitted at a given time. If one unit of data travels one hop, it is considered 1 unit hop. Similarly, if $\frac{1}{2}$ unit of data travels two hops, it is also considered 1 unit hop.

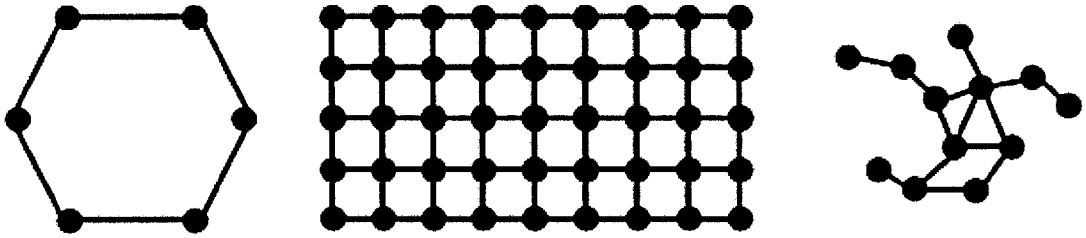

Figure 3-2: Examples of topologies studied. Arcs connect nodes if they can directly communicate. (a) A ring topology with 6 nodes. (b) A 5x9 regular grid mesh topology. (c) A general mesh topology.

3.4.1 Ring Topology

A ring is a rather simple network topology, and is the basis of the work in Chapter 2. Each node can transmit to one of two immediate neighbors, and the graph is fully connected. We model the ground antenna beams as being narrow so that a ground node can transmit to either of its two neighbors without causing interference. Similarly, only one node can receive any transmission from a ground node. The only interference constraint other than when the relay node transmits is that only one link can be active at a time for each node. One advantage of a ring topology is the simplicity of the routing problem, as there are only two paths between any two nodes without a relay node.

Unicast Traffic

For rings with up to 40 ground nodes and no relay node, the results of the bounding approaches are presented in Table 3.1. It is noteworthy that for each size, the upper

bound and lower bound are equal, showing that both are tight and the optimal schedule length.

Table 3.1: Ring Minimum Frame Length Bounds without Relay - Unicast

n	LB	UB	n	LB	UB
2	2	2	22	242	242
3	6	6	23	276	276
4	8	8	24	288	288
5	15	15	25	325	325
6	18	18	26	338	338
7	28	28	27	378	378
8	32	32	28	392	392
9	45	45	29	435	435
10	50	50	30	450	450
11	66	66	31	496	496
12	72	72	32	512	512
13	91	91	33	561	561
14	98	98	34	578	578
15	120	120	35	630	630
16	128	128	36	648	648
17	153	153	37	703	703
18	162	162	38	722	722
19	190	190	39	780	780
20	200	200	40	800	800
21	231	231			

Intuition behind the lower bound is straightforward. Without a relay node, when n is even, only $\frac{n}{2}$ links can be active at a time. And if each packet is routed to minimize the number of hops, the total number of unit hops required for all packets is $\frac{n^3}{4}$. Thus, the minimum schedule length cannot be less than $\frac{n^2}{2}$ for n even. Similar reasoning yields a lower bound of $\frac{n^2+n}{2}$ for n odd.

Intuition behind the upper bound requires slightly more elucidation. Without a relay node, when $n > 2$ is even, there are $n-1$ flows originating at each source. Figure 3-3 shows the multiple unicast flows sourced by one node. There is one unit of traffic from each source that must travel $\frac{n}{2}$ hops. We route these traffic units clockwise around the ring. In the space of 2 time units, it is possible for each clockwise link to be active one unit, or each counterclockwise link to be active one unit, which is illustrated in Figure 3-4. Initially, each node has 1 unit each of traffic needing to

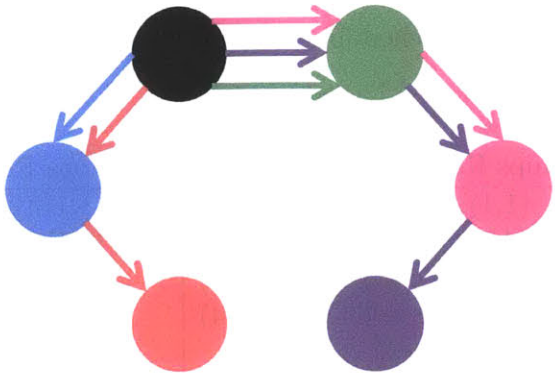

Figure 3-3: All unicast flows sourced by the black node. Each flow shares the color of its destination node. Each flow is routed according the path with the fewest hops, except for the purple flow. The purple node is 3 hops from the black node, either clockwise or counterclockwise. In this routing solution, the purple flow is routed clockwise.

Figure 3-4: Nodes are assigned even and odd numbers so that each link connects one odd and one even node. Nodes sharing a color and pattern can be activated simultaneously.

travel 1,2,...,$\frac{n}{2}$ units clockwise, and 1 unit each of traffic needing to travel 1,2,...,$\frac{n}{2}-1$ units counterclockwise that it needs to send or forward. After two units of time, each node can send one unit of traffic clockwise, specifically the traffic that has the furthest distance to travel. Table 3.2 shows a solution in which all traffic is delivered in $\frac{n^2}{2}$ time units. In this solution, first clockwise-routed traffic is all delivered, and then counterclockwise-routed traffic is delivered. Whenever a link is active, it transmits packets with the greatest number of remaining hops to its destination.

This feasible solution achieves the lower bound, so we know that both upper and lower bounds are tight for all ring sizes with $n > 2$ even. A similar exercise can be used to show that the lower bound of $\frac{n^2+n}{2}$ is achievable for all odd values of $n > 2$.

When a relay node is added to the directional ring network with unicast traffic, there is no change in the upper and lower bounds from those in Table 3.1. While

Table 3.2: Minimum Length Scheduling Solution for Ring (n even, Unicast Traffic)

Time Units Elapsed	Clockwise Packets at Each Node (# Hops Remaining, # Units)	Counterclockwise Packets at Each Node (# Hops Remaining, # Units)
0	(1,1),(2,1),...,($\frac{n}{2}$,1)	(1,1),(2,1),...,($\frac{n}{2}-1$,1)
2	(1,1),(2,1),...,($\frac{n}{2}-2$,1),($\frac{n}{2}-1$,2)	(1,1),(2,1),...,($\frac{n}{2}-1$,1)
6	(1,1),(2,1),...,($\frac{n}{2}-3$,1),($\frac{n}{2}-2$,3)	(1,1),(2,1),...,($\frac{n}{2}-1$,1)
12	(1,1),(2,1),...,($\frac{n}{2}-4$,1),($\frac{n}{2}-3$,4)	(1,1),(2,1),...,($\frac{n}{2}-1$,1)
$\vdots$	$\vdots$	$\vdots$
$\frac{n^2-2n}{8}-n$	(1,$\frac{n}{2}$)	(1,1),(2,1),...,($\frac{n}{2}-1$,1)
$\frac{n^2-2n}{8}$	0	(1,1),(2,1),...,($\frac{n}{2}-1$,1)
$\frac{n^2-2n}{8}+2$	0	(1,1),(2,1),...,($\frac{n}{2}-3$,1),($\frac{n}{2}-2$,2)
$\frac{n^2-2n}{8}+6$	0	(1,1),(2,1),...,($\frac{n}{2}-4$,1),($\frac{n}{2}-3$,3)
$\vdots$	$\vdots$	$\vdots$
$\frac{n^2}{2}-(n-2)$	0	(1,$\frac{n}{2}-1$)
$\frac{n^2}{2}$	0	0

initially surprising, the lack of benefit when a relay node is present is justifiable. For even values of $n > 2$, the solution just presented for the same network without a relay shows that there is a solution that achieves an average rate of $\frac{n}{2}$ unit-hops per unit time. And because no more than $\frac{n}{2}$ links may be simultaneously active for such a network, this is the constant rate of network-wide traffic flow. Consider a solution for the problem with a relay node. For accounting purposes, we say that a unit of traffic requires m unit-hops to reach its destination when the shortest path from its location to destination is m hops, excluding paths through the relay node. If a unit of traffic m hops from its destination is transmitted to the relay node, it becomes just 1 hop from its destination. Thus a hop to the relay node counts as $m-1$ hops. In the considered solution, the relay node transmits for some small amount of time, ϵ. Because the relay node sources no traffic, it must also receive for a duration of ϵ. During the time that the relay node receives, one of the ground nodes transmits to it, and there are $n-1$ remaining ground nodes to form $\frac{n}{2}-1$ links. The number of unit-hops advanced during that time of length ϵ is at most $2(\frac{n}{2}-1)\epsilon$, which is only the case if the traffic sent to the relay node still has $\frac{n}{2}$ hops to traverse. Then,

in the time that the relay node transmits, nothing else transmits, and the data it transmits traverses the remaining one hop to its destination, so the number of unit-hops in the time the relay node transmits is ϵ. Thus, the total number of unit-hops progressed in the time of length 2ϵ during which the relay node either receives or transmits is at most $(n-1)\epsilon$. But the best solution that does not use the relay node always advances $\frac{n}{2}$ unit-hops per unit time. Thus, the relay node does not improve the minimum schedule length for unicast traffic in the ring with an even number of ground nodes.

A similar exercise can be used to demonstrate that the relay node does not provide any benefit for unicast traffic in the ring with an odd number of ground nodes.

Broadcast Traffic

Without a relay node, each of the n ground nodes sources traffic that must be received by $n-1$ other nodes. Thus, the required number of unit-hops in a solution is n^2-n. No more than $\lfloor \frac{n}{2} \rfloor$ may be active at any time, so schedule length is lower bounded by $\frac{n^2-n}{\lfloor n/2 \rfloor}$, which is the lower bound achieved by the technique outlined in Section 3.3.2. The upper bound achieved by the technique outlined in Section 3.3.2 is given with the lower bound in Table 3.3.

The upper bound is a result of a greedy heuristic algorithm, so it does not with any degree of certainty gauge the tightness of either bound. However, it can be shown that the lower bound is achievable. As was seen in the unicast traffic case for the ring network with an even number of ground nodes, it is possible either for every node to transmit one unit of data counterclockwise or to transmit one unit of data counterclockwise over the space of 2 time units. If every packet is routed clockwise, then each packet needs to traverse $n-1$ hops in a feasible schedule. After two time units, each packet can traverse one unit, so each is $n-2$ hops from its ultimate destination. After four time units, each is $n-3$ hops from its ultimate destination. This can be continued until every packet has been delivered in $2n-2$ time units, for even values of n, which is the lower bound found earlier. In similar fashion, it can be shown that in a schedule of length $2n$, all packets can be successfully delivered in a

Table 3.3: Ring Minimum Frame Length Bounds without Relay - Broadcast

n	LB	UB	n	LB	UB
2	2	2	22	42	53
3	6	6	23	46	55
4	6	6	24	46	58
5	10	11	25	50	60
6	10	11	26	50	62
7	14	16	27	54	63
8	14	15	28	54	64
9	18	21	29	58	66
10	18	21	30	58	74
11	22	25	31	62	75
12	22	26	32	62	78
13	26	21	33	66	77
14	26	32	34	66	78
15	30	36	35	70	80
16	30	34	36	70	86
17	34	40	37	74	88
18	34	42	38	74	90
19	38	45	39	78	93
20	38	44	40	78	94
21	42	54			

ring network with an odd number of nodes.

In the presence of a relay node, the appropriate bounding techniques from Section 3.3.2 yield the bounds shown in Table 3.4.

In part because the approach to find the upper bound is a greedy heuristic algorithm, and also in part because it searches only for slotted solutions in which a link is either active or inactive for an entire length of one time unit, the upper bounds in Table 3.4 are, for the most part, sub-optimal. However, we can show that indeed the lower bounds are achievable, and specifically that for a network with n ground nodes and a relay node, a feasible schedule can be found that has length $\frac{2(n^2-n)}{\lfloor \frac{3(n-1)}{2} \rfloor}$.

Consider a network with an even number of ground nodes, as the network in Figure 3-5 with 8 ground nodes. It can be shown that by symmetry, the link from a node to its clockwise neighbor can be active $\frac{n-1}{2n}$ of the time, while each link can also transmit to the relay node $\frac{1}{2n}$ of the time, with the relay node transmitting $\frac{1}{2}$ of the time.

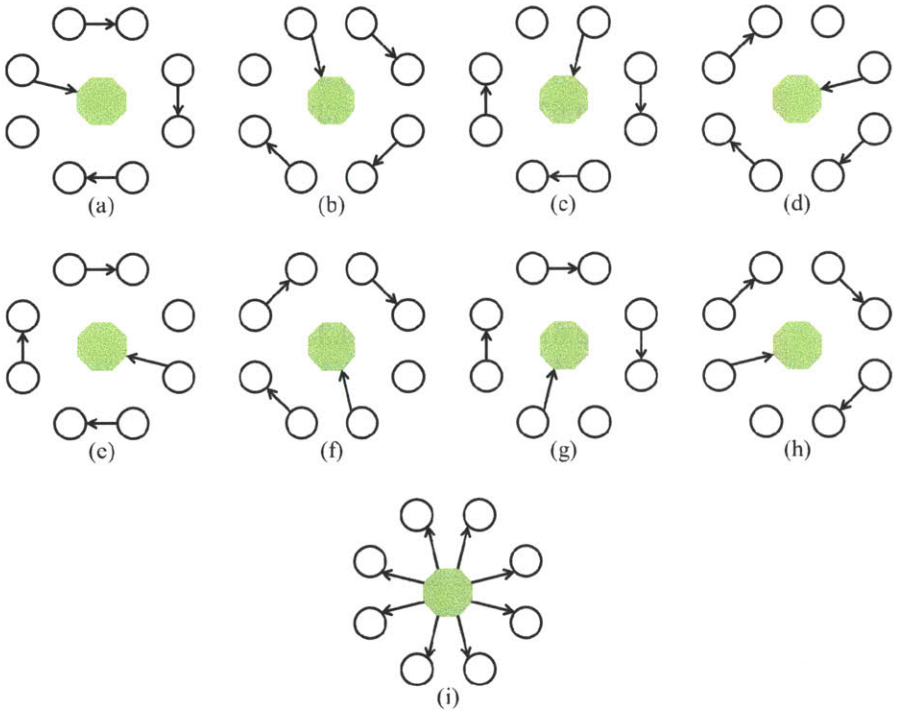

Figure 3-5: The black circles represent 8 ground nodes, while the green octagon is a relay node. The link activation sets in (a)-(h) are each active $\frac{1}{16}$ of the time, while the link activation set in (i) is active the other $\frac{1}{2}$ of the time. Each clockwise link is active the same overall fraction of time.

Table 3.4: Broadcast Ring w/ Relay Min Frame Length Bounds

n	LB	UB	n	LB	UB
2	2	2	22	29.81	42
3	4	4	23	30.67	44
4	6	6	24	32.47	46
5	6.67	8	25	33.33	48
6	8.57	10	26	35.13	50
7	9.33	12	27	36	52
8	11.2	14	28	37.8	54
9	12	16	29	38.67	56
10	13.85	18	30	40.47	58
11	14.67	20	31	41.33	60
12	16.5	22	32	43.13	62
13	17.33	24	33	44	64
14	19.16	26	34	45.80	66
15	20	36	28	46.67	68
16	21.82	30	36	48.46	70
17	22.67	32	37	49.33	72
18	24.48	34	38	51.13	74
19	25.33	36	39	52	76
20	27.14	38	40	53.79	78
21	28	40			

The schedule can be divided into time slots of duration $\frac{2}{3n-4}$. A different ground node transmits $\frac{2}{3n-4}$ units of traffic that it sources to the relay node during each of the first n slots. During the same n slots, each ground node also transmits traffic it sources to its clockwise neighbor during $\frac{n}{2}-1$ of the slots. This is illustrated in Figure 3-5. During the next n slots, the relay node broadcasts the data it received during the first n slots. These $2n$ slots constitute a frame. The frame is repeated. Each ground node still transmits $\frac{2}{3n-4}$ units of its own traffic to the relay node, but instead of transmitting its own traffic to the next clockwise neighbor, it forwards the traffic it had received previously from its counterclockwise neighbor. After $n-1$ frames, each node will have sent $\frac{n-2}{3n-4}$ units of traffic all the way around the ring, and $\frac{2n-2}{3n-4}$ units of traffic to the relay node, all of which will have been broadcast to the entire network. Thus, in $n-1$ frames of $2n$ slots each, with each slot having duration $\frac{2}{3n-4}$, for a total time of $\frac{4n^2-4n}{3n-4}$ units. For even values of n, this equals the lower bound of

$\frac{2(n^2-n)}{\lfloor\frac{3(n-1)}{2}\rfloor}$.

In a similar manner, for odd values of n, the schedule can be divided into time slots of duration $\frac{2}{3n-3}$. During the first n slots, each ground node transmits to the relay node in one of the slots. During the same n slots, each ground node transmits traffic it sources to its clockwise neighbor during $\frac{n-1}{2}$ of the slots. And in the second set of n slots, the relay node broadcasts traffic it received in the first n slots. Transmissions proceed as with n even, so that 1 unit of traffic from each source has been received by all nodes after $n-1$ frames of $2n$ slots each, with each slot having duration $\frac{2}{3n-3}$, for a total time of $\frac{4n^2-4n}{3n-4}$ units. This is equal to the lower bound.

As n increases, the minimum schedule length with a relay node of $\frac{n^2-n}{\lfloor n/2\rfloor}$ approaches $\frac{2}{3}$ the schedule length without it, or $\frac{2(n^2-n)}{\lfloor 3(n-1)/2\rfloor}$. The corresponding 50% improvement in minimum flow rate confirms the expectation for the ring topology with directional antennas that the relay node is more beneficial for broadcast traffic than for unicast traffic.

3.4.2 Regular Grid Mesh Topology

For the regular grid mesh topology, nodes are arranged in even rows and columns as seen in Figure 3-2. Row heights and column widths are 100 meters each. Antennas are omnidirectional with a transmission radius of 110 meters and an interference radius of 150 meters so that each node can transmit to its immediate horizontal and vertical neighbors, and in addition interferes with the closest diagonal neighbors. With omnidirectional antennas, a transmitter can send a data packet to any or all nodes in its transmission range simultaneously and interferes with all nodes in its interference range whenever active.

Unicast Traffic

Because they result from exhaustive maximal independent set enumeration, the upper bounds for unicast traffic are tight, and therefore more telling than the lower bounds. As the number of nodes increases, it becomes prohibitive to generate all maximal

independent sets. In such cases the upper bound would not be tight, and it is helpful to know how tight the lower bound is for reference. For that reason the lower bounds are also given in Table 3.5. In the networks presented, the relay improves performance by as much as 100% for the 2×2 grid and decreases as the size of the network increases, down to 16.7% for the 5×5 network.

Table 3.5: Regular Grid Minimum Schedule Length Bounds - Unicast Traffic

	With Relay		Without Relay	
Size	LB	UB	LB	UB
2×2	6	12	12	16
3×3	48.67	64	78	84
4×4	127.8	168	160.75	178.5
5×5	280.57	364	327.33	360.73

Broadcast Traffic

Neither the upper nor lower bounds are provably tight for broadcast traffic, so it is unclear exactly how much improvement can be gained from a relay node. But comparison of the minimum schedule length upper bound with a relay node to the lower bound without it in Table 3.6 shows that there is substantial improvement potential by using a relay node, as the least improvement out of the networks presented is in the 2×2 grid. In that case, the minimum schedule length with a relay node is at most $\frac{2}{3}$ the length of the schedule without a relay, corresponding to an improvement in throughput of at least 50%.

Table 3.6: Regular Grid Minimum Schedule Length Bounds - Broadcast Traffic

	With Relay		Without Relay	
Size	LB	UB	LB	UB
2×2	4.8	5.33	8	8
3×3	10.29	13	23.33	25
4×4	20.87	25.6	43.5	59
5×5	33.33	40.89	72.63	90

3.4.3 Random Mesh Topology

Just as with the regular grid mesh topology, each ground node in this section transmits omnidirectionally with transmission range of 110 meters and interference range of 150 meters. In each scenario, ground node locations are randomly distributed uniformly over a square 250 meters by 250 meters. For each size of network, from 2 to 20 ground nodes, we generated 10 scenarios of random nodes. If a scenario resulted in a disconnected graph without a relay node, we repeated the process such that all scenarios result in networks that are connected even without a relay node.

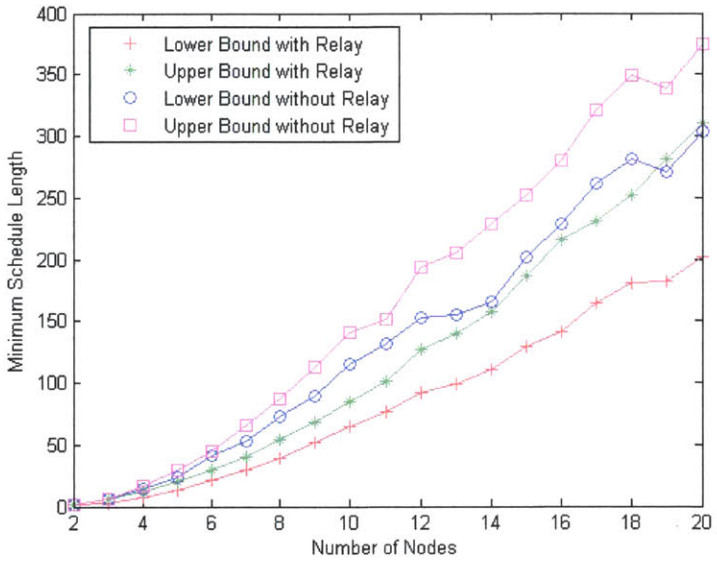

Figure 3-6: Average minimum schedule length bounds with random node placement with and without relay node for all-to-all unicast traffic in networks of varying size.

Unicast Traffic

Averaged over 10 scenarios for each network size, upper and lower bounds on minimum schedule length with and without a relay node for all-to-all unicast traffic are displayed in Figure 3-6. By including all maximal link activation sets, the approach to find upper bounds results in optimal schedule length. Average lower bound is displayed to demonstrate the tightness of the bound. For networks with between 2 and 20

nodes, the performance is on average 30.1% better with a relay node than without it.

Broadcast Traffic

Also averaged over 10 scenarios for each network size, upper and lower bounds on minimum schedule length with and without a relay node for all-to-all broadcast traffic are displayed in Figure 3-7. None of the approaches for bounding minimum schedule length for broadcast traffic are demonstrably tight, so the the guaranteed benefit of the relay node is found by comparing the schedule length upper bound with the relay node to the lower bound without it. This comparison demonstrates an average guaranteed improvement of 41% by adding a relay node. The figure also demonstrates that the bounding approaches result in reasonably tight bounds for these random mesh networks.

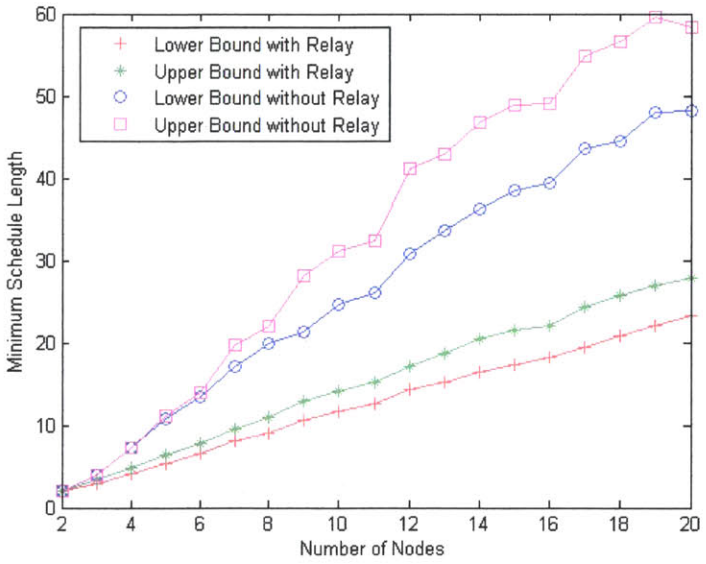

Figure 3-7: Average minimum schedule length bounds with random node placement with and without relay node for all-to-all broadcast traffic in networks of varying size.

3.5 Summary and Conclusion

The benefit of our proposal of an advantaged relay node to ground tactical radio networks has been demonstrated by comparing minimum schedule lengths, which correspond to achievable traffic flow rates. Because the difficulty in computing the optimal length of a schedule increases as the size of the network expands, we present methods to find upper and lower bounds on minimum schedule length. The computation of the procedures we have developed are more manageable than a MILP approach to optimization, and the gap between upper and lower bounds tends to be small enough to gain valuable insight into the problem.

These bounding techniques do well to quantify the benefit of a relay node and can be used to further investigate the effects of a relay node under different ground and relay node interference patterns and different traffic models. Such conversions to fit different models can be made with ease. In general, the benefit of a relay node is in reducing the number of hops that packets have to travel, which must be weighed against the interference it causes. As the net gain from a relay node is positive under several scenarios, it should garner serious consideration as a means to improve scalability in ground tactical radio networks.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Conclusion and Future Work

In this study, we examined two problems. In the first, we modeled a mobile airborne multi-hop radio network and used different methods of topology management. Dynamic link qualities necessitate a method of changing which links are active. The MILP formulation resulted in the best performance, but its high computational requirements are not realistic for real-time application in flight operations. The distributed heuristic algorithms require much less computation, which is their principle advantage over the MILP approach. By adding layers of sophistication to local link exchange algorithms, we have been able to obtain solutions that perform well when compared to the solutions obtained using the MILP formulation. Having a control node selected based on link loads results in better changes to the link activation set compared to a randomly selected control node, and the only cost is communicating the link loads. Adjusting the size of a local neighborhood considered for link exchange operations can balance the tradeoff between the need to communicate link loads, computation, and performance. Enlarging the number of potential new link activation sets comes at a computational cost, but again results in better performance. We have also shown that group endpoint methods are effective in reducing the number of potential solutions by ignoring links between nodes in the same flight group and concentrating only on those connecting nodes in different flight groups, as they are most likely to be heavily loaded. These methods perform well because of the traffic pattern. If instead, there were some other set of links likely to be the most heavily

loaded, as similar approach might be used to focus on making changes only to those links. The most significant result is that link exchange algorithms can be used to achieve performance comparable with that achieved using the MILP approach, but with a limited number of modifications to the link activation set and with much less computation.

In the second problem, a ground multi-hop radio network is limited in terms of throughput because the density of the nodes leads to interference. Our proposed solution is to add a relay node that is privileged in that its transmission and interference patterns differ from the other ground nodes. This relay node provides new paths for traffic to be routed. The specific properties that we use in our model may not represent with high fidelity those experienced in practice. However, the techniques for measuring its impact are easily modified.

Although the true impact of a relay node can only be understood by comparing optimal performance without the the relay node to optimal performance with it, the complexity of finding and verifying and optimal solution can be prohibitive. For this reason, we have developed and used techniques whereby we characterize the impact of a relay node by finding upper and lower bounds on optimal performance. These techniques perform well to find rather tight bounds. The novel techniques that we developed were used in conjunction with heuristic algorithms and an approach from literature to demonstrate that the relay node solution indeed improves throughput significantly in most of the models studied.

In the future, these techniques might be adopted and modified as needed to investigate the impact on throughput of one or more relay nodes in a network under different transmission patterns. While we have not studied the drawbacks of incorporating an elevated relay node in a ground tactical radio network, we expect, based on the gains of 30% to 40% and higher in throughput in most of the scenarios we studied, that net impact of a relay node is positive.

Bibliography

[1] Norman Polmar. *The Naval Institute Guide to the Ships and Aircraft of the U.S. Fleet.* Eighteenth Edition. Naval Institute Press, Annapolis, 2005.

[2] Dennis J. Baker, Anthony Ephremides, Julia A. Flynn, "The Design and Simulation of a Mobile Radio Network with Distributed Control." *IEEE Journal on Selected Areas in Communications*, 2(1):226-237, Jan. 1984.

[3] Anthony Ephremides, Jeffrey E. Wieseltheir, Dennis J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling." *Proceedings of the IEEE*, 75(1):56-73, Jan. 1987.

[4] Joseph A. Bannister, Luigi Fratta, Mario Gerla, "Topological Design of the Wavelength-Division Optical Network." *IEEE INFOCOM 1990*, 3:1005-1013, Jun. 1990.

[5] Aradhana Narula-Tam and Eytan Modiano, "Dynamic Load Balancing for WDM-based Packet Networks." *IEEE INFOCOM 2000*, 2:1010-1019, Jun. 2000.

[6] Amrinder S. Arora, Suresh Subramaniam, Hyeong-Ah Choi, "Logical Topology Design for Linear and Ring Optical Networks." *IEEE Journal on Selected Areas in Communications*, 20(1):62-74, Jan. 2002.

[7] Rajiv Ramaswami, Kumar N. Sivarajan, Design of Logical Topologies for Wavelength-Routed Optical Networks. *IEEE Journal on Selected Areas in Communications*, 14(5):840-851, Jun. 1996.

[8] Biswanath Mukherjee, "WDM-Based Local Lightwave Networks, Part II: Multihop Systems." *IEEE Network*, 6(4):20-32, Jul. 1992.

[9] Donald Gross and Carl M. Harris, *Fundamentals of Queueing Theory.* Wiley, New York, 1974.

[10] Craig M. Payne, *Principles of Naval Weapon Systems.* Second Edition. Naval Institute Press, Annapolis, 2010.

[11] David B. Johnson and David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks." *Mobile Computing*, 353:153-181, 1996.

[12] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols." *Mobile Computing and Newtorking*: 85-97, 1998.

[13] Piyush Gupta and P. R. Kumar,"The Capacity of Wireless Networks," *IEEE Transactions on Information Theory.* 2:388-404, 2000.

[14] Peter Griessler, J. B. Cain, Ryan Hanks, "Modeling Architecture for DTDMA Channel Access Protocol for Mobile Network Nodes Using Directional Antennas," *IEEE MILCOM 2007.* 1-6, 2007.

[15] Adam Baddeley, "Sky-borne Links," *Military Information Technology.* 14(7):13-16, 2010.

[16] Christopher G. Martin, "Progress in Tactical Wideband Waveform Development," *MilCIS 2010.*

[17] "Soldier-Level Inegrated Communications Environment (SLICE) Soldier Radio Waveform (SRW) Functional Description Document (FDD)," 1(3), 2003.

[18] Bruce Hajek and Galen Sasaki, "Link Scheduling in Polynomial Time," *IEEE Transactions on Information Theory.* 34:910-917, 1988.

[19] Rajiv Ramaswami and Keshab K. Parhi, "Distributed Scheduling of Broadcasts in a Radio Network," *INFOCOM '89.* 2:497-504, 1989.

[20] Olga Goussevskaia, Roger Wattenhofer, Magnus M. Halldorsson, Emo Welzl, "Capacity of Arbitrary Wireless Networks," *INFOCOM 2009.* 1872-1880, 2009.

[21] Leonardo Badia, Alessandro Erta, Luciano Lenzini, Michele Zorzi, "A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks," *IEEE Network.* 22:32-38, 2008.

[22] Murali Kodialam, Thyaga Nandagopal, "Characterizing Achievable Rates in Multi-hop Wireless Networks: The Joint Routing and Scheduling Problem," *MobiCom '05.* 42-54: 2005.

[23] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, Lili Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Wireless Networks.* 11:471-487,2005.

[24] Etsuji Tomita, Akira Tanaka, Haruhisa Takahashi, "The Worst-Case Time Complexity for Generating All Maximal Cliques and Computational Experiments," *Theoretical Computer Science.* 363(1):28-42, 2006.