ABSTRACT

Title of dissertation: PERFORMANCE ANALYSIS AND DESIGN
OF MOBILE AD-HOC NETWORKS

Senni Perumal,
Doctor of Philosophy, 2012

Dissertation directed by: Professor John S. Baras
Department of Electrical and
Computer Engineering

We look at the problem of modeling and designing a wireless mobile ad-hoc network (MANET) given certain performance specifications. We divide the mobile scenario into a series of time snapshots. At each snapshot, we assume that certain inputs like node topology information, traffic pattern between nodes, Medium Access Control (MAC) layer used and physical link propagation matrices are specified. Based on the type of MAC layer used, we develop reduced load performance models to estimate MAC layer throughput and delay. Once we can analyze a given network, we design the network to improve network performance including optimizing the total throughput and making a disconnected network connected via addition of Aerial Platforms. We assume that a source has multiple paths to transmit to its destination and we select the probabilities of routing the flows along these multiple paths to optimize the total network throughput. In order to optimize the total network throughput, we need to calculate throughput sensitivities with respect to the routing probabilities. Based on the type of performance models, we use various methods to calculate throughput sensitivities including Automatic

Differentiation and explicit formulas. We show an example of this modeling and design methodology using a realistic mobile wireless scenario for the 802.11 Distributed Coordination Function (DCF) MAC layer. We extend an already developed 802.11 MAC model with estimates for network delay and use Automatic Differentiation (AD) to calculate throughput sensitivities and optimize total network throughput.

We focus on the performance analysis and design of a wireless ad-hoc network using a virtual-circuit or reservation based medium access layer with the communication channel partitioned in time and frequency cells. In a reservation based MAC network, source nodes reserve a session's link capacity end-to-end over the entire path before sending traffic over the established path. Reservation based MAC protocols promise easy provisioning of quality of service and better utilization at high loads. An example of a generic reservation based MAC protocol is Unifying Slot Assignment Protocol (USAP). USAP is the distributed resource allocation protocol used in Rockwell Collins' tactical battlefield wireless ad-hoc protocol suite and is also adapted for use in the Joint Tactical Radio System. Any reservation based medium access protocol (including USAP) uses a simple set of rules to determine the cells or timeslots available at a node to reserve link capacity along the path to the next node. Given inputs of node locations, traffic pattern between nodes and link propagation matrices, we develop models to estimate blocking probability and throughput for reservation based wireless ad-hoc networks. These models are based on extending reduced load loss network models for a wireless network. For the generic USAP with multiple frequency channels, the key effect of multiuser interference on a link is modeled via reduced available link capacity where the effects of transmissions and receptions in the link neighborhood are modeled using USAP reservation rules. We compare our results with simulation and obtain good results using our extended reduced load loss network models but with reduced available link capacity distribution obtained by simulation. For the case of the generic USAP using a single frequency channel, we develop better models for unicast traffic using reduced load loss network models but with the sharing of the wireless medium between a node and its neighbors modeled by considering cliques of neighboring interfering links around a particular link and assuming that these cliques block independently when calculating the link blocking probabilities. We compare results of this model with simulation and show good match. We also develop models to calculate source-destination throughput for the reservation MAC (a modified form of USAP) as used in the Joint Tactical Radio System to support both unicast and multicast traffic. These models are also based on extending reduced load loss network models for wireless multicast traffic with the sharing of the wireless medium between a node and its (upto 2 hop) neighbors modeled by considering cliques of interfering nodes around a particular node and assuming that these cliques block independently. We compare results of this model with simulation and show good match with simulation. Once we have developed models to estimate throughput and blocking probabilities, we use these models to optimize total network throughput. In order to optimize total throughput, we compute throughput sensitivities of the reduced load loss network model using an implied cost formulation and use these sensitivities to choose the routing probabilities among multiple paths so that total network throughput is maximized.

In any network scenario, MANETs can get disconnected into clusters. As part of the MANET design problem, we look at the problem of establishing network connectivity and satisfying required traffic capacity between disconnected clusters by placing

a minimum number of advantaged high flying Aerial Platforms (APs) as relay nodes at appropriate places. We also extend the connectivity solution in order to make the network single AP survivable. The problem of providing both connectivity and required capacity between disconnected ground clusters (which contain nodes that can communicate directly with each other) is formulated as a summation-form clustering problem of the ground clusters with the APs along with inter-AP distance constraints that make the AP network connected and with complexity costs that take care of ground cluster to AP capacity constraints. The resultant clustering problem is solved using Deterministic Annealing in order to find (near) globally optimal solutions for the minimum number and locations of the APs to establish connectivity and provide required traffic capacity between disconnected clusters. The basic connectivity constraints are extended to include conditions that make the resultant network survivable to a single AP failure. In order to make the network single AP survivable, we extend the basic connectivity solution by adding another summation form constraint so that the AP network forms a biconnected network and also by making sure that each ground cluster is connected to atleast two APs. We establish the validity of our algorithms by comparing them with optimal exhaustive search algorithms and show that our algorithms are near-optimal for the problem of establishing connectivity between disconnected clusters.

PERFORMANCE ANALYSIS AND DESIGN OF MOBILE AD-HOC NETWORKS

by

Senni Perumal

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Ashok Agrawala
Professor Richard J. La
Professor Sennur Ulukus
Professor Raghu S. Raghavan, Dean's Representative

DEDICATION

To my father, mother, sister and brother.

ACKNOWLEDGMENTS

I am grateful to my advisor, Prof. John S. Baras, for his continuous support, his endless patience and encouragement during my graduate studies at the University of Maryland. His energy, enthusiasm, deep mathematical insight and expertise in different and varying topics have always been a constant source of motivation for me. The fact that he fosters a research environment free of constraints, where students are encouraged to explore different research areas and problems, allowed me to work on various problems that may not be part of this thesis, but introduced me to various mathematical tools and methodologies. This involvement helped me to build a concrete mathematical background and increased research maturity.

I am also grateful to my thesis committee members, Professor Ashok Agrawala, Professor Richard J. La, Professor Sennur Ulukus, and Professor Raghu S. Raghavan, for agreeing to serve on my committee. Professor La and Professor Agrawala provided me with very useful feedback during my thesis proposal examination.

During my graduate studies at the University of Maryland I had the privilege to take classes with some exceptional teachers. I would like to take this opportunity to thank Profs. Anthony Ephremedis, John Baras, P. S. Krishnaprasad, Mark Shayman, Babis Papadopoulos, Steven Tretter and Prakash Narayan. Their dedication to teaching and the extra effort they always made to organize and present the course material in an interesting manner made the task of learning more enjoyable. Their deep knowledge, experience, intuition, and graciousness towards their students had a profound impact.

The life of a graduate student involves dealing with bureaucratic issues from time to time. I would like to thank Kimberly Edwards, Althia Kirlew and Diane Hicks for their efficiency in keeping all these administrative details to an absolute minimum and in taking time to help me grow as a person. I would also like to thank the staff of both ECE and ISR for always trying to do their best helping students with official matters.

I would also like to thank a number of colleagues for their support and help during my graduate studies: Vahid Tabatabaee, Maben Rabi, George Papageorgious, Pedram Hovareshti, Punyaslok Purkayastha, Kiran Kumar Somasundaram, Kaustubh Jain, Vladimir Ivanov, Ion Matei, Ayan Roy-Chowdhury and Konstantinos Bitsakos.

Most and above all I would like to thank my family for their never-ending support and unconditional love.

This work was supported by a SBIR Phase I contract W15P7T-07-C-P402 from the U.S. Army CECOM to AIMS Inc.; by a SBIR Phase II contract W15P7T-07-C-P613 from the U.S. Army CECOM to AIMS Inc; by the Communication and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011; by NASA Marshall Space Flight Center under award no. NCC8-235; and by the DARPA NMS program.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or CECOM or NASA or the U.S. Government.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless Mobile Ad-hoc Networks (MANETs) are self-configuring mobile wireless communication networks where nodes form and maintain a wireless multi-hop network without any centralized infrastructure and control. Nodes in a MANET can act as both traffic sources and relays to forward traffic along multi-hop paths. MANETs are popular as a method of communication between nodes when the situation precludes the use of traditional fixed wireline networks or wireless cellular networks. Examples include military battlefield networks, mobile collaborative robotic networks and vehicular networks where nodes are mobile and there is no pre-existing communications infrastructure, disaster relief networks where pre-existing communication infrastructure has been wiped out, and sensor networks where it is not feasible (either financially, due to physical constraints, or due to mobility) to use wired networks. Recently wireless ad-hoc networks are becoming popular as a means of providing instant communication access to mobile users anywhere (for example, within a city, campus or office block) and anytime without the need to invest too much in fixed infrastructure. MANETs can also be used in conjunction with cellular wireless networks, where nodes close to each other can communicate directly amongst each other without going through any cellular towers. The increasing popularity of MANETs makes the performance modeling and design of such networks an important issue.

This chapter serves as an introduction to the rest of the thesis. In the next section we provide the motivation for the current work on MANET modeling and design. Section 1.2 introduces the problems that are addressed as well as the approach and tools used to solve these problems. Finally in section 1.3 we briefly go over the contributions of the thesis.

1.1 Motivation

Wireless mobile ad-hoc networks are characterized by mobile nodes communicating wirelessly with each other without any centralized control. This leads to a variety of issues, for example, how to schedule transmissions distributedly so that a node's transmission does not interference with those of other nodes in its vicinity, wireless channel fluctuations resulting in fluctuating link conditions, the requirement for a node to act as a router of traffic, the need to operate under energy constraints including heterogeneity in nodes' transmit power, mobility of users and environment conditions leading to changing network topology. Military MANETs involve additional constraints of survivability, robustness, need for prioritized communication, heterogeneity in terms of node capabilities (example, tanks vs soldiers vs aerial platforms), and the requirement of always being connected despite forming disconnected ground clusters. There is an interdependence of a node's performance on one another due to sharing of the wireless medium. Thus the characterization and design of MANETs is a non-trivial task. In fact, as of today, the performance of designed MANETs is unacceptable. There have been many actual experiments (military experiment in New Jersey with wireless nodes using OLSR for routing, 802.11 for the MAC and physical layers, and TCP/IP) where throughput at three hops falls to 20%-30% of maximum available ([1], slide 12), or where overhead in a wireless link reaches up to 98% ([2], slide 3). Most significantly there does not exist today a design environment, a systematic methodology, and a toolset to assist the engineer in designing MANET with predictable performance so as to meet requirements for a mission or more general use.

The Medium Access Control (MAC) layer in a wireless network is tasked with scheduling a node's transmission so as to prevent collisions with neighboring nodes' wireless transmissions and also to recover from unintended collisions. The broadcast nature of the wireless medium and the need for decentralized access to the wireless medium in a MANET as well as changing node locations implies that the MAC layer plays a very important part in determining the performance of a MANET. There are two basic mechanisms for multiple nodes to access a common channel: contention based access and reservation based or circuit-switched access. In a contention based access scheme, like 802.11 Distributed Coordination Function (DCF) [3], every node at each hop on the source-destination path contends with its neighboring transmitting nodes to obtain access to the communication channel. In contrast, in a reservation based or circuit-switched access scheme, like Unifying Slot Assignment Protocol (USAP) [4], a source node reserves the wireless channel along the entire route for some duration and then transmits data. Usually, in a reservation based access scheme, there is a separate control portion for exchange of control messages. Contention based access schemes are efficient in sharing resources for bursty traffic and may not need time synchronization among the nodes. The advantages of circuit-switched access schemes include easy provisioning of quality of service and better utilization at high loads. The reservation based USAP is used as the distributed resource allocation protocol in Rockwell Collins' tactical battlefield wireless ad-hoc protocol suite and is also adapted for use in the Joint Tactical Radio System (JTRS) [5]. While a lot of work has been done in modeling 802.11 DCF [3,6–10], there has not been any work done in modeling reservation-based USAP.

In order for nodes in a MANET to communicate with each other, the network has to be connected. There has been an enormous amount of research into wireless ad-hoc networks focusing on improving routing, scheduling, throughput, QoS, auto-configuration, etc.; and most of these works assume that the nodes form a connected network. But this assumption does not always hold true. Connectivity among nodes depends on several factors including node density, transmission power, transmitter and receiver characteristics, propagation path loss, node mobility, area of deployment, etc. Hence it is highly probable that a MANET has nodes that are disconnected from each other. Also in the case of military or disaster relief networks, the operational scenario may be such that there are disconnected clusters of nodes (where the nodes within a cluster can communicate with each other but the different clusters are too far apart to be connected, like clusters on opposite sides of a hill) but still there is need for communication between the different clusters. One of the methods suggested to improve connectivity, capacity, robustness, and survivability of MANETs is to use Aerial Platforms (APs) as relays in the network. While connectivity between different clusters can be bridged by adding additional ground resources to act as relays, this may not always be feasible due to the various factors including the nature of the terrain (steep hill or water body) or operational considerations. Furthermore, Aerial Platforms like Unmanned Aerial Vehicles (UAVs), helicopters, blimps, etc. have inherent advantages like rapid deployment to the theater of operation, range extension and beyond line of sight capability to provide connectivity between disconnected clusters, and the capability to support new services. Addition of a network of APs can enable a fragmented communication network consisting of disconnected clusters to become fully connected and provide the capacity needed for communication between the various disconnected source-destination pairs. Even in the case when all the nodes can communicate with other other (either through single or multiple hops), APs can be used to increase the capacity of the network by providing additional communication pathways between source-destination nodes and can potentially reduce delays in the ground network by reducing congestion in the ground network. The additional pathways through the APs also adds to the robustness and survivability of the ground network. In addition, a network of aerial platforms can also be used to reliably connect high priority nodes.

1.2 Problems Addressed and Approach

We first look at providing a systematic methodology and toolset for the analysis and design of MANETs to satisfy given performance criteria. We focus on the routing, MAC and physical layer. We model the various network components like the MAC protocol and physical layer via approximate reduced load models that link the various layers via a set of inter dependent equations that is valid for the entire network. These equations are then solved using fixed point iterations to obtain a set of consistent values for the performance metrics of interest like throughput, delay, and packet loss. In order to design a network to satisfy performance metrics, we rely on sensitivity analysis and optimization techniques for parameter tuning. Unfortunately, there may be no closed form expression for the sensitivities sought and in such cases, we use Automatic Differentiation (AD) to find the required sensitivities. Chapter 2 describes our modeling and design methodology in detail and applies it to the 802.11 DCF MAC protocol for a realistic scenario developed from a field exercise performed as part of the CBMANET project. We assume that we know the exogenous traffic rates for each source-destination pair and use multiple paths with a set of routing probabilities to forward traffic between a source and destination. We extend the 802.11 model developed in [10] to calculate the end-to-end delay assuming finite queues at each node. We use this model in our time-varying scenario to calculate performance metrics of throughput and delay. We then use this model to calculate throughput sensitivities with respect to routing probabilities using Automatic Differentiation and use these sensitivities to maximize total network throughput.

In chapter 3, we focus on the performance analysis and design of a wireless ad-hoc network using a virtual-circuit or reservation based medium access layer with the communication channel partitioned in time and frequency cells. In particular, we look at a generic reservation based MAC protocol called Unifying Slot Assignment Protocol (USAP). USAP [4] is the distributed resource allocation protocol used in Rockwell Collins' tactical battlefield wireless ad-hoc protocol suite and is also adapted to the Mobile Data Link (MDL) layer of the Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW) [5]. We develop performance models for USAP and use these models to calculate blocking probabilities for source-destination traffic as well as to optimize network performance. Our approach to modeling and optimization of reservation based protocols is based on fixed point methods and extending reduced load approximations for loss network models to a wireless network. Reduced load loss network models [11] were originally used to compute blocking probabilities in circuit switched networks. The main challenge in developing loss network models for wireless networks is the coupling between wireless links. We extend these loss models for wireless networks by looking at the sharing of channel capacity among neighboring nodes. We again assume that we know the exogenous traffic rate for each source-destination pair and use multiple paths with a set of routing probabilities to forward traffic between a source and destination. In order to calculate throughput sensitivities to maximize total throughput, we use an implied cost formulation [12]. The implied cost formulation used to calculate throughput sensitivities expresses the notion that when a call is admitted, it increases the current network throughput but also increases the blocking probability for future calls and hence reduces the future throughput (or implied cost).

USAP, as described in [4], is a generic distributed TDMA/FDMA slot assignment protocol for mobile multihop ad-hoc wireless network. The communication channel is divided into periodic frames and each frame is partitioned into orthogonal time-frequency cells with a portion of the cells used for management/control traffic and the rest used for user traffic. The generic USAP protocol operates in two modes: hard scheduling mode (virtual circuit connection-oriented mode) where nodes reserve a sessions link capacity end-to-end over the entire path; and soft scheduling mode (datagram scheduling) where nodes perform per-hop scheduling of links for single packets after the packets arrival at the node. We develop models for unicast traffic using USAP Hard Scheduling (which is basically a generic reservation protocol) where each frame has multiple frequency channels, and use these models to both approximate the performance of a MANET and to optimize network performance. We extend the reduced load loss network models to a wireless network by looking at the sharing of wireless medium between a node and its neighbors. This sharing of the wireless channel results in a node's available link capacity to be dependent on its neighborhood traffic. We model this effect via reduced available link capacity calculated using USAP reservation rules and traffic among neighboring nodes. We compare our results with simulation and obtain good results using our extended reduced load loss network models but with reduced available link capacity distribution obtained by simulation. We calculate throughput sensitivities to maximize total throughput by using an implied cost formulation.

We then look at USAP Hard Scheduling (as described in [4]) for the special/simpler case where each frame has only a single frequency channel. For this case, we develop better models to estimate the link blocking probabilities for the reduced load loss network approximation by direct analysis of the link neighborhood instead of first estimating the available link capacity pmf and then estimating the link blocking probability for each value of available link capacity. We model the performance of a single frequency channel virtual- circuit MAC protocol using reduced load loss network models extended to consider the sharing of the wireless medium by looking at cliques at a particular link. The blocking probability of a call at a particular link is calculated by considering cliques of neighboring interfering links that cannot transmit simultaneously and assuming that these cliques block independently. The neighboring interfering links and the conflict graph are calculated using the virtual- circuit reservation rules. We compare our results with simulation and show good match for large networks across various offered loads.

We then look at the case of USAP as used in the Mobile Data Link (MDL) layer of the Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW) [5]. JTRS uses USAP but modified for multicast in order to schedule transmissions so as to achieve contention free transmissions. We develop models for the MDL layer of JTRS and use these models to estimate route blocking probabilities for both multicast and unicast traffic. The models again use reduced load loss network approximations extended for a multicast wireless network with sharing of the wireless medium between a node and its (upto 2 hop) neighbors modeled (like in the previous case of USAP Hard Scheduling where each frame has only a single frequency channel) by considering cliques of interfering nodes around a particular node and assuming that these cliques block independently. We compare the results of our modeling with simulation and obtain good results using our extended reduced load loss network models.

In chapter 4, we look at the problem of topology control via dynamic addition of Aerial Platform (AP) nodes in order to connect a wireless ad-hoc network composed of disconnected clusters and also to improve the performance of such networks. While it is assumed that all the nodes in a wireless network form a connected network, this may not be true. We first look at the problem of providing (basic) connectivity between disconnected ground clusters by placing a minimum number of APs at appropriate places to act as relay nodes so that not only are the ground clusters connected to the APs but the APs are also connected to each other. Since Aerial Platforms are scarce and expensive resources, we want to find the minimum number of APs so that the network is connected. We pose the problem of connecting the ground clusters with the APs and the APs amongst themselves as two min-max constraints involving the maximum communication distances between the ground clusters and APs and between the APs respectively.

In order to avoid the computation complexity and difficulties of the *minimax* problems, we transform the min-max constraints into summation form clustering constraints using an approximation for the maximum of a number of variables. As the particular form of our constraints results in a non-convex optimization problem, we use Deterministic Annealing (DA) to avoid local minima and obtain near-optimal solutions. The connectivity between the ground clusters and the APs is the distortion cost to which is added the constraint for connectivity amongst the APs. We compare our connectivity results with an exhaustive grid search algorithm and show that the DA solution produces optimal results.

In addition to connecting the disconnected ground clusters via a minimum number of Aerial Platforms, we also look at placing these Aerial Platforms so that the traffic between ground clusters is supported by the AP to ground cluster links, i.e., there are enough pathways between the ground clusters and various APs to support the required inter-cluster capacity. In order to minimize the number of APs added, we maximize the sum of the AP to ground cluster link utilization. We solve this problem within the Deterministic Annealing framework by adding another cost term to the basic distortion cost (that relates the connectivity distance between the ground clusters and the APs); and by choosing the cluster prior probabilities to be proportional to the total capacity needed with other clusters. The additional cost function depends only on the assignment probabilities of the APs and is chosen so that the end resultant assignment probabilities of the APs are all almost equal to each other. The Deterministic Annealing solution results in the minimum number of APs placed so that the capacity out of each cluster is supported by the associated APs and with the used link capacity of each AP to its associated clusters being almost equal (the solution is load balanced).

We then extend the basic connectivity solution in order to make the resultant network single AP survivable by ensuring that each ground cluster is connected to atleast two Aerial Platforms and by making the AP network biconnected. We make sure that each ground cluster is connected to atleast two APs by modifying the highest two association probabilities of a cluster to the APs during each iteration of the DA algorithm and forcing them to be equal so that each cluster is connected to atleast two APs. We make sure the AP network is biconnected by adding another constraint to the Deterministic Annealing formulation that relates the distance between an AP and its second closest AP and thus make sure that each AP is connected to atleast two previously added APs resulting in an AP network that is biconnected. Finally we propose a heuristic algorithm to connect high-priority clusters and a maximum number of non-priority clusters when the number of available APs is less than the minimum required to connect all the ground clusters.

1.3 Contributions of the Thesis

Wireless Mobile Ad-hoc Networks (MANETs) are increasingly in popularity. The performance modeling and design of such networks is a non-trivial task due to the interdependence of design parameters on one another. We first look at providing a systematic methodology to performance modeling and design of such networks. We divide the scenario into multiple time snapshots and calculate performance metrics and parameter sensitivities (used for design) for each snapshot. Our approach is based on using approximate reduced load models that link the various layers via a set of inter dependent equations that is valid for the entire network. These equations are then solved using fixed point iterations to obtain a set of consistent values for the performance metrics of interest like throughput, delay, and packet loss. We apply this methodology to a 802.11 DCF model as described in [10] on a realistic scenario derived from field experiments performed as part of the CBMANET project. For the 802.11 DCF model, we derive equations for the end-to-end delay.

We develop various models to estimate throughput and blocking probability for source-destination connections using reservation based virtual-circuit MAC protocols. We also use these models to calculate throughput sensitivities in order to maximize total network throughput. Unifying Slot Assignment Protocol as described in [4] divides the communication channel into periodic frames and each frame is partitioned into orthogonal time-frequency cells with a portion of the cells used for management/control traffic and the rest used for user traffic. USAP uses a generic set of reservation rules in order to reserve end-to-end connections. We develop reduced load loss network models to estimate the blocking probability of each source-destination unicast connection for a network using USAP with each frame divided into multiple frequency channels. The sharing of the wireless channel between a link and its neighboring links is modeled by reduced available link capacity which is calculated using the USAP reservation rules and the traffic among neighboring nodes. We show the validity of this approach of using a pmf on the available link capacity by matching results with simulation. The throughput sensitivities are calculated using an implied cost formulation and are used to maximize total throughput. For the special case of USAP with each frame having only a single frequency channel, we develop better models to estimate the link blocking probabilities for the reduced load loss network approximation by analysis of the link neighborhood directly using cliques instead of first estimating the available link capacity pmf and then estimating the link blocking probability for each value of available link capacity. We show that this method matches simulation results very well. We also develop models for USAP as used in the Mobile Data Link (MDL) layer of the Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW). These models estimate throughput for both multicast and unicast traffic. We show that the approach of using reduced load loss network models extended to a wireless network for multicast traffic with sharing of the wireless medium between a node and its (upto 2 hop) neighbors modeled via cliques matches well with simulation. The models developed for the various flavors of USAP and the methodology for maximization of total network throughput require knowledge of the entire network but run extremely fast (compared to simulation) and could perhaps be used as part of an online design tool using a rolling horizon look-ahead window so as to optimize total network traffic real-time for a MANET.

MANETs are not always connected and in some situations there is need for these disconnected clusters to be connected to one another. We develop fast almost optimal algorithms using Deterministic Annealing (DA) for placement of a minimum number of Aerial Platforms (APs) in order to connect a wireless network composed of disconnected clusters. We further develop DA algorithms to place APs so that the the connected network remains connected even if a single AP fails as well as an extension to make sure that the connected ground and AP network is such that the traffic between ground clusters is supported by the AP to ground cluster links.

Chapter 2

Performance Modeling and Design Methodology for MANETs

2.1 Introduction

Despite recent interest and progress in multi-hop wireless networks, we still lack systematic methodologies and tools that would allow for the efficient design and dimensioning of such networks with the provision of accurate performance bounds. The main reason for this is the different nature of wired and wireless networks. Key quantities, such as the link capacity, that remain constant in a wired network, vary in wireless communication environments with the transmission power, the interference, the node mobility and the channel condition.

It is possible to develop packet level simulation models of the physical (PHY) layer and the medium access control (MAC) protocol using various software packages. However the packet level simulation of multi-hop wireless networks with the appropriate PHY and MAC layer modeling turns out to be too complex and time consuming for the design and analysis of wireless networks in realistic settings. Our objective is to develop low complexity combined analytical and computational (numerical) models, which can efficiently *approximate* the performance of wireless networks. Such models have several applications in the design and analysis of wireless networks.

Our alternative approach is based on fixed point methods and reduced load network models for performance evaluation and optimization. Reduced load loss network models [11] were originally used to compute blocking probabilities in circuit switched networks [13] and later were extended to model and design ATM networks [14–17]. In [17] reduced load approximations were used effectively to evaluate quite complex ATM networks, with complex and adaptive routing protocols, and multi-service multi-rate traffic (different service requirements). The main challenge in developing reduced load network models for wireless networks is the coupling between wireless links. This coupling is due to the transmission interference between different nodes in proximity with each other. Reduced load models have been developed for contention based 802.11 DCF wireless networks [8, 10]. Once we have developed good performance models, we optimize the network through the calculation of performance metric sensitivities. We use various methods including explicit formulas or Automatic Differentiation (AD) for sensitivity analysis.

We assume we know the exogenous traffic rate for each source-destination pair and use multiple paths with a set of routing probabilities to forward traffic between a source and destination. In our approach, we model routing, MAC and PHY layers of the network architecture. We formulate a set of equations modeling the implicit dependencies between the various layers of the protocol stack. With this implicit model we are able to recover network performance parameters such as throughput and delay. Once we develop good approximate performance models, we calculate throughput sensitivities with respect to the routing probabilities and use them to maximize total network throughput by computing the optimal load distribution among multiple paths of a source-destination connection.

This chapter is organized as follows. The next section describes our MANET modeling and design methodology in detail along with the system's possible inputs and outputs. Section 2.3 describes a realistic scenario developed from traces of a field experiment performed as part of the CBMANET project. We use this time-varying scenario to generate results and validate developed models. In section 2.4, we apply the modeling and design methodology to the random access 802.11 Distributed Coordination Function (DCF) protocol. In section 2.4, we briefly present the throughput model for the 802.11 DCF MAC as developed in [10], describe the delay analysis developed for this model, discuss how we use Automatic Differentiation for performance metric sensitivity computations and finally provide results of running the 802.11 model and design framework for the scenario described in section 2.3.

2.2 Modeling and Design Methodology

The proposed approach is to formulate the network design problem as a multi-objective constrained optimization problem using performance estimation network models. The approach is shown schematically in Figure 2.1. The various elements in the figure are described in the following paragraphs. The inputs to the design problem include the specific network topology, traffic patterns, network conditions and any constraints that we want to satisfy. The outputs include values for various MANET design parameters, suggestions for topology control like additions of advantaged nodes (e.g., Aerial Platforms), QoS/performance metric estimates like throughput, delay and packet loss, as well as tradeoff curves between various design parameters. Section 2.2.1 describes the possible inputs and outputs to the design methodology in detail.

We first develop models (the box *Performance estimation models* in figure 2.1)

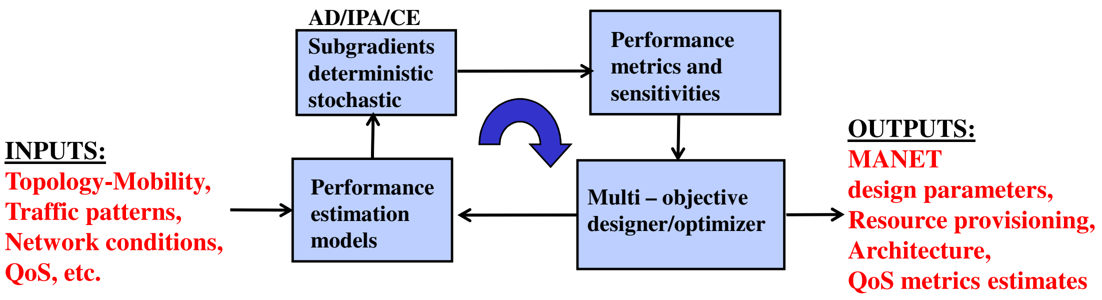

Figure 2.1: Proposed Design Methodology

of various wireless protocol layers like the routing, medium access control (MAC), and physical layers that allow us to efficiently approximate its behavior. Each layer is modeled as a set of equations that link that layer's operation through specific layer-dependent and cross-layer performance parameters with the specified inputs and output performance metrics like throughput, delay, etc. Losses in a wireless network are primarily due to the physical wireless medium and interference between adjacent nodes. These losses are taken care of at each layer via loss parameters that reduce the traffic as it is forwarded through nodes. Thus these models can be thought of reduced load performance estimation models. Fixed point iterations are then applied to the resultant set of indirectly linked equations (that approximate each layer) in order to obtain implicitly defined relations between the layer-dependent parameters, the cross layer parameters, and the output performance metrics (throughput, delay, packet loss). These equations are iterated until they yield a consistent set of solutions for the output performance metrics of the whole network under consideration. This method is very powerful and has been used very successfully for complex wireline networks ([11, 13–17]). It allows for different modules to be substituted in to develop different implicit performance models. A library of such models of varying complexity can be developed that will then allow the engineer to design a network to satisfy his quality of service constraints. The main challenge in applying such models to wireless networks is the coupling between wireless users. Coupling is due to the interference caused by transmissions by nodes in proximity with each other. By using probabilistic physical and MAC layer models, interference and contention can be approximated as inter-link traffic dependent loss factors. This approach has been successfully used to model the 802.11 Distributed Coordination Function (DCF) MAC in [8] and [10].

Once we have developed these performance estimation models, we can then perform tradeoff analysis and optimization using various multi-criteria constrained optimization methods. This is shown schematically in Figure 2.1. Most of these methods rely on calculating sensitivities of the performance metrics of interest with respect to system parameters. If it is analytically possible to derive the sensitivities, we can use them. But usually the fixed point method provides a computational scheme that, after convergence, describes the performance metric (e.g, throughput) as an implicit function of the design parameters (e.g., routing parameters). Thus, we may not have analytic expressions of the performance metric evaluations, but instead, we have a program that computes the values of the performance metric, while implicitly providing the dependence of the values on the design parameters. We then use Automatic Differentiation (AD) to compute the gradients.

AD is a numerical method to compute the derivatives of a program [18]. Using the fact that a computer program is in fact a sequence of primary operations, automatic differentiation records the relationships between them and using the chain rule, it is able to provide the derivative of a function in a short amount of time. We use ADOL-C (Automatic

Differentiation by OverLoading in C++), the source code of which is available at [19] to implement automatic differentiation on our fixed point model. Operator Overloading consists of changing the type of the variables involved in the computation to a proprietary type given by the Automatic Differentiation tool to allow it to compute derivatives based on its linked libraries.

A given MANET scenario is divided into multiple time snapshots with the modeling and design methodology applied to each snapshot. The performance approximations and optimizations performed at each snapshot are only valid when the system attains equilibrium. We assume that the conditions in a given snapshot stay constant for enough time that the system attains (near) equilibrium before the next snapshot. The time snapshots chosen can be either periodic or aperiodic (based on significant/threshold change in inputs). The MANET design problem at each snapshot is decomposed into several key and interrelated components that lead to a modular design environment where various modules can be easily incorporated. The broad steps involve:

1. Dynamic computation and maintenance of neighborhoods. Mathematically this involves characterizing and maintaining up to date information of the Adjacency Matrix, $A(t)$, where $A_{ij}(t)$ is 1 if node j can hear node i and 0 otherwise. These values depend on the environmental conditions and the power used by the nodes.

2. Algorithms for computing various types of paths (k-shortest paths, k-disjoint paths, paths that form connected dominating sets) between designated sets of origin-destination nodes. The inputs to the algorithms are the adjacency matrix and source-destination pairs and the outputs are multiple paths between each source-destination pair.

3. Assignment of performance metrics/QoS to paths based on the developed models for various protocol layers.

4. Use the models and performance predictions for optimization and tradeoff analysis.

The MANET design problem is considered in two interrelated instances: *off-line* and *on-line*. In the off-line case, the algorithms developed can be slow. These algorithms are used to characterize the baseline scenario, set up operational parameters of the baseline scenario, and recommend changes to the scenario in terms of topology control and dynamic addition of nodes to meet performance requirements. In the on-line case, the algorithms developed must be fast and of low complexity. These algorithms are used to react to real-time unexpected variations in the baseline scenario. Here probes and measurements are used to dynamically adapt to the changing scenario. We envisage having a rolling horizon window with the on-line algorithms used to predict performance in this window.

2.2.1 Inputs and Outputs

The inputs to the MANET network modeling and design problem include both the baseline scenario (mission) and the desired performance characteristics. The scenario (or mission) under investigation is specified as a set of frames of the scenario topology characteristics at different time instants t over the duration of the scenario T. The detailed list of inputs is described below:

1. The **duration of the scenario** T and the **set of time instants** $t \in [0, T]$ at which the scenario topology is specified. The time instances t can be chosen either periodi-

cally or aperiodically whenever the traffic load, connectivity graph between nodes, or channel conditions change measurably.

2. **Set of nodes** $\mathcal{N}$ with $|\mathcal{N}| = N$ where each node is identified by a number from 1 to N. Node Attributes include:

   (a) **Node Type**: used if nodes have different characteristics.

   (b) **M(t)**: Node location vector of size N where each element $M_i(t)$ is a 3-d vector representing the location of node i with respect to some fixed zero vector at the same time instances t as in bullet 1 above.

   (c) **Radio Characteristics**: This is used used to calculate energy used and to calculate connectivity. In order to calculate connectivity, we can use either of:

      - Distance-Based: Specify set of Maximum reception ranges corresponding to different transmit power levels.
      - SINR-Based: Specify Minimum and Maximum Transmitter Power (or a range of transmit powers allowed) and Nominal Receiver Sensitivity.

   (d) **Node processing power and queue lengths**: These are used in calculating node delay.

3. **D(t)**: Traffic demand matrix (or Information Exchange Requirements (IERs)) for the nodes $\mathcal{N}$ at the specified time instants t as in bullet 1. The traffic demand matrix of size $N * N$ consists of elements $D_{ij}(t)$ representing the traffic characteristics as well as requirements between source node i and destination node j ($\forall i, j \in \mathcal{N}$). The traffic demand shall allow for mixed traffic of voice, video, and data with different

quality of service requirements and priority. Each element $D_{ij}(t)$ consists of:

   (a) **Traffic Characteristics**: including traffic arrival rate (in bits per second or connections per second), packet size and traffic holding time (for connection oriented traffic).

   (b) **Throughput requirements**: specified in packet per second or bits per second.

   (c) **Delay requirements**: specified via maximum or mean value.

   (d) **Reliability**: specified as the probability that an individual source packet arrives at its destination.

   (e) **Priority/QoS**: depending on traffic type.

4. **E(t)**: Environment matrix for the nodes $\mathcal{N}$ at the same time instances t as in bullet 1 above. Each element $E_{ij}(t)$ describes the condition of the network environment between source node i and destination node j. This description contains all factors that affect the communication between nodes i and j and includes:

   (a) **Packet loss probability**: This depends on the environment including weather conditions.

   (b) **Signal power attenuation factor**: This can be obtained via TIREM and also considering weather conditions.

   (c) **Receiver Environment**:

      - Ambient Noise at Receiver

      - Jamming: As represented by the loss of capacity on selected links.

5. **RF Channel Models**: Specifies the channel behavior and loss model. E.g., $1/R^{\alpha}$, Rayleigh (with optional parameters), Scattering, Shadowing, etc.

6. **Aerial Platform (AP) Characteristics**:

   (a) Limitations on height, flight time, speed, etc.

   (b) Relay characteristics (ground to air, air to ground, air to air): like maximum coverage distance, channel capacity between APs and between AP-ground nodes.

7. **Uncertainty Features and Network Survivability Values**: Uncertainty in node location, traffic load, channel representation (uncertainty in **E(t)**) can also be input as part of the scenario. Link and node outages can be given as certain probability that the link or node fails. Survivability could be based on survivability under lost links only, lost nodes, flapping links or flapping nodes.

Given the inputs at each snapshot, the performance approximations for this snapshot are valid only when the system attains equilibrium. It is assumed that the conditions in a given snapshot stay constant for enough time that the system attains equilibrium before the next snapshot. The possible outputs from the design tool include:

- Characteristics of Baseline Network (with no addition of nodes or any optimizations)
  - Network End-to-End delay, throughput, survivability
  - Identification of network bottlenecks.

  - Answer if the required traffic throughput, delay, and survivability goals are met with the baseline network.

- Optimize the Baseline Network to meet performance objectives.

  - Select ”optimal” values for various model parameters.

  - If the network requirements are NOT met, find out if addition of Aerial Platforms (APs) can improve the network performance.

  - Select required transmit power for each AP and/or node (this determines connectivity and energy usage).

  - Characterize the new throughput/delay/survivability values and determine the improvement over the baseline scenario.

  - A curve (or equation) showing additional throughput as a function of additional APs.

- Tradeoff curves to help the designer optimize among various criteria, for e.g., Network Utilization (which can be maximized at the expense of delay), Network end-to-end delay, and Network Survivability.

2.3 30 Node Baseline Scenario

The realistic scenario used to generate results and validate the developed models is one derived from traces of a field experiment performed as part of the CBMANET project. The scenario considered is a fast moving network of 30 vehicles heading towards a rendezvous point. The scenario duration is for 500 seconds with vehicles moving at

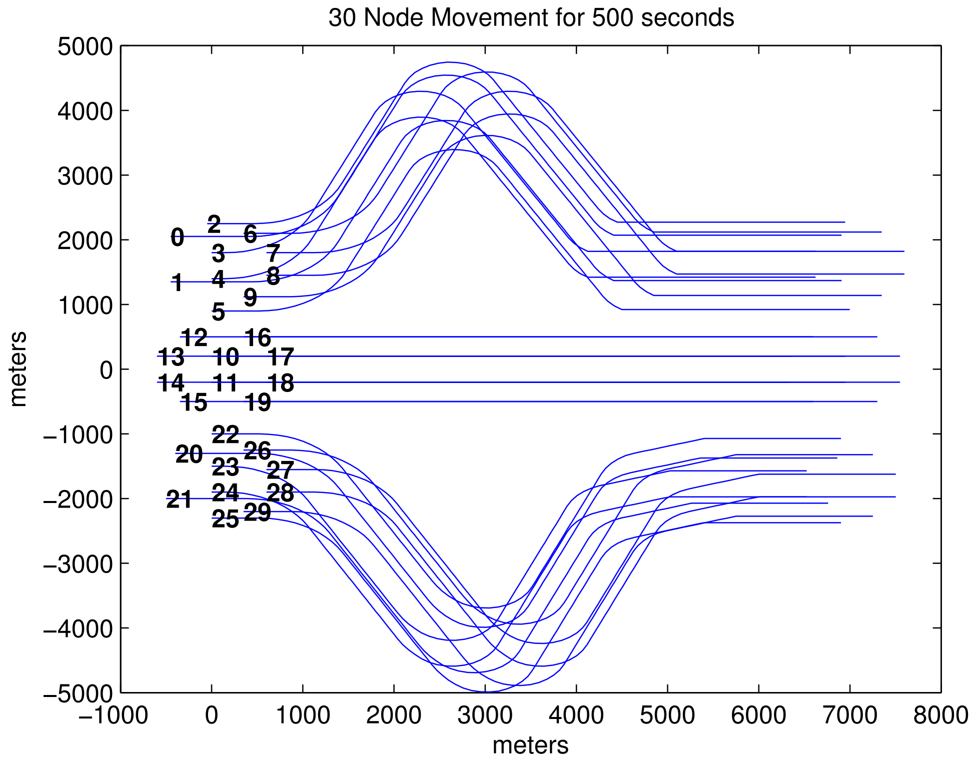

Figure 2.2: 30 Node Movement for 500 seconds

speeds between 22-60 mph. The vehicles start together, then branch into 3 clusters of 10 nodes each due to obstructions (2 steep hillocks), and finally rejoin (see figure 2.2). Upto two Aerial Platforms (APs) are used to maintain communication connectivity when the clusters become disconnected. The number and location of the APs are determined by a fast Deterministic Annealing algorithm described in chapter 4. From 0-30s, the ground nodes move together forming a connected network. From 30-420s, the nodes form 3 clusters as shown in figure 2.2 with cluster 2 (nodes 10-19) moving in between the two hills while clusters 1 (nodes 0-9) and 3 (nodes 20-29) go around the hills to the left and right of cluster 2 respectively. The clusters start to lose communication connectivity around 75s, then become disconnected from each other, and finally reconnect around

400s. APs are brought in to provide communication connectivity between the otherwise disconnected clusters from 75-400s.

The scenario is specified every 5 seconds (the ground nodes move an average of 100 meters in 5s). At every 5 second interval, the ground node positions, the traffic demands (offered load) & routes between source-destination pairs, and the environment conditions are input to the performance models. All ground nodes and APs have identical omni-directional radios with receiver sensitivity of -95dBm, receiver threshold of 10dB, and transmit power of 5W. We assume that the nodes use a Rake receiver (with diversity 5) and use BPSK (Binary Phase Shift Keying) modulation with thermal noise at the receiver modeled as $kTB * NF$ (NF = Noise Figure). The environment is modeled as a Rayleigh fading channel. We assume a $\frac{1}{R^\alpha}$ attenuation model for signal transmission through the radio channel; i.e., the signal strength decreases as $\frac{1}{R^\alpha}$ with distance (R) from the transmitter. The radio specification and the path loss exponent α together determine a maximum connectivity distance between nodes. The path loss exponent α is taken to be 4.5 between ground nodes, 3.9 between ground and aerial nodes, and 3.0 between the aerial nodes. This results in a maximum connectivity distance of 857m between ground nodes, 2423m between ground-aerial nodes, and 25099m between aerial nodes. Figure 2.3 shows the connectivity among the nodes for the scenario considered at snapshot 0. The maximum channel rate between any two connected nodes is set to 1 Mbps.

There are 17 source-destination connection pairs chosen in this scenario. The traffic between each source-destination pair is routed via the first K shortest distance paths. There are 13 intra-cluster connections (4 each in cluster 1 and 2; and 5 in cluster 3) with K, the number of paths per connection, equal to 2 or 3. The remaining 4 connections

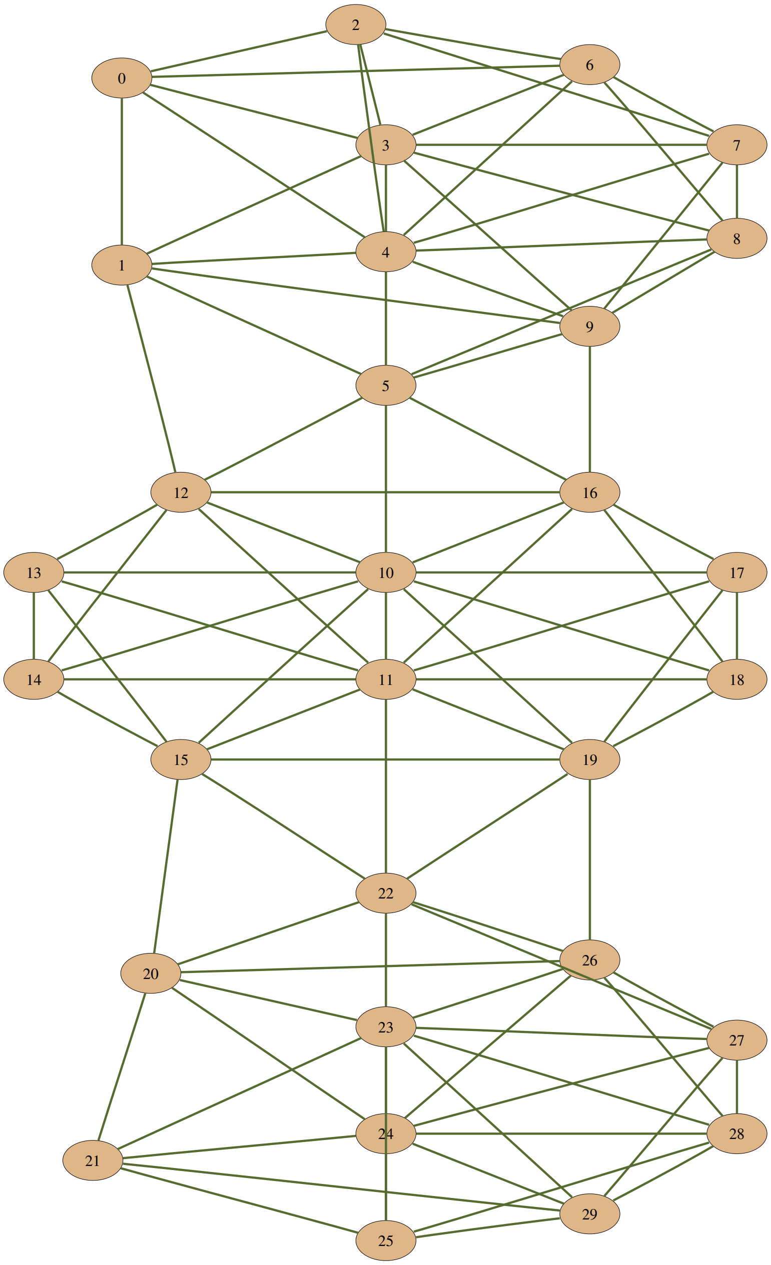

Figure 2.3: 30 Node Connectivity at Time Snapshot 0

span clusters with K ranging from 2 to 4 paths. Connection 11 between source node 20 and destination node 0 is the longest connection (with $K = 4$).

2.4 Random Access 802.11 DCF: Modeling and Design

We use the 802.11 DCF model developed in [10] to calculate throughput. The next section gives a brief overview of the throughput model. Section 2.4.2 describes the end to end delay analysis developed for the model in [10] assuming that each node has a M/M/1/N queue. Section 2.4.3 describes how we use Automatic Differentiation for throughput sensitivity computations. Finally in section 2.4.4, we provide results of running the 802.11 DCF throughput and delay models as well as the design methodology on the scenario described in section 2.3.

2.4.1 802.11 DCF Throughput Model Overview

We briefly go over the 802.11 DCF throughput model developed in [10] so that we can define the variables necessary for the end to end delay analysis developed in section 2.4.2. The 802.11 DCF MAC protocol is considered with the RTS/CTS mechanism.

Consider a network that consists of N nodes and a path set P that is used to forward traffic between the source destination (S-D) pairs in the network. Let P_i denote the set of the paths that goes through a node i. The scheduler behavior is specified by the scheduler coefficient $k_{i,p}$, which is the average serving rate of path p packets at node i. For simplicity, it is assumed that all packets have the same length. Let $\lambda_{i,p}$ be the arrival rate and $T_{i,p}$ be the service time of path p packets at node i. *The service time, $T_{i,p}$ is the time that node i scheduler spends serving a path p packet, and starts from the time that the scheduler selects a path p packet to be served and not from the time that the packet becomes head of the queue.* Note that the transmission time includes the retransmissions of a packet too. For instance, in the 802.11 MAC layer, if a transmission fails the packet will be retransmitted upto m times and after that the packet will be discarded.

The scheduling rate is a function of MAC and PHY layer packet failure probabilities. In the 802.11 RTS/CTS protocol there are two stages for packet transmission: the first stage is a RTS-CTS transfer and the second is Data packet-ACK transfer . The probability of transmission failure (PHY or MAC layer) for a packet of path p at node i is denoted by $\beta_{i,p}$. It is assumed that a MAC layer failure can only occur in stage 1 whereas a physical layer failure can occur in both stages.

The total average throughput $\bar{\rho_i}$, of node i, is,

$$\bar{\rho_i} = \sum_{p \in P_i} k_{i,p} E(T_{i,p}). \tag{2.1}$$

In order to model a FCFS queuing policy, it is assumed that the scheduler coefficients are:

$$k_{i,p} = \begin{cases} \lambda_{i,p} & \text{if } \sum\limits_{p' \in P_i} \lambda_{i,p'} E(T_{i,p'}) \leq 1 \\ \frac{\lambda_{i,p}}{\sum\limits_{p' \in P_i} \lambda_{i,p'} E(T_{i,p'})} & \text{otherwise} \end{cases} \tag{2.2}$$

As described in (2.2) if utilization of node i is less than one, all incoming packets can be served. However if the utilization of node i is greater than one, the scheduling rates can be obtained by normalizing the arrival rates by the average utilization to account for the server busy time.

Further the fraction of time $\rho_{i,p}$ that node i is serving path p packets is given by

$$\rho_{i,p} = k_{i,p} E(T_{i,p}). \tag{2.3}$$

Consider a simple source-based multi-path routing methodology. The routing model specifies a fixed set of paths and the fraction of incoming traffic that is sent over each path at the source node. Note that due to PHY and MAC layer loss parameters the incoming traffic rate at successive nodes of a path is a is non-increasing with every hop. The incoming traffic rates of the nodes are derived from the scheduling and loss rates of their upstream links as follows:

$$\lambda_{h_{i,p},p} = k_{i,p}(1 - \beta_{i,p}^{m}) \text{ for all } i, p. \tag{2.4}$$

The main output parameters for the random access modeling are $\beta_{i,p}$ and $E(T_{i,p})$ the packet transmission failure probability and expected service time of a packet of path p packets at node i. The details of the model and equations are given in [10]. This set of equations will be used as an implicit function to derive loss parameters and packet average service times from the node throughput. This model accounts for the effect of hidden nodes and multiple paths that share nodes in the network. Further the model also accounts for a finite attempt factor m after which the MAC layer packet is discarded.

The packet service time, $T_{i,p}$ is the time to finish a *successful or unsuccessful* transmission of a path p packet at node i, *after* it is scheduled for transmission at node i. The average service time $E(T_{i,p})$ has four components: $d_{i,p}$ is the time spent for successful transmission of path p packets at node i, $u_{i,p}$ is the average time consumed for successful transmission of node i neighbors, $b_{i,p}$ is the average back-off time of node i for path p packets, $c_{i,p}$ is the average time spent in failed transmissions.

$$E(T_{i,p}) = (1 - \beta_{i,p}^{m})d_{i,p} + u_{i,p} + b_{i,p} + c_{i,p} \tag{2.5}$$

In [10], a set of equations that relate each component of the service time to the transmission times of 802.11 PHY layer and transmission failure probabilities are provided.

2.4.2 802.11 DCF Delay Analysis

The delay analysis is based on the service time that is computed in the fixed point model. Thus, delay is computed after the fixed point model is computed and it does not add complexity to the fixed point model.

We start with the total average throughput $\bar{\rho}_i$ at node i computed in equation (2.1). If we model the queue at each node as an M/M/1/N queue then probability of having n packets in the queue at node i is,

$$\pi_{i,n} = \frac{(1-\bar{\rho}_i)\bar{\rho}_i{}^n}{1-\bar{\rho}_i{}^{N+1}} \tag{2.6}$$

Probability of dropping a path p packet due to congestion at node i is:

$$\gamma_{i,p} = \lambda_{i,p}\pi_{i,N} \tag{2.7}$$

The expected service time of a packet at node i is:

$$S_i = \frac{\sum\limits_{p\in P_i} \lambda_{i,p} E(T_{i,p})}{\sum\limits_{p\in P_i} \lambda_{i,p}} \tag{2.8}$$

The expected queue length is:

$$Q_i = \sum_{n=0}^{N} n\pi_{i,n} \tag{2.9}$$

Expected waiting time for a packet in the queue is:

$$\tau_i = S_i Q_i \tag{2.10}$$

The expected delay for a packet from path p at node i is,

$$D_{i,p} = \tau_i + E(T_{i,p}) \tag{2.11}$$

We can compute the delay over each path of the network by summing up the corresponding delay of the nodes in the path.

2.4.3 Sensitivities and Automatic Differentiation for Design

Although the fixed point models for random access 802.11 DCF described in the previous sections provide the basis for performance analysis of a given network configuration, we need a methodology for network configuration and optimization. We use optimal routing design as an example to illustrate our proposed design methodology. Given a set of paths between source-destination pairs, we use the gradient projection method to find the optimal values for the routing parameters (routing probabilities) to maximize the total network throughput. The gradient projection method requires iterative computation of the throughput gradient. If the throughput gradients can be computed analytically after convergence of the fixed point iterations, we can use them. But for the 802.11 DCF model, the fixed point method provides a computational scheme that, after convergence, describes the performance metric (i.e., throughput) as an implicit function of the design parameters (i.e., routing parameters). Thus, we do not have analytic expressions of the performance metric evaluations, but instead, we have a program that computes the values of the performance metric, while implicitly providing the dependence of the values on the design parameters. We use Automatic Differentiation (AD) to compute the gradients.

AD is a numerical method to compute the derivatives of a program [18]. Using the fact that a computer program is in fact a sequence of primary operations, automatic differentiation records the relationships between them and using the chain rule, it is able to provide the derivative of a function in a short amount of time. We use ADOL-C (Automatic Differentiation by OverLoading in C++), the source code of which is available at [19] to implement automatic differentiation on our fixed point model. Operator Overloading consists of changing the type of the variables involved in the computation to a proprietary type given by the Automatic Differentiation tool to allow it to compute derivatives based on its linked libraries.

ADOL-C computes the derivatives of real-valued variables and operations that take reals into reals. All the parameters in the 802.11 DCF FPA model are real-valued and hence we were able to successfully use AD to compute the throughput gradients.

2.4.4 Results: 802.11 DCF Model and Design

We run the random access 802.11 DCF fixed point model and design framework described in the previous three sections on the 30 ground node and 2 AP scenario as described in section 2.3. Table 2.1 lists the source-destination pairs chosen, the constant bit rate traffic offered and the number of paths K between each source and destination. The MAC packet size is set to 8544 bits and buffer size at each node is set to 35.

We run the entire scenario first with equiprobable flow splits among the various paths for a connection and then with flow split values optimized using Automatic Differentiation to maximize total throughput. Figure 2.4 shows the variation of total throughput and worst connection (which is the longest connection numbered 11) throughput with

Table 2.1: Src-Dst Traffic Characteristics: 802.11 DCF 30 Node Scenario

Conn #	Src-Dst	Rate (kbps)	Number of Paths
0	1-18	20	4
1	1-3	100	2
2	2-9	100	2
3	4-6	100	2
4	7-5	100	2
5	10-1	100	2
6	14-17	100	2
7	16-11	100	2
8	17-18	100	2
9	19-12	100	2
10	20-11	50	3
11	20-0	50	4
12	20-29	100	2
13	21-10	100	3
14	21-22	100	2
15	23-28	100	2
16	23-25	100	2

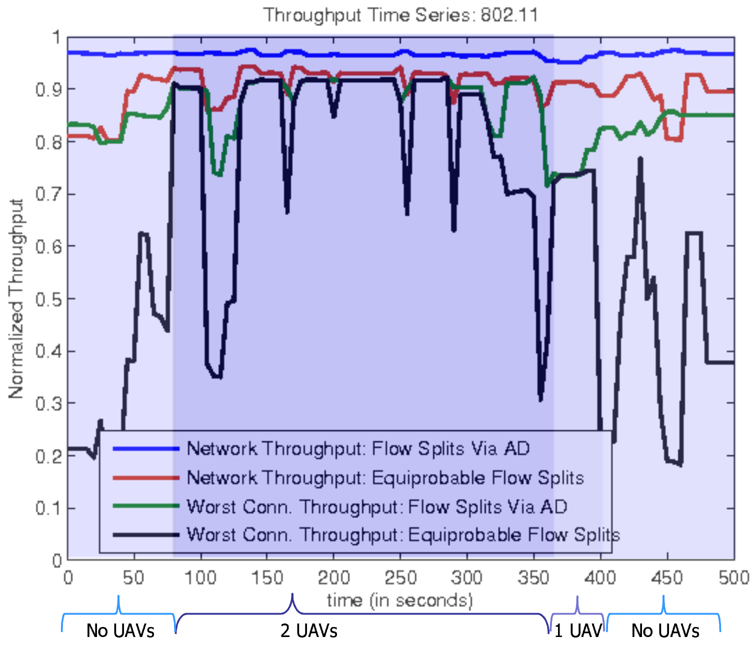

Figure 2.4: Throughput Time Series: 802.11 model

time for the two cases. Connection 11 with source node 20 and destination node 0 (figure 2.3) spans all the 3 clusters and exhibits the worst throughput since it has the maximum number of hops per path. Note that the total network throughput increases with the flow splits obtained by AD. In this scenario, the worst connection throughput also increases with the flow splits obtained by AD but this may not always be the case. It is only the total network throughput that is maximized and not individual connection throughput. Figure 2.5 shows the variation of average delay with time for connection 11 both with equiprobable flow splits and with flow splits determined by AD to maximize throughput. Since lower throughput implies bigger queues, maximizing the throughput also reduces the queuing delay.

To capture the effects of offered load on throughput (carried load) and delay, we run the scenario at a particular time (snapshot 0) but with offered loads for all connections

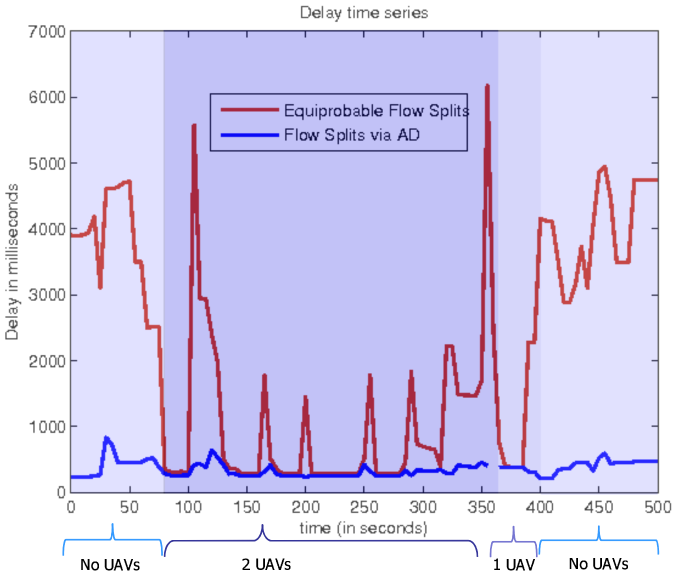

Figure 2.5: Delay Time Series for Connection 11: 802.11 model

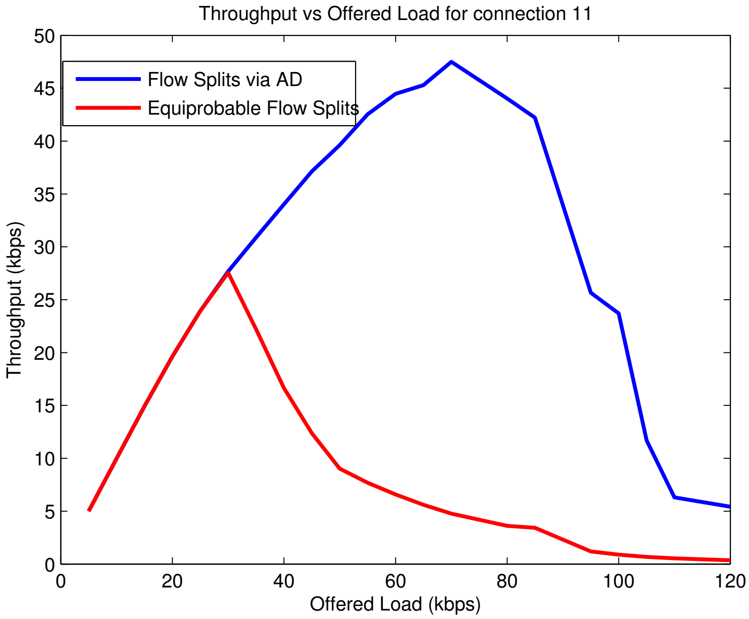

Figure 2.6: Carried Load vs Offered Load For Connection 11 at snapshot 0: 802.11 model

scaled by some common factor δ. As δ is varied, the offered load of all the connections varies. Figure 2.6 shows the variation in carried load with offered load for connection 11

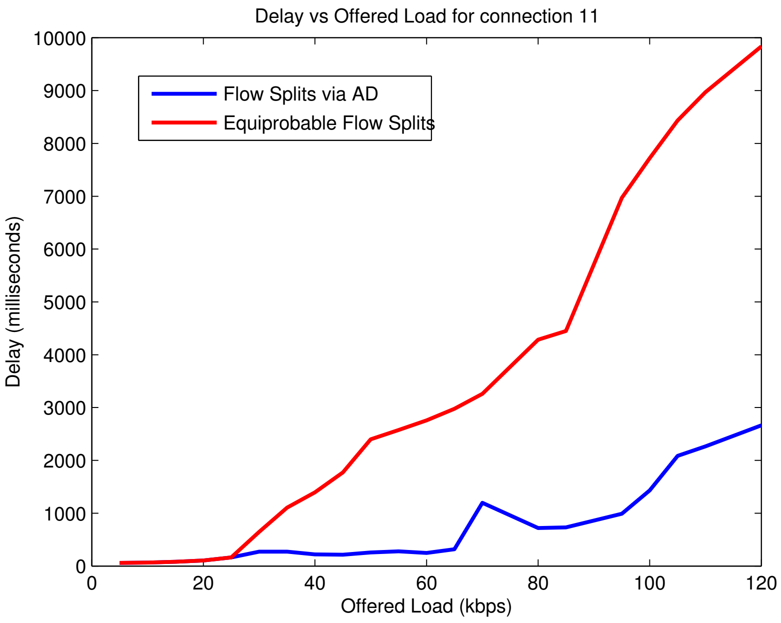

Figure 2.7: Delay vs Offered Load For Connection 11 at snapshot 0: 802.11 model

at time snapshot 0. We see that as the offered load is increased, the throughput for the longest connection (11) increases to a maximum and then decreases. At low total offered load, the network has not yet reached the capacity region and hence the throughput increases. But as offered load increases, there is more contention along all the paths and hence throughput decreases. The maximum connection throughput value is both higher and occurs at higher offered load when the flow splits are determined by AD as opposed to equiprobable flow splits. Figure 2.7 shows the variation in delay with offered load for connection 11. As the offered load increases, the contention to access the channel in neighborhoods increase, resulting in larger average time to access the channel thereby resulting in higher delay. Moreover as the offered load increases, the queuing delay increases which also contributes to higher delay.

Chapter 3

Modeling and Design of Reservation Based MANETs

3.1 Introduction

The Medium Access Control (MAC) layer in a MANET schedules a node's transmissions in a decentralized manner so as to prevent collisions with neighboring nodes' transmissions and is also tasked with recovering from unintended collisions (due to mobility, simultaneous channel access, etc.). Thus the MAC layer controls access to the wireless medium and hence plays a very important role in determining the performance of a wireless network. There are two basic mechanisms for multiple nodes to access a common channel: contention based access and reservation based or circuit-switched access. In a contention based access scheme, every node at each hop on the source-destination path contends with its neighboring transmitting nodes to obtain access to the communication channel. In contrast, in a reservation based or circuit-switched access scheme, a source node reserves the wireless channel along the entire route for some duration and then transmits data. In reservation based MAC protocols, the communication channel is usually divided into two separate portions, one for control traffic and the other for user traffic. In this chapter, we look at the modeling and design of reservation based MANETs. In particular, we look at performance analysis and design for a generic reservation based MAC protocol called Unifying Slot Assignment Protocol (USAP).

USAP [4] is the distributed resource allocation protocol used in Rockwell Collins' tactical battlefield wireless ad-hoc protocol suite called Wireless-wideband Networking Engine (WNE). The WNE USAP channel access is also adapted to the Mobile Data Link (MDL) layer of the Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW) [5]. USAP divides the communication channel into periodic frames and each frame is partitioned into orthogonal time-frequency cells with a portion of the cells reserved for control/management traffic and the rest for user traffic. The USAP protocol operates in two modes: hard scheduling mode (virtual circuit connection-oriented mode) where nodes reserve a session's link capacity end-to-end over the entire path; and soft scheduling mode (datagram scheduling) where nodes perform per-hop scheduling of links for single packets after the packet's arrival at the node. The USAP hard scheduling mode represents a generic reservation based MAC protocol and we focus on this mode of USAP.

We develop performance models for USAP Hard Scheduling and use them to both approximate the performance of a MANET and to optimize network performance. These models can assist us to design wireless networks and protocols, and to predict their performance. The performance metrics of interest in reservation based systems is the blocking probability and throughput for each source-destination pair. Our approach to performance evaluation and optimization is based on fixed point methods and reduced load approximations for loss network models. Loss network models [11] were originally used to compute blocking probabilities in circuit switched networks [13] and later were extended to model and design ATM networks [14–17]. In [17] reduced load approximations were used effectively to evaluate quite complex ATM networks, with complex and adaptive routing protocols, and multi-service multi-rate traffic (different service requirements). The main challenge in developing loss network models for wireless networks is coupling between wireless links. This coupling is due to sharing of the wireless medium between a node and its neighbors, resulting in a node's available link capacity to be dependent on its neighborhood traffic. We model this sharing of capacity by extending the reduced load loss network models to wireless networks by appropriately estimating individual link blocking probabilities. These link blocking probabilities are calculated using USAP reservation rules and traffic among neighboring nodes.

We assume we know the exogenous traffic rate for each source-destination pair and use multiple paths with a set of routing probabilities to forward traffic between a source and destination. The reduced load loss network model coupled with the wireless link blocking probability model and multiple path routing give us a set of a non-linear equations that are run iteratively to obtain a fixed point estimate of performance metrics like blocking probability and throughput. We then use the reduced load loss network model to calculate throughput sensitivities which are used to compute the optimal load distribution among multiple paths to maximize network throughput. We use an implied cost formulation [12] to calculate the throughput sensitivities. The implied cost formulation to calculate throughput sensitivities expresses the notion that when a call is admitted, it increases the current network throughput but also increases the blocking probability for future calls and hence reduces the future throughput (or implied cost).

This chapter is organized as follows. Section 3.2 introduces the generic USAP Hard Scheduling protocol as described in [4] which is basically a generic virtual-circuit reservation protocol. Section 3.3 describes related work in the field of circuit-switched networks and reduced load loss network modeling. Section 3.4 describes our fixed point models for USAP Hard Scheduling as described in 3.2. These models are based on using reduced load loss network models and estimating available link capacity pmf at each node by considering the USAP reservation rules and neighboring traffic. Section 3.5 discusses how we calculate throughput sensitivities and use them to maximize network throughput by computing optimal load distribution among multiple paths of a source-destination connection. Section 3.6 presents results of our USAP Hard Scheduling model of the previous two sections and compares them against simulation. Individual connection throughput shows good match with simulation when we use the reduced load loss network model but with reduced link capacity distribution obtained from simulation. Hence we can optimize network throughput using throughput sensitivities obtained from reduced load loss network models along with reduced link capacity distribution obtained from simulation.

In section 3.7, we look at the special case of the periodic USAP frame having only a single frequency channel and look at ways of estimating the link blocking probabilities for the reduced load approximation directly through analysis of the link neighborhood instead of first estimating the available link capacity pmf and then estimating the link blocking probability for each value of available link capacity as in section 3.4. We show a good match with simulation.

Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW) uses USAP modified for multicast to schedule transmissions so as to achieve contention free transmissions. In section 3.8 we look at USAP as used in the Mobile Data Link (MDL) of the JTRS WNW. Section 3.9 describes our modeling of MDL and the fixed point models used for USAP Hard Scheduling. Finally, in section 3.10 we present the time varying scenario used and the results of our fixed point model for USAP Hard Scheduling as used in MDL including its comparison with simulation. We show that our models for USAP as used in MDL matches well with simulation.

3.2 Generic USAP MAC Protocol

USAP [4] is the distributed resource allocation protocol used in Wireless-wideband Networking Engine (WNE: Rockwell Collins' tactical battlefield wireless ad-hoc protocol suite). The WNE USAP channel access is also adapted to the Mobile Data Link (MDL) layer of the Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW) [5]. We focus on USAP as described in [4].

USAP protocol [4] allows a transmitter to choose slots from the pool of unassigned slots in its neighborhood, coordinate the announcement and confirmation of this assignment with neighboring nodes upto two hops away, and detect and resolve conflicting assignments. USAP partitions the channel into time and frequency cells and constructs a periodic communication frame structure (see Fig. 3.1). Bootstrap minislots are pre-allocated to nodes for exchange of network management information and are used to reserve data channel cells. Broadcast slots support multicast/broadcast data and we do not consider them in our unicast modeling. Reservation slots support unicast data traffic. USAP frame *reservation* slots consists of $M \times F$ cells, where M is the number of *time slots* and F is the number of *frequency channels* in a frame. Once a cell is assigned to link (i, j), there is no contention as no node transmissions in the two-hop neighborhood interfere with link (i, j) transmission. It is assumed that each node has a single transceiver and hence nodes

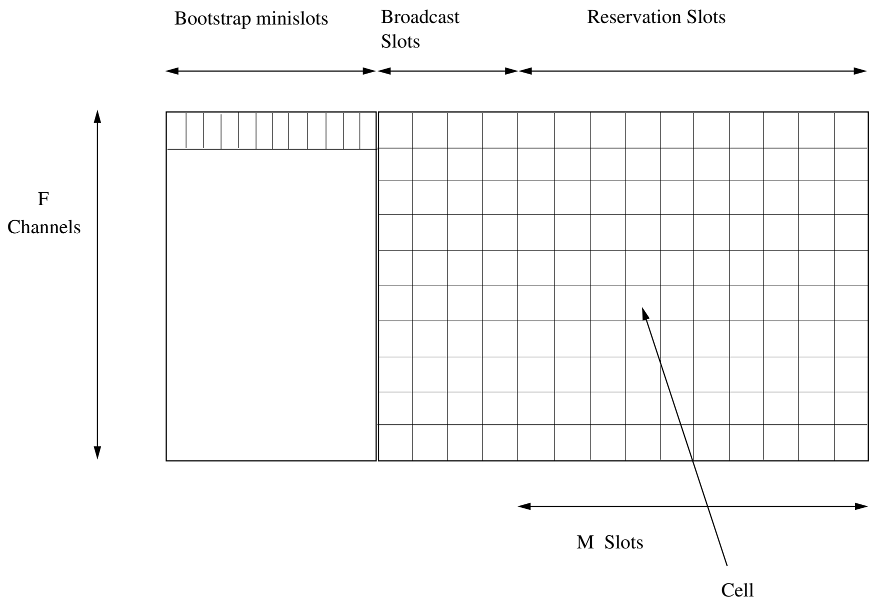

Figure 3.1: The USAP TDMA Frame Structure

i and j cannot transmit or receive on any other frequency channel corresponding to that time slot.

In the control channel (bootstrap mini-slots), every node broadcasts the cells that are reserved for transmission and reception by itself and its neighbors. In this way, every node acquires information about the reserved slots in its 2-hop neighborhood. Let $T(l)$ and $R(l)$ be the transmitting and receiving nodes of link l respectively. To avoid collision, cell reservation by node $T(l)$ for transmission to $R(l)$ is based on the following rules:

1. $T(l)$ cannot reserve other cells (with different frequency) on those time slots which already have scheduled incoming or outgoing cell transmissions to and from $T(l)$ and $R(l)$.

2. $T(l)$ cannot reserve cells already used by incoming calls to the neighbors of $T(l)$.

3. $T(l)$ cannot reserve cells already used by outgoing calls from the neighbors of $R(l)$.

These rules form the basis for our available link capacity approximation that will be described later.

USAP can function under a connection-oriented (hard scheduling) or connectionless (soft scheduling) framework. We model USAP hard scheduling mode where cells are reserved for the entire call duration on all links of the path from the source to destination. The performance metric for the hard scheduling case is the percentage of calls blocked for each connection. A call is blocked if there is not enough available capacity (cells in frame) on all links of the path. There is no significant queuing for the hard scheduling case; hence, delay is not an essential performance metric.

3.3 Related Work

The USAP Hard Scheduling mode can be modeled as a loss system. A loss system is defined as a collection of resources to which calls, each with an associated holding time and class, arrive at random instances. An arriving call is either admitted into the system or is blocked and lost; if the call is admitted, it uses certain resources simultaneously and remains in the system for the duration of the holding time. At the end of the holding time, the call releases all its used resources. The admittance decision is based on the call's class and the system's state. A loss system is different from a queuing system because a call's system sojourn time is equal to its holding time.

The simplest loss system is the Erlang loss system, which consists of a single link of C circuits to which calls of one class arrive. Each call occupies one of the circuits and an arriving call is blocked when the link is full. The calls arrive according to a Poisson process with rate λ, and the call holding times are independent and identically distributed with mean $1/\mu$. This system can be modeled as a Markov process for exponential holding times. The fraction of the calls blocked, B, is given by the Erlang-B formula,

$$B = \frac{\rho^C/C!}{\sum_{c=0}^{C} \rho^c/c!}$$

where the offered load $\rho = \lambda/\mu$.

The extension of the Erlang loss system to multiple call classes can be modeled via the stochastic knapsack. The stochastic knapsack consists of C resource units to which objects from K classes arrive. Class-k objects (or calls) arrive at the knapsack (or link) according to a Poisson process with rate λ_k and the K arrival processes are independent. If a class-k object is admitted, it holds b_k resource units for a holding time that is distributed with mean $1/\mu_k$. Holding times are independent of each other and of the arrival processes. Let n_k denote the number of class-k objects in the knapsack. Then the state space $\mathcal{S}$ of the knapsack, i.e., the set of allowable objects is given by

$$\mathcal{S} = \left\{\mathbf{n} \in \mathcal{I}^K : \mathbf{b} \centerdot \mathbf{n} \leq C\right\}$$

where $\mathbf{b} = (b_1, \ldots, b_k)$, $\mathbf{n} = (n_1, \ldots, n_k)$, and $\mathcal{I}$ is the set of non-negative integers. The equilibrium distribution of the stochastic knapsack is given by

$$\pi(\mathbf{n}) = \frac{1}{G} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}, \quad \mathbf{n} \in \mathcal{S}$$

where the offered load $\rho_k = \lambda_k/\mu_k$ and the normalization constant G is given by

$$G = \sum_{\mathbf{n} \in \mathcal{S}} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}$$

A Markov process captures the dynamics of the stochastic knapsack if the holding times are exponentially distributed. But the product form equilibrium distribution holds for more arbitrary holding time distributions. Since the arrival processes are Poisson, the blocking probability B_k for a class-k object is given by

$$B_k = 1 - \sum_{\mathbf{n} \in \mathcal{S}_k} \pi(\mathbf{n})$$

where, $\mathcal{S}_k$ is the subset of the state space in which the knapsack can admit an arriving class-k object and is given by

$$\mathcal{S}_k = \{\mathbf{n} \in \mathcal{S} : \mathbf{b} \centerdot \mathbf{n} \leq C - b_k\}$$

The blocking probabilities for a stochastic knapsack can be efficiently calculated using a simple recursion for the occupancy distribution $q(c)$. The occupancy distribution $q(c)$ is the probability that exactly c resource units are used in the knapsack with capacity C and is given by

$$q(c) = \sum_{\mathbf{n} \in \mathcal{S}(c)} \pi(\mathbf{n}) \quad \text{where } \mathcal{S}(c) = \{\mathbf{n} \in \mathcal{S} : \mathbf{b} \centerdot \mathbf{n} = c\}$$

The occupancy distribution $q(c)$ satisfies the following recursion

$$cq(c) = \sum_{k=1}^{K} b_k \rho_k q(c - b_k)$$

The computational complexity to find the occupancy distribution is $O(CK)$. The blocking probability B_k for a class-k call can now be calculated as

$$B_k = \sum_{c=C-b_k+1}^{C} q(c)$$

Consider a wired network of J links with link j having capacity C_j units. Again let there be K call classes with each class-k call arriving as a Poisson process with rate λ_k, requiring b_k bandwidth units along route $R_k \subseteq \{1, \ldots, J\}$, and with mean holding time $1/\mu_k$. The calls from the K classes arrive as independent Poisson processes and the holding times of the calls are independent of each other and of the Poisson arrival processes. An arriving class-k call is admitted into the network if b_k bandwidth units are free in each link $j \in R_k$; else the call is blocked and lost. Let $\mathcal{K}$ be the set of all classes and $\mathcal{K}_j$ be the set of classes that use link j, i.e.,

$$\mathcal{K}_j = \{k \in \mathcal{K} \,:\, j \in R_k\}$$

The state space $\mathcal{S}$ for this loss network with fixed routing is given by

$$\mathcal{S} = \left\{ \mathbf{n} \in \mathcal{I}^K \,:\, \sum_{k \in \mathcal{K}_j} b_k n_k \leq C_j,\ j = 1, \ldots, J \right\}$$

Let S_k be the subset of states for which a class-k call can be admitted. It is given by

$$\mathcal{S}_k = \left\{ \mathbf{n} \in \mathcal{S} \,:\, \sum_{l \in \mathcal{K}_j} b_l n_l \leq C_j - b_k,\ j \in R_k \right\}$$

The equilibrium distribution for the loss network is given by

$$P(\mathbf{X} = \mathbf{n}) = \frac{1}{G} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}, \quad \mathbf{n} \in \mathcal{S}$$

where the offered load $\rho_k = \lambda_k/\mu_k$ and the normalization constant G (also called the partition function) is given by

$$G = \sum_{\mathbf{n} \in \mathcal{S}} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}$$

The equilibrium distribution is similar to the stochastic knapsack except that the state space is different. Similar to the stochastic knapsack, the probability of blocking a class-k call, B_k, is given by

$$B_k = 1 - \frac{\sum_{\mathbf{n} \in \mathcal{S}_k} \prod_{l=1}^{K} \rho_l^{n_l}/n_l!}{\sum_{\mathbf{n} \in \mathcal{S}} \prod_{l=1}^{K} \rho_l^{n_l}/n_l!}$$

For all but the smallest networks it is impractical to compute the normalization constant G directly. Louth [20] showed that calculating the normalization constant for arbitrary topologies in an NP-complete problem. Observe that the number of routes taken (which can equate to the number of call classes) can grow as fast as exponentially with the number of links J. Even for the trivial case of the number of call classes being equal to the number of links J and with route R_k being just the link k, the size of the state space $|\mathcal{S}| = \prod_{j=1}^{J} C_j$ grows rapidly with the capacities $C_1, \ldots, C_J$. Even in the restricted case where links have capacity 1 and arrival rates are equal, the task of computing the partition function is $\#P$-complete [20].

Since it is impractical to compute the partition function G directly, there has been lot of work towards finding approximations to the loss network model in order to avoid these computational problems. It has been shown [11] that under the limiting regime in which the capacities C_j, $j = 1, \ldots, J$, and the offered loads ρ_k, $k = 1, \ldots, K$ are increased together (with ratios C_j/ρ_k held fixed), then there is a parameter a_j associated with link j, such that the asymptotic blocking probability of each route, B_k, $k \in \mathcal{K}$, is as

$$1 - B_k \longrightarrow \prod_{j \in R_k} (1 - a_j)^{b_k} \tag{3.1}$$

Thus the asymptotic blocking probabilities are as if the links block independently, with link j blocking with probability a_j.

This leads to a class of approximation procedures for a multi-service loss network called the *reduced load approximation* or *fixed point approximation*. It can be shown that the blocking probabilities calculated by these approximation procedures tend to the asymptotic blocking probabilities given by equation 3.1. For the case of a multiservice loss network with fixed routing, the fixed point approximation is given by the following equations

$$\rho_{j,k} = \rho_k \prod_{i \in R_k - \{j\}} (1 - B_{i,k}) \tag{3.2}$$

$$B_{j,k} = Q_k \left[C_j; \rho_{j,m},\, m \in \mathcal{K}_j\right] \tag{3.3}$$

where $Q_k \left[C_j; \rho_{j,m},\, m \in \mathcal{K}_j\right]$ is the probability of blocking a class-k object in a stochastic knapsack with capacity C_j to which objects from classes in $\mathcal{K}_j$ arrive as Poisson processes with offered load $\rho_{j,m},\, m \in \mathcal{K}_j$. The fixed point ($\rho_{j,k}$, $B_{j,k}$) is achieved by iteration. The probability of blocking a class-k call, B_k is approximated as

$$B_k \approx 1 - \prod_{j \in R_k} (1 - B_{j,k}) \tag{3.4}$$

There are two underlying assumptions behind the reduced load approximation:

- *Link independence*: It is assumed that each link blocks independently of other links, so that the probability that a call is accepted on a particular route is equal to the product of the probabilities that the call is accepted on each individual link on that route.

- *Poisson arrival at every link*: It is assumed that each class-k call arrives at a link on its route as a Poisson process with a reduced offered load that is equal to the original offered load thinned due to blocking on other links in its route.

3.4 Generic USAP Hard Scheduling Model

We model USAP Hard Scheduling using a reduced load loss network approximation extended for a wireless network with the sharing of the wireless medium modeled via reduced average link capacity calculated using USAP reservation rules and traffic among neighboring nodes. The next section describes the extended reduced load loss network approximation while section 3.4.2 describes how we estimate the available link capacity.

3.4.1 Reduced Load Loss Network Approximation

We are given the statistics of all ongoing source-destination connections along with the routes assigned to these connections. We assume that calls for route r arrive as a Poisson process with rate λ_r and with mean holding time $1/\mu_r$. The call demand in terms of the number of (reserved) cells per frame is n_r.

Offered load, $\rho_{l,r}$, of route r calls arriving at link l is reduced due to blocking on other links in the route and is given by

$$\rho_{l,r} = \frac{\lambda_r}{\mu_r} \prod_{k \in r/\{l\}} (1 - B_{k,r}) \tag{3.5}$$

where, $B_{k,r}$ is the probability of blocking a call on link k along route r. The blocking probability, B_r, for a connection traversing route r towards its destination is given by

$$B_r = 1 - \Pi_{l \in r}(1 - B_{l,r}) \tag{3.6}$$

Denote by $Q_r\left[C_l; \rho_{r'}, r' \in R_l\right]$ the blocking probability for route r calls on a link l with capacity C_l which has a set of routes R_l going through it. We have,

$$Q_r\left[C_l; \rho_{r'}, r' \in R_l\right] = 1 - \sum_{c=0}^{c=C_l-n_r} q_{C_l}(c) \tag{3.7}$$

where the $q_{C_l}(c)'s$, are the probabilities of having c cells occupied on link l with capacity C_l (i.e., knapsack occupancy probabilities, chapter 2 of [12]).

In the wireless network case the link capacity is not fixed, and it depends on the number of ongoing connections in the 2-hop neighborhood of the link. For any given value of link capacity m, the occupancy probabilities can be computed easily using the standard recursive stochastic knapsack algorithm (chapter 2 of [12]). If we assume that the link l capacity is between $C_l^{\min}$ and $C_l^{\max}$ with some given probability distribution, we have

$$B_{l,r} = \sum_{m=C_l^{\min}}^{C_l^{\max}} Pr[C_l = m]\, Q_r\left[m; \rho_{l,r'}, r' \in R_l\right] \tag{3.8}$$

From the occupancy probabilities, we also compute η_{ij}, the average number of cells reserved by link $l = (i, j)$:

$$\eta_{ij} = \sum_{m=C_l^{\min}}^{C_l^{\max}} Pr[C_l = m] \sum_{c=0}^{m} c q_m(c) \tag{3.9}$$

We need to estimate link capacity probabilities $P[C_l = m]$ in equations (3.8), (3.9). We describe our methodology for estimating the capacity distribution in the following section.

3.4.2 Link Capacity Estimation

Consider link l with transmitting node i and receiving node j. Let $N(i)$ denote the neighbors of node i. Neighbors of i and j can be split into the following (see figure 3.2): neighbors common to i and j, i.e., $N(i) \cap N(j)$; neighbors of i hidden from j, i.e., $N(i) - N(j)$; and neighbors of j hidden from i, i.e., $N(j) - N(i)$. Denote by: Γ_i^T and $\Gamma_{i,X}^T$ the average number of reserved slots used by node i in transmitting to all its neighbors and to neighbors in set X respectively; Γ_i^R and $\Gamma_{i,Y}^R$ the average number of reserved slots used by i in receiving from all its neighbors and from neighbors in node set Y respectively.

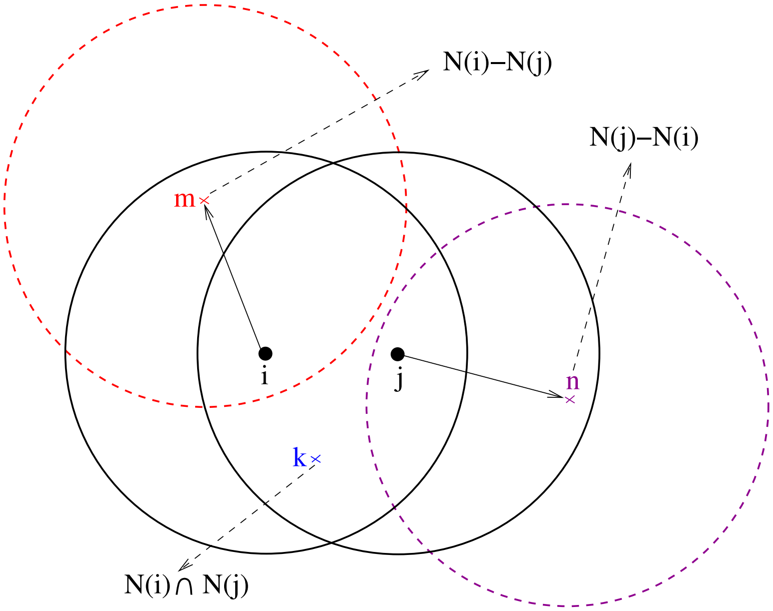

Figure 3.2: Neighbors of i & j: TXs (i,m) & (j,n) can share same cell

Then,

$$\begin{aligned}\Gamma_i^T &= \sum_{k\in N(i)} \eta_{ik} = \Gamma^T_{i,N(i)\cap N(j)} + \Gamma^T_{i,N(i)-N(j)} + \eta_{ij}\\ \Gamma_i^R &= \sum_{k\in N(i)} \eta_{ki} = \Gamma^R_{i,N(i)\cap N(j)} + \Gamma^R_{i,N(i)-N(j)} + \eta_{ji}\end{aligned}$$

We calculate $C_l^{\min}$ and $C_l^{\max}$, a low and high estimate of the number of cells available to link l for sending traffic, by using USAP reservation rules and computing minimum and maximum estimates for the number of slots and cells used by i, j, and their neighbors. A *slot* refers to all the cells (over all frequency channels) at that particular time slot in the frame. Corresponding to the first USAP reservation rule (section 3.2), we denote by $R^1_{\min}(i,j)$ and $R^1_{\max}(i,j)$ the low and high estimate respectively of the number of *time slots* used by the nodes i and j to transmit and receive from their neighbors barring the transmission from i to j. Corresponding to the second reservation rule, we denote by $R^2_{\min}(i,j)$ and $R^2_{\max}(i,j)$ the low and high estimate respectively of the number of *cells* used by the neighbors of i (except j) to receive traffic from their neighbors except the transmissions from i and j to neighbors of i. Similarly corresponding to the third reservation rule, we denote by $R^3_{\min}(i, j)$ and $R^3_{\max}(i, j)$ the low and high estimate respectively of the number of *cells* used by the neighbors of j (except i) to transmit traffic to their neighbors except the transmissions from j's neighbors to j and i. We assume a uniform distribution between $C_l^{\min}$ and $C_l^{\max}$.

3.4.2.1 Computing $R^1_{\min}(i, j)$ and $R^1_{\max}(i, j)$

From the USAP reservation rules and neighbor sets of i and j (figure 3.2), we infer that transmissions between sets $[i \rightarrow N(i) - N(j)]$ and $[j \rightarrow N(j) - N(i)]$ can share cells. The same applies to the sets $[N(i) - N(j) \rightarrow i]$ and $[N(j) - N(i) \rightarrow j]$. We also deduce that any transmissions between the sets $[i \rightarrow N(i) \cap N(j)]$, $[N(i) \cap N(j) \rightarrow i]$, $[j \rightarrow N(i) \cap N(j)]$, and $[N(i) \cap N(j) \rightarrow j]$ cannot share cells. Furthermore, when nodes i and j transmit to or receive from a common neighbor k in the set $N(i) \cap N(j)$, then these transmissions or receptions not only cannot share the same cell but also cannot share the same time slot. We consider two cases $F = 1$ and $F > 1$ when computing $R^1_{\min}(i, j)$ and $R^1_{\max}(i, j)$.

Case F $= 1$: Since transmissions $[i \rightarrow N(i) - N(j)]$ and $[j \rightarrow N(j) - N(i)]$ can share cells (with $F = 1$, share time slots), the minimum number of slots needed is the maximum of the two. Similar arguments hold for transmissions $[N(i) - N(j) \rightarrow i]$ and $[N(j) - N(i) \rightarrow j]$. Since transmissions between the sets $[i \rightarrow N(i) \cap N(j)]$, $[N(i) \cap N(j) \rightarrow i]$, $[j \rightarrow N(i) \cap N(j)]$, and $[N(i) \cap N(j) \rightarrow j]$ cannot share cells, now with $F = 1$, these sets have to be allocated in different time slots. Therefore,

$$R^1_{\min}(i,j) = \Gamma^T_{i,N(i)\cap N(j)} + \Gamma^R_{i,N(i)\cap N(j)} + \Gamma^T_{j,N(i)\cap N(j)}$$
$$+\Gamma^R_{j,N(i)\cap N(j)} + \max(\Gamma^T_{i,N(i)-N(j)}, \Gamma^T_{j,N(j)-N(i)})$$
$$+ \max(\Gamma^R_{i,N(i)-N(j)}, \Gamma^R_{j,N(j)-N(i)}) + \eta_{ji}$$
$$R^1_{\max}(i,j) = \Gamma^T_i + \Gamma^R_i + \Gamma^T_j + \Gamma^R_j - 2\eta_{ij} - \eta_{ji}$$

Case F > 1: If $F > 1$, then

$$\begin{aligned} R^1_{\min}(i,j) &= \max(\Gamma^T_i + \Gamma^R_i - \eta_{ij}, \Gamma^T_j + \Gamma^R_j - \eta_{ij}) + \\ &\quad O_{N(i)\cap N(j)} \\ R^1_{\max}(i,j) &= \Gamma^T_i + \Gamma^R_i + \Gamma^T_j + \Gamma^R_j - 2\eta_{ij} - \eta_{ji} \end{aligned}$$

where, $O_{N(i)\cap N(j)}$ is the overflow due to nodes i and j transmitting and receiving from common nodes k in set $N(i) \cap N(j)$. This overflow is calculated based on a filling argument; basically the cells corresponding to common node k should be in different time slots. The cells not considered in $R^1_{\min}(i,j)$ take up additional cells in a different frequency and need to be considered when calculating $C^{\max}_l$. Let the number of these cells be $S^1_{\min}(i,j)$.

$$\begin{aligned} S^1_{\min}(i,j) &= R^1_{max}(i,j) - R^1_{min}(i,j) \\ &\quad - \min(\Gamma^T_{i,N(i)-N(j)}, \Gamma^T_{j,N(j)-N(i)}) \\ &\quad - \min(\Gamma^R_{i,N(i)-N(j)}, \Gamma^R_{j,N(j)-N(i)}) \end{aligned}$$

3.4.2.2 Computing Conflict Graph and Cliques for $R^2_{\min}(i,j)$ and $R^3_{\min}(i,j)$

$R^2_{\min}(i,j)$ is the low estimate of the number of cells used by neighbors of i (except j) to receive traffic from their neighbors (except transmissions from i and j). This corresponds to finding the maximum of sum of average cells used by those neighboring nodes (except j) of i that cannot receive simultaneously. Hence we create a conflict graph whose vertices are the links in the neighborhood of i (excluding links with receiver node j) that receive data from nodes other than i and j and with edges between those links (vertices) that cannot receive simultaneously. From this conflict graph, we find all the maximal cliques and use them to find $R^2_{\min}(i,j)$. An edge is drawn between two vertices in the conflict graph, say, links (k,l) and (m,n), if $k = (m \text{ or } n)$, if $l = (m \text{ or } n)$, if n is within transmit range of k, or if l is within transmit range of m. The set of maximal cliques can be found by either Bierstone's method [21] or Bron's method [22]).

$R^3_{\min}(i,j)$ is the maximum of sum of average cells used by neighboring nodes (except i) of j that cannot transmit simultaneously and is computed in a similar manner to $R^2_{\min}(i,j)$ with the conflict graph created from links around node j (excluding links with transmitting node i) that transmit to nodes other than i and j.

3.4.2.3 Computing $R^2_{\min}(i,j)$, $R^3_{\min}(i,j)$, and $\eta^{comn}_{\min}(i,j)$

The rationale behind computing $R^2_{\min}(i,j)$ and $R^3_{\min}(i,j)$ is that at least these $R^2_{\min}(i,j) + R^3_{\min}(i,j)$ number of cells need to be reserved and hence need to be subtracted from available cells when computing $C^{\max}_l$. But there could be some common links between the maximal cliques used to compute $R^2_{\min}(i,j)$ and $R^3_{\min}(i,j)$ that contribute a total of $\eta^{comn}_{\min}(i,j)$ cells. Let $r^2_{\min}(i,j)$ be the sum of the average cells reserved by the links in a maximal clique around node i, let $r^3_{\min}(i,j)$ be the sum of the average cells reserved by the links in a maximal clique around node j, and let $\eta^{comn}(i,j)$ be the sum of average cells reserved by the common links between $r^2_{\min}(i,j)$ and $r^3_{\min}(i,j)$. Hence we set $R^2_{\min}(i,j)$, $R^3_{\min}(i,j)$, and $\eta^{comn}_{\min}(i,j)$ to be that $r^2_{\min}(i,j)$, $r^3_{\min}(i,j)$, and $\eta^{comn}(i,j)$ respectively that maximizes the sum $r^2_{\min}(i,j)+r^3_{\min}(i,j)-\eta^{comn}(i,j)$.

3.4.2.4 Computing $R^2_{\max}(i,j)$, $R^3_{\max}(i,j)$, and $\eta^{comn}_{\max}(i,j)$

$$\begin{aligned}
R^2_{\max}(i,j) &= \sum_{k\in\{N(i)-j\}} (\Gamma^R_k - \eta_{ik} - \eta_{jk}) \\
R^3_{\max}(i,j) &= \sum_{k\in\{N(j)-i\}} (\Gamma^T_k - \eta_{kj} - \eta_{ki}) \\
\eta^{comn}_{\max}(i,j) &= \sum_{k\in\{N(j)-i\}} \sum_{m\in\{N(i)-j\}} \eta_{km}
\end{aligned}$$

3.4.2.5 Computing $C^{\min}_l$ and $C^{\max}_l$

For $F = 1$,

$$\begin{aligned}
C^{\min}_l &= \max\big\{0,\ M - \big(R^1_{\max}(i,j) + R^2_{\max}(i,j) \\
&\qquad + R^3_{\max}(i,j) - \eta^{comn}_{\max}(i,j)\big)\big\} \\
C^{\max}_l &= \max\big\{0,\ M - \big(R^1_{\min}(i,j) + R^2_{\min}(i,j) \\
&\qquad + R^3_{\min}(i,j) - \eta^{comn}_{\min}(i,j)\big)\big\}
\end{aligned}$$

For $F > 1$,

$$C^{\min}_l = \max\{0,\ M - R^1_{\max}(i,j) - \big[R^2_{\max}(i,j)+$$

$$
\begin{aligned}
& R^3_{\max}(i,j) - \eta^{comn}_{\max}(i,j) - M \times (F-1)\big]^+\} \\
C_l^{\max} \; = \; & \max\{0,\ M - R^1_{\min}(i,j) - \big[S^1_{\min}(i,j) + \\
& R^2_{\min}(i,j) + R^3_{\min}(i,j) - \eta^{comn}_{\min}(i,j) - \\
& M \times (F-1)]^+\}
\end{aligned}
$$

3.5 USAP Hard Scheduling Throughput Sensitivities

Total throughput $TH(\mathbf{C_l})$ for USAP Hard Scheduling is the total cell demands that are not blocked and depends on the vector of free capacities $\mathbf{C_l}$ over all the links l , i.e.,

$$TH(\mathbf{C_l}) = \sum_{s \in S} \sum_{r=1}^{k_s} n_s \alpha_{r_s} \frac{\lambda_s}{\mu_s}(1 - B_r) \tag{3.10}$$

where S is the set of all source-destination connections, k_s is the total number of routing paths for a connection s, n_s is the call demand (number of reserved cells per frame) for connection s, and α_{r_s} is the fraction of calls that are routed over path r for connection s.

For the reduced load approximation of a multi-service loss network, it is possible to analytically calculate the throughput sensitivities using the implied cost formulation (see section 5.7 of [12]). In order to connect to the implied cost formulation, n_s is equal to the rate at which a call on route r for connection s earns revenue. Consider adding a single call to route r of connection s in equilibrium. This call is admitted with probability $1 - B_r$; if admitted the call uses an average of n_s/μ_s cells or earns an average revenue of n_s/μ_s, but reduces the future expected revenue or throughput due to the additional blocking that its presence causes. This expected loss in future revenue or throughput is called the implied cost (c_{r_s}) of route r call of connection s. Hence throughput sensitivities are given by:

$$\frac{\partial}{\partial \alpha_{r_s}} TH(\mathbf{C_l}) = \lambda_s(1 - B_r)\Big(\frac{n_s}{\mu_s} - c_{r_s}\Big) \tag{3.11}$$

$$\text{where, } c_{r_s} = \frac{1}{\mu_s}[TH(\mathbf{C_l}) - TH(\mathbf{C_l} - \mathbf{n_{lr_s}})] \tag{3.12}$$

and $\mathbf{n_{lr_s}}$ is a vector specifying the call demand for route r of connection s over all the links l. The implied costs are approximated using link independence assumption. Thus the total implied cost for route r of connection s is approximated to be the sum of individual link implied costs along all links l of route r of connection s and is given by:

$$c_{r_s} = \sum_{l \in r_s} c_{lr_s} \tag{3.13}$$

A fixed point approximation procedure is used to find the link implied costs similar to that in [12] (section 5.7). The equations are:

$$c_{lr_s} = \sum_{r'_{s'} \in R_l} \frac{\Delta_{r'_{s'}lr_s} \rho_{l,r'_{s'}}}{\mu_s} \left[n_{s'} - \mu_{s'} \sum_{i \in r'_{s'} - l} c_{ir'_{s'}} \right] \tag{3.14}$$

where,

$$\Delta_{r'_{s'}lr_s} = \sum_{m=C_l^{min}}^{C_l^{max}} \Big(P[C_l = m]\{Q_{r'_{s'}}[m - n_s; \rho_{l,q}, q \in R_l] - Q_{r'_{s'}}[m; \rho_{l,q}, q \in R_l]\} \Big) \tag{3.15}$$

Having obtained the throughput sensitivities with respect to routing probabilities, we use the gradient projection method to find the optimal values for routing probabilities to maximize total network throughput.

3.6 Generic USAP Hard Scheduling Results and Validation

The 30 node scenario specified in 2.3 is used. There are 17 source-destination connection pairs chosen in this scenario. The traffic between each source-destination pair is routed via the first K shortest distance paths. All the connections have holding time of 2 minutes. Connection 11 between source node 20 and destination node 0 is the longest connection (with $K = 4$). Table 3.1 list the source-destination connection pairs chosen, the number of paths K, the call arrival rate in calls/min and the holding time in minutes.

3.6.1 USAP Parameters

The USAP frame period is set to 125ms and the capacity of all the frequency channel is set to 1 Mbps. The number of frequency channels (F) is set to 2 and the number of reservation time slots (M) is set to 25. Only half of the USAP frame period is used for reservation slots. Based on the sum capacity of all channels, M, F, and the fraction of frame period used for reservation slots, 1250 bits can be carried per reservation cell. Hence for a connection to have a call demand (n_r) of 1 reservation slot per frame, the call demand rate (for e.g., the voice coder rate) is 10 kbps. We assume that the voice coder rate is 10 kbps (hence voice calls use 1 reservation cell per frame) and the voice coder frame period is 125ms. All chosen source-destination pairs have a call demand of 1 reservation slot per frame.

Table 3.1: Src-Dst Traffic Characteristics: Generic USAP HS 30 Node Scenario

Conn Number	Src-Dst	Num. of Paths	λ_s (calls/min)	hold time (min)
0	1-18	4	0.5	2
1	1-3	2	2.5	2
2	2-9	2	2.5	2
3	4-6	2	2.5	2
4	7-5	2	2.5	2
5	10-1	2	2.5	2
6	14-17	2	2.5	2
7	16-11	2	2.5	2
8	17-18	2	2.5	2
9	19-12	2	2.5	2
10	20-11	3	1.25	2
11	20-0	4	1.25	2
12	20-29	2	2.5	2
13	21-10	3	2.5	2
14	21-22	2	2.5	2
15	23-28	2	2.5	2
16	23-25	2	2.5	2

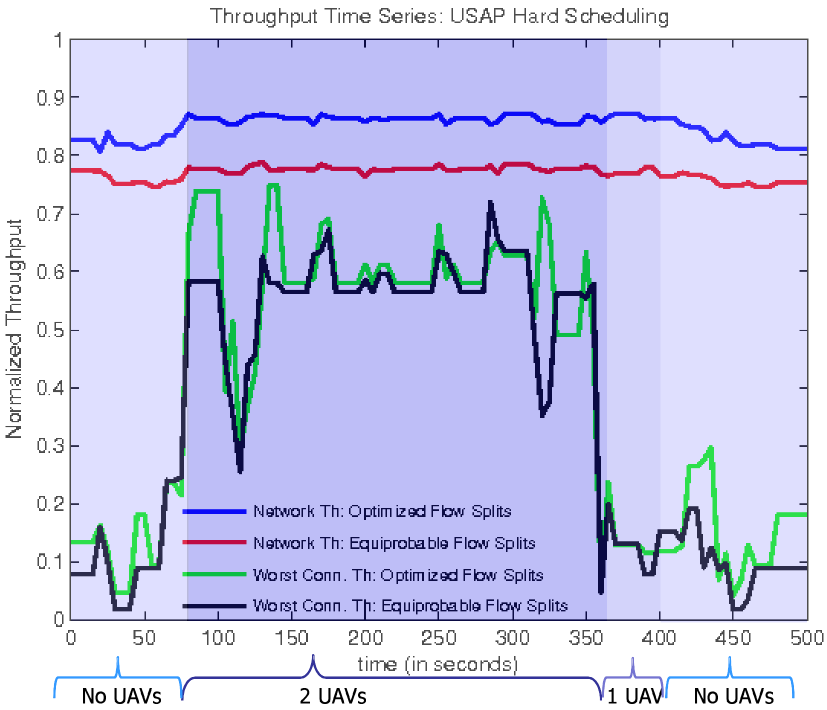

Figure 3.3: Throughput Time Series: USAP Hard Scheduling

3.6.2 Hard Scheduling With Capacity Estimation Model

We run the entire scenario first with equiprobable flow splits among the various paths of a connection and then with flow splits optimized using throughput sensitivities (section 3.5). Figure 3.3 shows the variation of total throughput and connection 11 throughput (worst connection throughput) for the two cases. Note the increase in total throughput for flow splits chosen to maximize total throughput. Also note that for some time instances, the individual throughput of the longest connection (i.e., connection 11) reduces when the flow splits are chosen to maximize total throughput. This can be explained by the fact that we maximize total throughput not individual connection throughput and reducing the longest connection throughput may increase more number of shorter (lesser hops) connection throughputs resulting in an increase in total throughput.

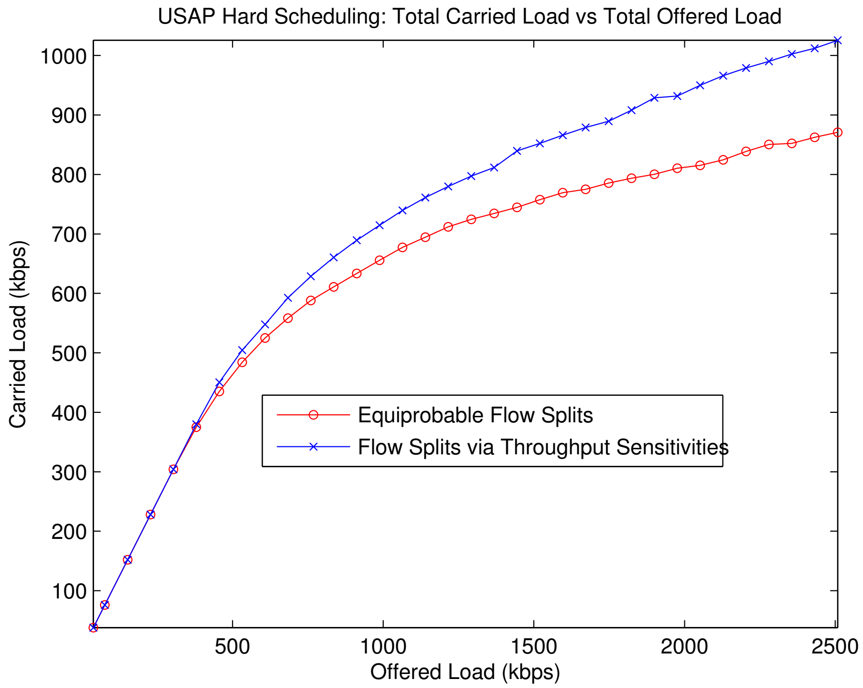

Figure 3.4: Total Carried Load vs Offered Load: USAP Hard Scheduling

To find out effects of offered load on throughput, we ran the scenario at time snapshot 0 but with all connection offered loads scaled by a common factor δ. Figure 3.4 shows the effect of offered load on total throughput for equiprobable flow splits and flow splits chosen to maximize throughput. We see that the total carried load increases when the flow splits are chosen to maximize total throughput. The total throughput/carried load in both cases saturates to some maximum value which is the maximum total capacity that the reservation based system can carry with a particular routing strategy.

3.6.3 Validation

We developed a simulation of USAP Hard Scheduling and use it to validate the reduced load loss network models. Table 3.2 compares the total throughput between

Table 3.2: USAP HS Total Throughput: Simulation VS Models

Load Factor	0.5	1.0	1.5	2.0
Sim Th	0.9742	0.8011	0.6370	0.5227
Model Th	0.9867	0.7738	0.6092	0.4983

simulation and reduced loss models for various load scaling factors. Although the total throughputs are close for both the simulation and models, the individual connection throughput varies quite a bit. Figure 3.5 compares the throughput of each connection between simulation and reduced load models (with reduced link capacity model as per section 3.4.2) . We see that there is a large mismatch for some connections (connection 0, 4, 5, 9, 10, 11). To find out if the mismatch is due to the reduced link capacity model used, we ran the reduced load loss network models using the reduced link capacity pmf obtained using simulation. Table 3.3 shows good match for the total throughput at various load scaling factors between simulation and reduced load loss network models using simulation's free capacity pmf. Figure 3.6 compares the individual connection throughput obtained in this manner against simulation. We see that the throughput matches very well for all connections. This leads us to use the reduced link capacity pmf obtained through simulation along with our reduced load loss network models and throughput sensitivity formulas (section 3.5) in order to optimize total throughput. The next section presents results of using this simulation based optimization method.

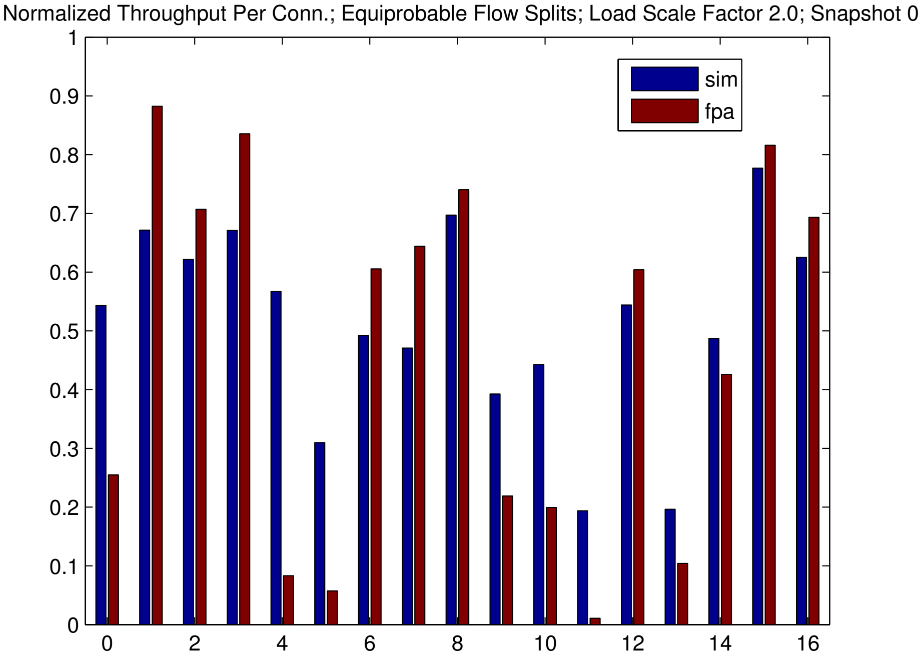

Figure 3.5: USAP Hard Scheduling Throughput: Simulation VS Models

Table 3.3: USAP HS Total Throughput: Simulation VS Models with Simulation Free Capacity PMF

Load Factor	0.5	1.0	1.5	2.0
Sim Th	0.9742	0.8011	0.6370	0.5227
Model Th (with sim's free capacity pmf)	0.9800	0.8019	0.6328	0.5171

3.6.4 Hard Scheduling With Simulation Based Capacity Estimation

Since using simulation based reduced link capacity pmf along with reduced load loss network models results in perfect match with simulation, we use simulation along with our reduced load models described in section 3.4.1 to obtain throughput sensitivities

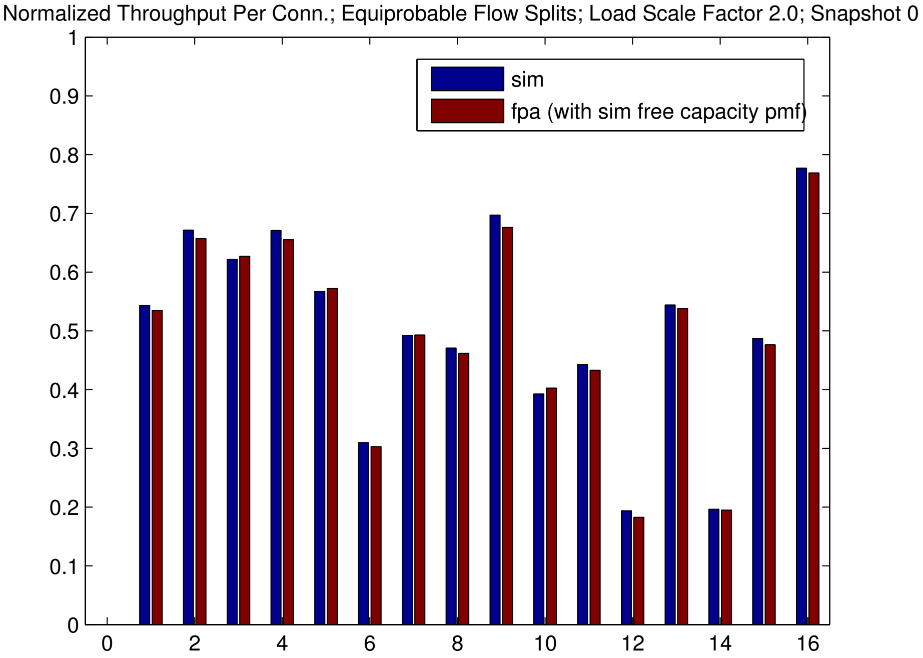

Figure 3.6: USAP Hard Scheduling Throughput: Simulation VS Model with capacity estimation pmf obtained from simulation

(section 3.5) and then optimize total throughput.

Figure 3.7 shows the variation of total throughput and worst connection throughput (i.e., connection 11) for the two cases of using equiprobable flow splits and flow splits chosen to optimize throughput. The total throughput increases with the flow splits chosen to maximize total throughput.

Figure 3.8 shows the effect of varying offered load (at time snapshot 0) on total throughput for equiprobable flow splits and flow splits chosen to maximize throughput. The total throughput in both cases saturates to some maximum value which is the maximum total capacity that the reservation based system can support.

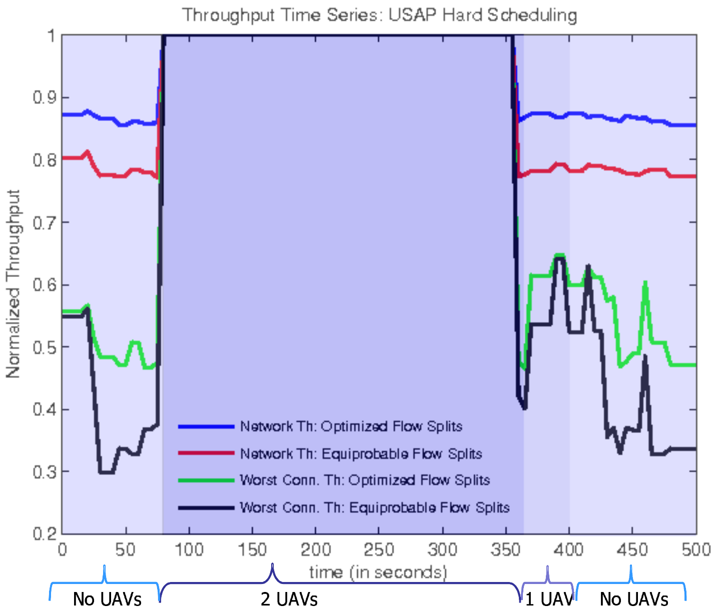

Figure 3.7: Throughput Time Series: USAP Hard Scheduling Model with simulation based capacity estimation

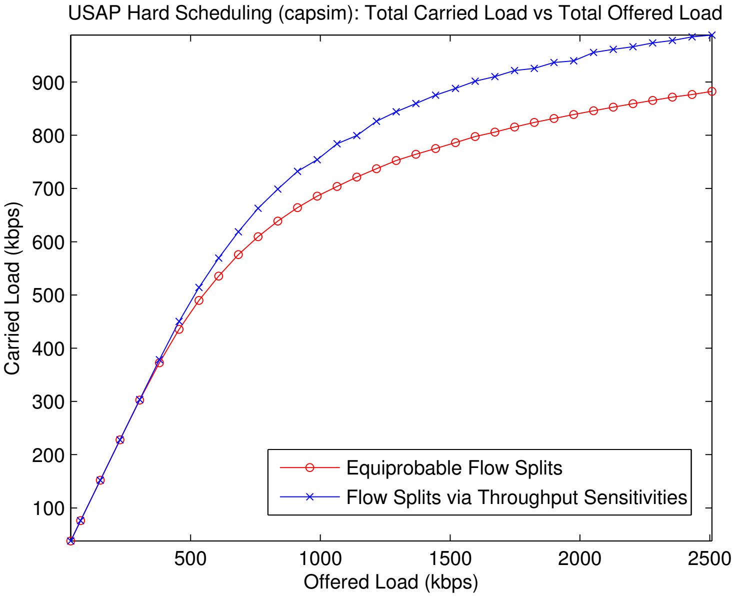

Figure 3.8: Total Carried Load vs Offered Load: USAP Hard Scheduling Model with simulation based capacity estimation

3.7 Single Channel USAP Hard Scheduling

Sections 3.6.3 and 3.6.4 show that as long as the available link capacity pmf is correctly estimated, then we can use the extended reduced load approximation for loss networks to correctly estimate the blocking probabilities via a fixed point calculation. In this section, we look at the simpler case of a USAP frame with only one frequency channel. We look at ways of estimating the link blocking probabilities for the reduced load approximation for a single channel USAP frame directly through analysis of the link neighborhood instead of first estimating the available link capacity pmf and then estimating the link blocking probability for each value of available link capacity.

3.7.1 Single Channel USAP Hard Scheduling Model

Let us consider the simpler case of a USAP frame with only one frequency channel, i.e., $F = 1$. In this case, each time slot consists of only one cell. Thus the USAP reservation rules specifying those time slots and cells that cannot be used by a transmitting node i of link $l = (i, j)$ to send traffic to node j now become rules that specify those time slots that cannot be used by node i to transmit data to node j. The USAP reservation rules are:

1. Node i cannot reserve those time slots which already have scheduled incoming or outgoing transmissions to and from i and j.

2. Node i cannot reserve those time slots already used by incoming call transmissions to the neighbors of i.

3. Node i cannot reserve those time slots already used by outgoing call transmissions from the neighbors of j.

Using these reservation rules and the links in the neighborhood of a link l, we can find those links p that cannot share slots with link l. The calls in those links that cannot transmit simultaneously have to share M slots. Thus for each link l, we can build a conflict graph whose vertices are the links that cannot share slots with link l and with edges between those links that cannot share slots with one another. From this conflict graph, we can find all the maximal cliques. The calls in link l and the links of each maximal clique should be such that the sum of the slots used is less than or equal to the total number of slots M. Thus the set of the number of valid calls (the state space $\mathcal{S}$) in the entire wireless network has to atleast satisfy these constraints for each link along each route in the network.

Assume that there are K classes of calls in the wireless network. Each class k consists of calls along a route R_k and with a particular service type (voice, video, etc.). The calls from the K classes arrive according to independent Poisson processes with rates λ_k, $k = 1, \ldots, K$. The holding times of the calls are independent of each other and independent of the Poisson arrival processes. The holding times of each class k are identically distributed with mean $1/\mu_k$. Each class k calls requires b_k cells per USAP frame so that the call demand in terms of number of cells per frame is b_k. The offered load of each class k call ρ_k is λ_k/μ_k. The number of calls in the wireless network form a Markov process (K independent birth death processes) with the state space $\mathcal{S}$ characterized as follows. For each link l in the network that has calls going through it, find the sets of maximal cliques where each clique consists of links (with calls going through it) that cannot share slots with calls in link l and that cannot be active simultaneously. For each maximal clique, the number of calls in link l and in the links of the clique cannot use more than M slots. Let $\mathcal{K}_l$ be the set of call classes that use link l. Let $\mathcal{M}_l$ (called closed maximum clique link set) denote the union of link l and a maximal clique link set around link l. For each closed maximal clique link set $\mathcal{M}_l$ around link l, we have

$$\sum_{p \in \mathcal{M}_l} \sum_{k \in \mathcal{K}_p} b_k n_k \leq M \tag{3.16}$$

Let there be C_l maximum clique link sets around link l. Denote each closed maximum clique link set as $\mathcal{M}_l^c$ $(c = 1, \ldots, C_l)$. We can combine the state space constraints for link l as

$$\sum_{k \in \mathcal{K}_l} b_k n_k + \max_{c=1,\ldots,C_l} \left(\sum_{p \in \mathcal{M}_l^c / \{l\}} \sum_{k \in \mathcal{K}_p} b_k n_k \right) \leq M \tag{3.17}$$

We use an extended reduced load loss network approximation for computing the blocking probability of a class k call in the network. The extended reduced load loss network approximation assumes that the links block independently, that the calls arrive at links along the route as independent Poisson processes, and that the cliques of a link block independently. Since the links block independently, the blocking probability, B_k, of a class k call is given by

$$B_k = 1 - \Pi_{l \in R_k}(1 - B_{k,l}) \tag{3.18}$$

where, $B_{k,l}$ is the probability of blocking a class k call on link l along route R_k.

Let there be C_l maximum closed clique link sets around link l. Since we assume that the cliques block independently, the blocking probability of a class k call at link l, $B_{k,l}$ is given by

$$B_{k,l} = 1 - \Pi_{c \in \{1,\ldots,C_l\}}(1 - B_{k,l,c}) \tag{3.19}$$

where, $B_{k,l,c}$ is the blocking probability of a class k call at the clique c of link l. Thus

$$B_k = 1 - \Pi_{l \in R_k} \Pi_{c \in \{1,\ldots,C_l\}}(1 - B_{k,l,c}) \tag{3.20}$$

To compute $B_{k,l,c}$, consider the closed maximum clique link set $\mathcal{M}_l^c$ $(c = 1, \ldots, C_l)$ around link l. Each link p in the closed clique set $\mathcal{M}_l^c$, has a closed clique set around it with the same links as in $\mathcal{M}_l^c$. $\mathcal{M}_l^c$ is a stochastic knapsack with input traffic being all the calls that use any of the links in $\mathcal{M}_l^c$. Let the number of links in $\mathcal{M}_l^c$ that are also along route R_k be $n_{k,\mathcal{M}_l^c}$. The demand of each of these calls k in $\mathcal{M}_l^c$, $b_{k,l,c}$, is the demand b_k multiplied by number of links $n_{k,\mathcal{M}_l^c}$ used in $\mathcal{M}_l^c$. The offered load of each of these calls k using any link in $\mathcal{M}_l^c$, is the source offered load thinned by blocking along all links of R_k not in $\mathcal{M}_l^c$ and by all other cliques at each of the links in $\mathcal{M}_l^c$ of R_k.

$$\rho_{k,l,c} = \frac{\lambda_k}{\mu_k} \left(\prod_{p \in R_k - \mathcal{M}_l^c} B_{k,p} \right) \left(\prod_{p \in R_k \cap \mathcal{M}_l^c} \prod_{i \in \{1,\ldots,C_p\} - a_p} B_{k,p,i} \right) \tag{3.21}$$

where a_p is the position of the clique $\mathcal{M}_l^c$ in the links belonging to $R_k \cap \mathcal{M}_l^c$. Since we assume that the links along a route block independently and the links of $R_k \cap \mathcal{M}_l^c$, i.e., links of a route R_k in the same clique should block equally, we have

$$B_{k,l,c} = 1 - \left(\sum_{s=0}^{M - b_k * n_{k,\mathcal{M}_l^c}} q_M(s) \right)^{1/n_{k,\mathcal{M}_l^c}} \tag{3.22}$$

where the $q_M(s)'s$, are the probabilities of having exactly s slots occupied in a frame with a total of M slots (i.e., knapsack occupancy probabilities). These occupancy probabilities can be computed easily using Kaufman recursion.

3.7.2 Single Channel USAP Throughput Sensitivities

The throughput sensitivity formulation is the same as that in section 3.5 and is also based on using implied costs. But the method of estimating the implied costs for a route (see equations 3.13, 3.14, 3.15) is modified to take into account the method of estimating the link blocking probabilities using cliques around the link.

The route implied cost, t_{r_s} (notation changed from c_{r_s} used in section 3.5), is approximated using the fact that the cliques block independently. Thus the implied cost for route r of connection s, t_{r_s}, is approximated to be the sum of clique implied costs along all the links l of route r of connection s and is given by

$$t_{r_s} = \sum_{l \in R_{r_s}} \sum_{c \in \{1,\ldots,C_l\}} t_{r_s,l,c} \tag{3.23}$$

A fixed point approximation procedure is used to find the clique implied costs similar to that in [12] (section 5.7). The equations are:

$$\begin{aligned} t_{r_s,l,c} * n_{r_s,\mathcal{M}_l^c} \quad = \quad & \sum_{r'_{s'} \in \mathcal{M}_l^c} \frac{\Delta_{r'_{s'},l,c}\ \rho_{r'_{s'},l,c}}{\mu_s} \left[n_{s'} - \mu_{s'} \left(\sum_{p \in R_{r'_{s'}} - \mathcal{M}_l^c} \sum_{c' \in \{1,\ldots,C_p\}} t_{r'_{s'},p,c'} + \right. \right. \\ & \left. \left. \sum_{p \in R_{r'_{s'}} \cap \mathcal{M}_l^c} \sum_{c' \in \{1,\ldots,C_p\} - a_p} t_{r'_{s'},p,c'} \right) \right] \end{aligned} \tag{3.24}$$

where,

$$\Delta_{r'_{s'},l,c} \quad = \quad Q_{r'_{s'},l,c}(M - b_{s'} * n_{r'_{s'},\mathcal{M}_l^c}) - Q_{r'_{s'},l,c}(M) \tag{3.25}$$

where, $Q_{r'_{s'},l,c}(x)$, is the blocking probability of route $r'_{s'}$ at clique c of link l where the clique has x slots. Having obtained the throughput sensitivities with respect to routing probabilities, we use the gradient projection method to find the optimal values for routing probabilities to maximize total network throughput.

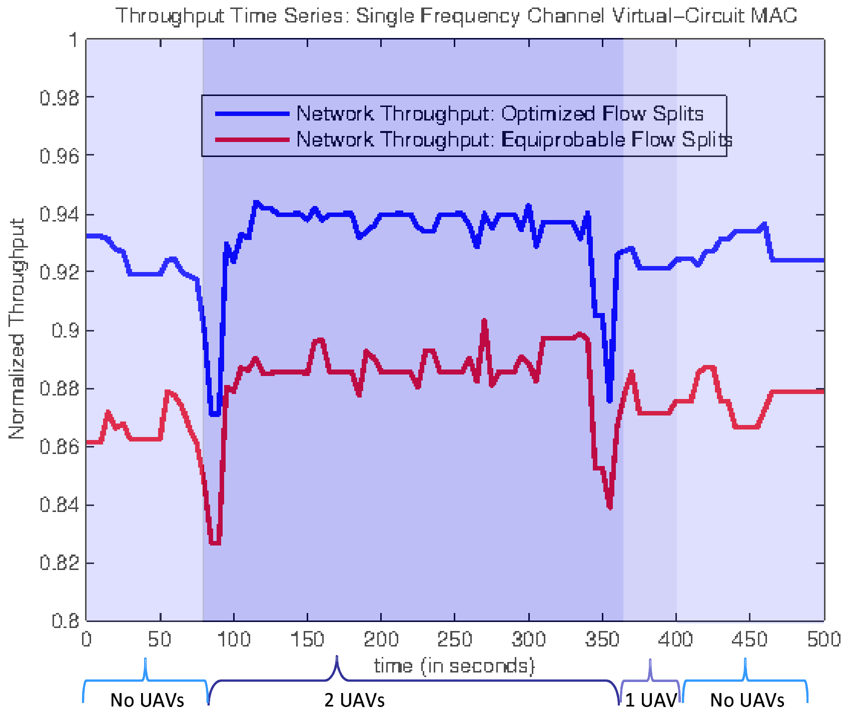

Figure 3.9: Throughput Time Series: Single Frequency USAP Hard Scheduling

3.7.3 Results and Validation

The 30 node scenario specified in 2.3 is used. There are 17 source-destination connection pairs chosen in this scenario with the traffic requirements the same as in table 3.1. The USAP parameters are also the same as described in section 3.6.1 except that the number of frequency channel F is set to 1 and the number of data time slots in a frame M is correspondingly increased to 50.

We run the entire scenario first with equiprobable flow splits among the various paths of a connection and then with flow splits optimized using throughput sensitivities (section 3.7.2). Figure 3.9 shows the variation of total throughput for the two cases. Note the increase in total throughput for flow splits chosen to maximize total throughput.

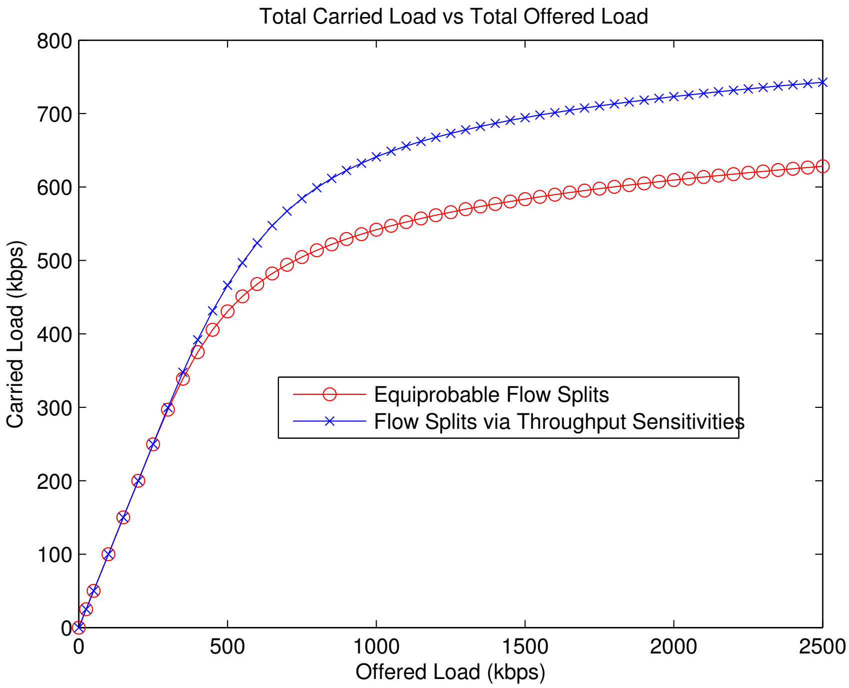

Figure 3.10: Total Carried Load vs Offered Load: Single Frequency USAP Hard Scheduling

To find out effects of offered load on throughput, we ran the scenario at time snapshot 0 but with all connection offered loads scaled by a common factor δ. Figure 3.10 shows the effect of offered load on total throughput for equiprobable flow splits and flow splits chosen to maximize throughput. We see that the total carried load increases when the flow splits are chosen to maximize total throughput. The total throughput/carried load in both cases saturates to some maximum value which is the maximum total capacity that the reservation based system can carry with a particular routing strategy.

Table 3.4: USAP HS (F=1) Total Throughput: Simulation VS Models

Load Factor	0.5	0.75	1.0	1.5	2.0
Sim Th	0.9983	0.9593	0.8742	0.6928	0.5561
Model Th	0.9985	0.9542	0.8613	0.6728	0.5415

3.7.3.1 Validation

We developed a simulation of USAP Hard Scheduling with single frequency channel and use it to validate the reduced load loss network models developed in section 3.7.1. Table 3.4 compares the total throughput between simulation and reduced loss models for single frequency channel USAP for various load scaling factors. We see that the total throughputs are close for both the simulation and models.

Figures 3.11 to 3.16 show the individual connection throughput as obtained by simulation (legend "*simulation*" in the figures), the new method (legend "*fpa*" in the figures) for various load scaling factors from 0.5 to 3.0. We see that there is a good match.

3.8 USAP used in MDL of JTRS

Mobile Data Link (MDL) ([5], [23], [24]) provides the channel access for the JTRS (Joint Tactical Radio System) Wideband Networking Waveform (WNW) using adaptive TDMA. It divides the communication channel into time and frequency cells with a portion of the cells used for management traffic and the rest used for user traffic. MDL uses the Unifying Slot Assignment Protocol (USAP) to schedule transmissions so as to achieve contention free transmissions. USAP is a dynamic distributed reservation based MAC

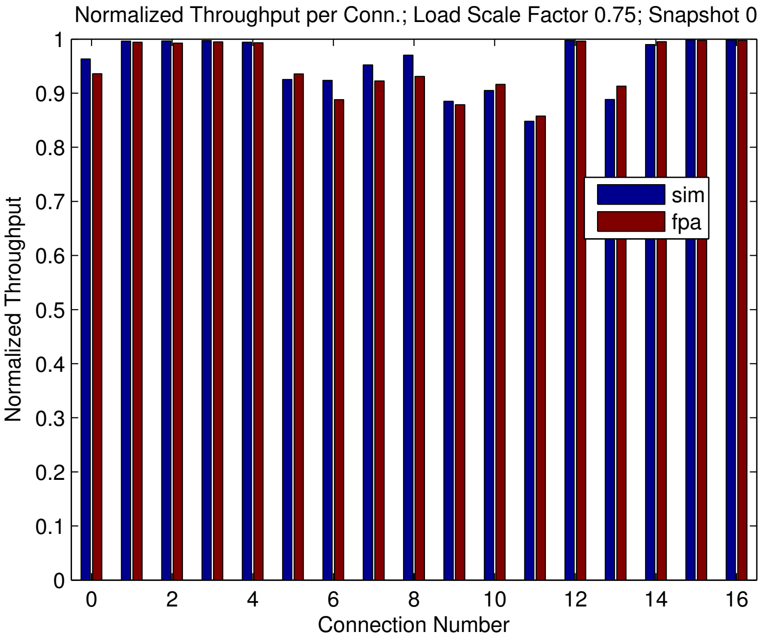

Figure 3.11: USAP Hard Scheduling F=1 (scale factor = 0.75): Simulation vs Model

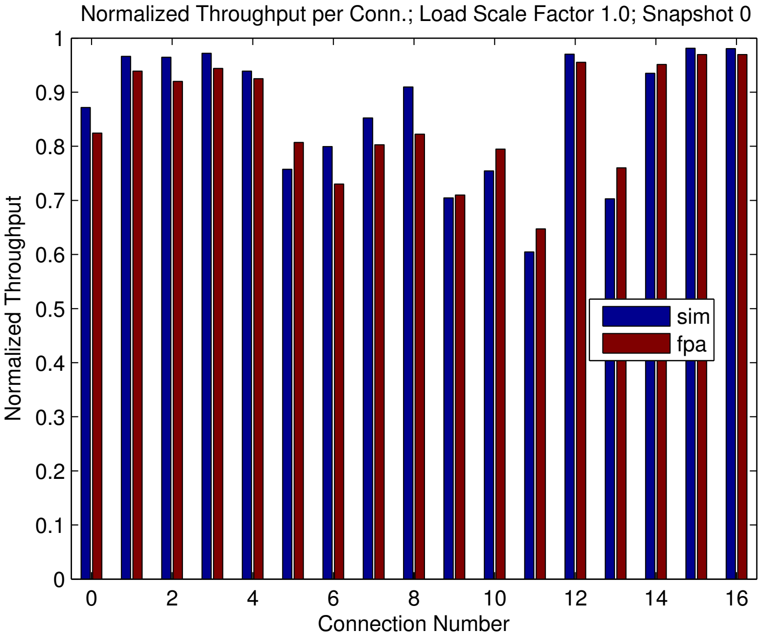

Figure 3.12: USAP Hard Scheduling F=1 (scale factor = 1.0): Simulation vs Model

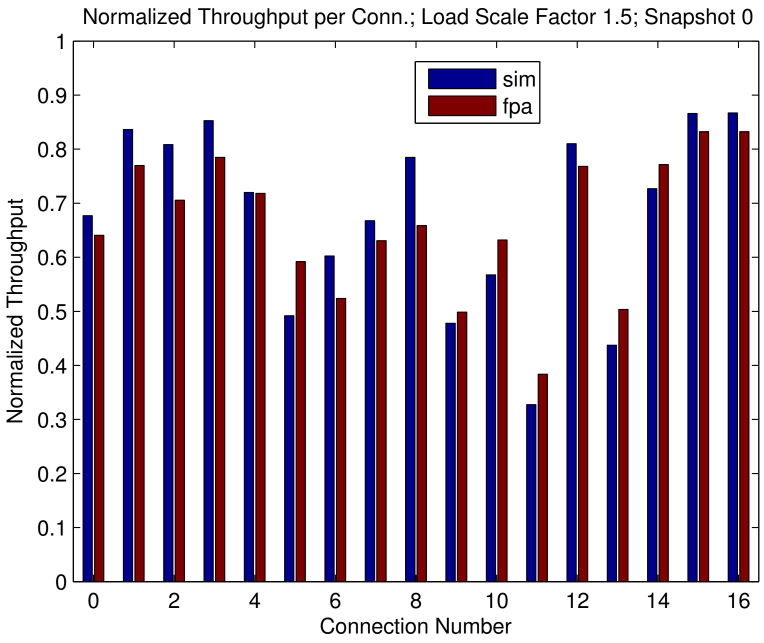

Figure 3.13: USAP Hard Scheduling F=1 (scale factor = 1.5): Simulation vs Model

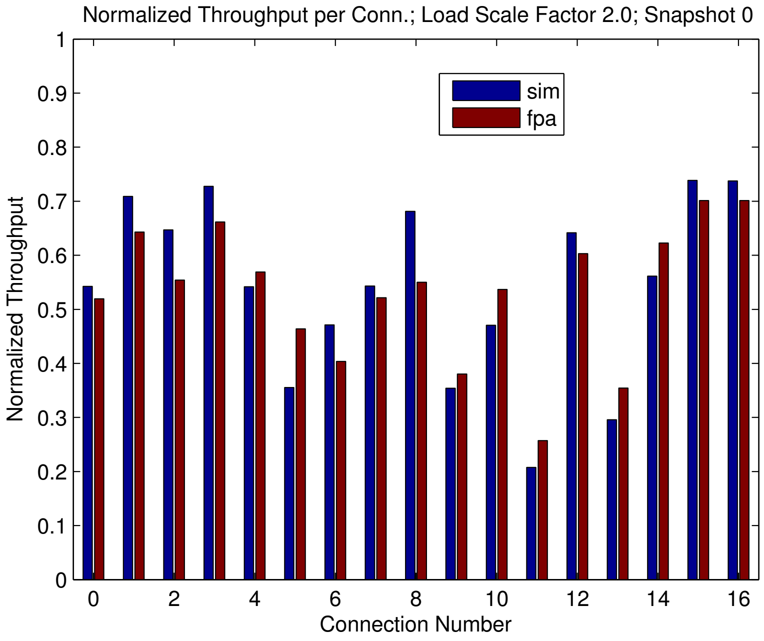

Figure 3.14: USAP Hard Scheduling F=1 (scale factor = 2.0): Simulation vs Model

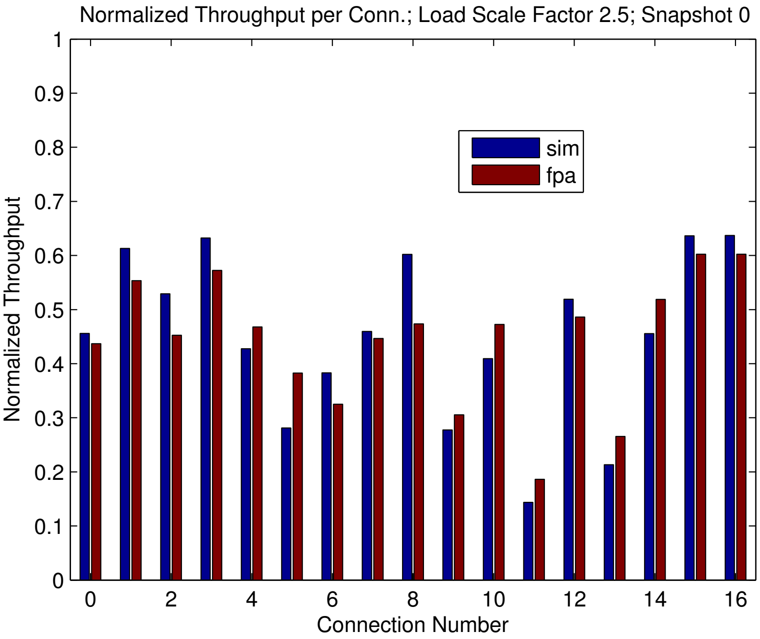

Figure 3.15: USAP Hard Scheduling F=1 (scale factor = 2.5): Simulation vs Model

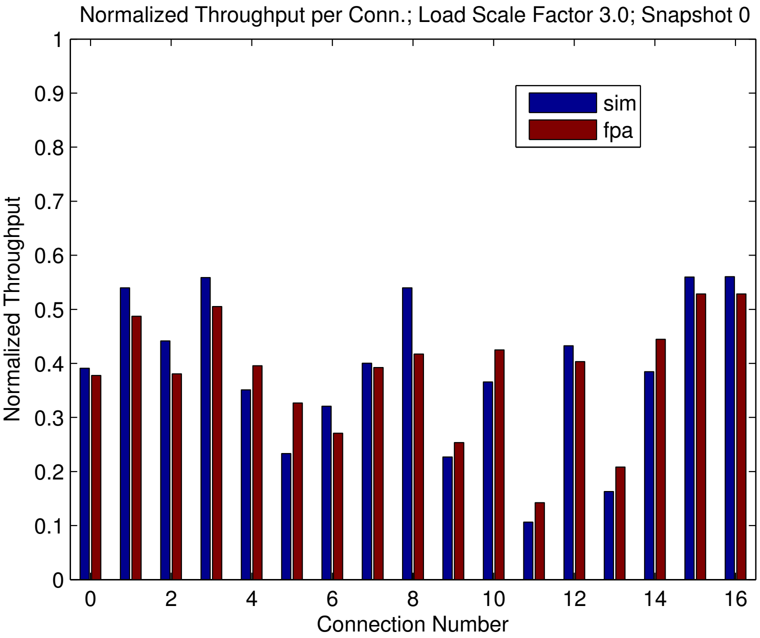

Figure 3.16: USAP Hard Scheduling F=1 (scale factor = 3.0): Simulation vs Model

which operates in two modes: hard scheduling mode (virtual circuit connection-oriented mode) where nodes reserve a session's link capacity end-to-end over the entire path; and soft scheduling mode (datagram scheduling) where nodes perform per-hop scheduling of links for single packets after the packet's arrival at the node. We look at modeling the hard scheduling mode of USAP.

MDL partitions the communication channel in time and frequency and constructs a periodic frame structure called Orthogonal Domain Multiple Access (ODMA). The MDL frame structure is shown in Fig. 3.17. The Synch, NiB and CNiB slots are used for management traffic while the RBS/FRS slots are used to send user traffic.

MDL uses a concept called Channelized Neighborhoods (CNs) which segregates nodes onto different frequency channels for spatial frequency reuse within the network. Each node is assigned to a default frequency channel called Default ODMA channel (DOC) and a node assigned to the k^{th} channel is denoted as belonging to *DOCk*. Nodes in a neighborhood that exchange a lot of traffic between one another usually belong to the same *DOCk*. Nodes belonging to a particular *DOCk* use this k^{th} frequency channel, in the portion of the frame called Rotating Broadcast Slots (RBSs), to send and receive traffic (multicast, broadcast and unicast) by *broadcasting* the traffic (all neighboring nodes have to listen to a node's transmission) amongst each other (Intra-DOCk communication). Traffic within a DOCk, i.e., Intra-DOCk communication, is routed via a set of backbone or artery nodes. These nodes are selected by a heuristic algorithm [23] in MDL to form an interconnected backbone within a CN. Ideally the artery nodes should form a minimal Connected Dominating Set (CDS) within a CN (the remaining CN nodes are one hop away from this set of artery/backbone nodes).

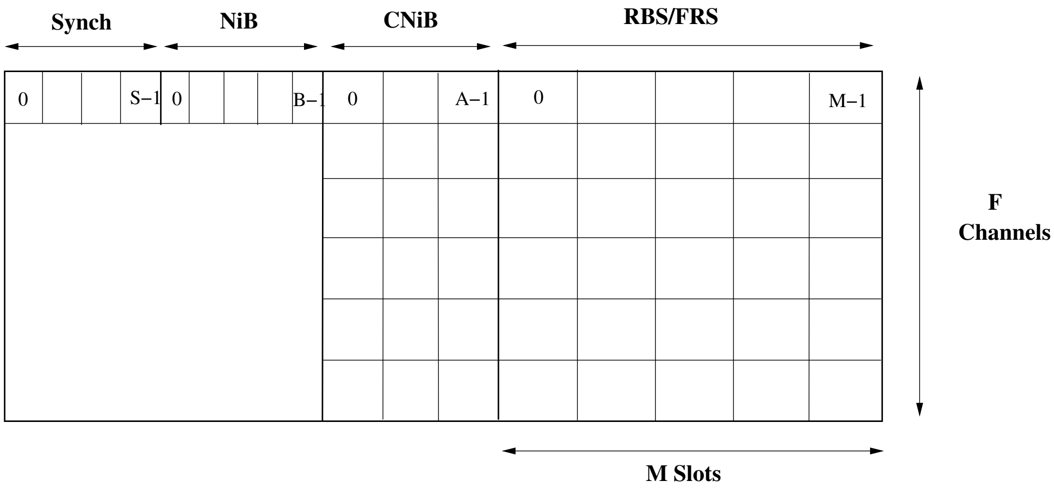

Figure 3.17: MDL TDMA Frame Structure

Traffic between two *DOCk*s (Inter-DOCk communication) is managed by setting up unicast links between nodes in the different *DOCk*s using the receiver's frequency channel (or receiver's DOCk) in the section of the frame called Fixed Reservation Slots (FRSs). FRS and RBS share the same portion of the frame with FRS given priority over RBS. So traffic that needs to be routed between two neighboring channelized neighborhoods (see Figure 3.18) $CN1$ (operating on frequency $F1$) and $CN2$ (operating on frequency $F2$) is first broadcast via backbone nodes to reach an edge or border node of CN1 using the RBS portion of the frame on frequency $F1$. Then the traffic is unicast from the edge node in CN1 to an edge node in CN2 uses FRSs on $F2$. Finally the traffic is broadcast via backbone nodes in $CN2$ using the RBS portion of the frame on frequency $F2$ to reach the intended destination/s in $CN2$.

The various slots in the periodic frame (figure 3.17) include:

1) Synch slots: are on a network-wide common channel. These slots are used to convey information needed to allow partitioned networks to merge.

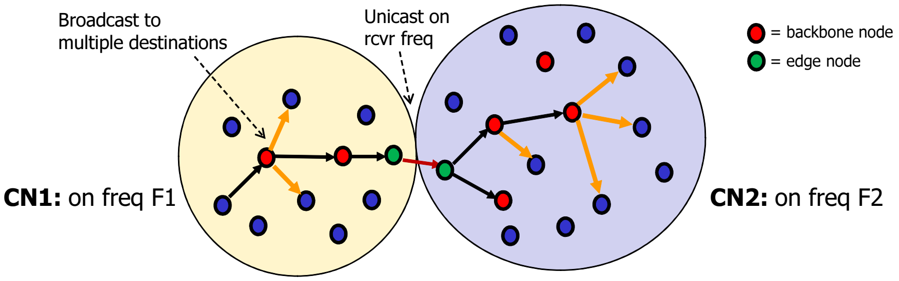

Figure 3.18: Multicast Tree Routing between CNs in MDL

2) Neighborhood Bootstrap (NiB) slots: are pre-assigned to nodes on a network-wide common channel. These slots are used to send slot assignment information (USAP records) necessary to reserve FRSs (for inter-DOCk communication). They also contain information to identify which *DOCk* a node belongs to as well as a node's CNiB slot. A node can find its entire neighborhood by hearing its neighbor's NiBs.

3) Channelized Neighborhood Bootstrap (CNiB) slots: occur on particular DOCk's frequency. These slots are used to convey USAP slot assignment information for this *DOCk*'s RBSs and USAP information used for assigning the CNiBs themselves. The CNiBs are dynamically assigned using the USAP reservation rules for broadcast.

4) Rotating Broadcast Slots (RBSs): are used to broadcast packets to all neighbors on a particular *DOCk* and are assigned via the CNiB slots. All nodes belonging to a particular *DOCk* must listen to broadcast slots on that *DOCk*, and are therefore prevented from doing anything else in that timeslot. Every RBS repeats from frame to frame but a particular RBS time slot shifts by one slot every frame period so that the RBS rotational cycle is the total number of RBS slots. This is done to somewhat mitigate the effect of FRS slots (that also repeat every frame but do not shift) which may be on a same *DOCk* channel but have higher priority.

5) Fixed Reservation Slots (FRSs): unicast packets to a specific neighbor (on the *DOCk* of the receiving node) and are assigned via the NiB slots. FRSs are primarily reservations between different *DOCk*'s (inter-DOCk); thus the idea is that these inter-DOCk connections would not require updating much faster than the NiB cycle. FRSs share the same slots as RBSs but have higher priority and override any RBSs assigned to the same slot. Since the RBS shift 1 slot every frame, the effect of the FRS on RBS traffic is somewhat mitigated.

A node listens/transmits on the common channel during Synch slots and NiB slots; and then switches to its *DOCk* for the CNiB section of the frame and for broadcast traffic in the RBSs, and switches to the receiver's *DOCk* for transmitting inter-DOCk unicast communication using FRSs.

RBS slots for a particular *DOCk* are on a specific frequency and are assigned with a 3-hop constraint if possible. The 3-hop reuse rule is included so that a node can borrow its neighbor's assigned slots that the neighbor is currently not using without violating 2-hop reuse. If no slots are available based on the 3-hop constraint rules, then the 2-hop reuse is used. We do not model borrowing of neighbor's slots and hence use the 2-hop reuse constraints. The 2-hop reuse constraint specifies those slots that cannot be used by node i to broadcast (on the RBSs) to its neighbors on frequency k (*DOCk*) and is the following:

1. Node i cannot reserve slots that already have scheduled incoming and outgoing transmissions to and from itself and *all* its neighbors.

The FRS unicast Inter-DOCk reservation is based on the following three rules for 2 hop reuse that specify those slots that cannot be used by node i to transmit to node j on node j's DOCk channel:

1. i cannot reserve those time slots which already have scheduled incoming or outgoing transmissions to and from i and j.

2. i cannot reserve slots containing incoming call transmissions (on j's DOCk) to the neighbors of i.

3. i cannot reserve those slots containing outgoing call transmissions from the neighbors of j (on j's DOCk).

These broadcast and unicast reservation rules are used to model the sharing of the wireless channel between a node and its (upto 2 hop) neighboring nodes and form the basis of our modeling of USAP as used in MDL. The next section 3.9 describes the modeling of MDL and the fixed point models used for USAP Hard Scheduling. In section 3.10 we present the time varying scenario used and results of our fixed point modeling for USAP Hard Scheduling as used in MDL of JTRS as well as its validation against simulation.

3.9 MDL and USAP Hard Scheduling Models

Let there be M time slots and F frequency channels in the RBS/FRS portion of an MDL frame (figure 3.17). We consider MANET scenarios where the nodes are divided into a set of groups (e.g., platoons). All the nodes in a group move together (i.e, form a connected sub-network) and exchange a lot of traffic amongst each other. The MANET scenario is specified as a sequence of time snapshots. At each time snapshot, node locations, src-dest traffic flows, and environmental conditions (path loss between nodes) are specified.

3.9.1 Assign Frequencies to Groups (across all time snapshots)

Since all the nodes in a group move together and exchange a lot of traffic amongst each other, all the nodes in a specific group are assigned the same frequency id (identified by an integer from 0 to $F-1$). Based on the initial location of the groups (using a group reference point location) and the total number of frequency channels F, the groups are assigned frequencies from the pool of available frequencies so that as far as possible neighboring groups have different frequency channels. If $F=1$, all groups are assigned frequency id of 0. If total number of groups $G \leq F$, then each group is assigned a different frequency id. But if $G > F$, then assign that group, which has lowest average distance to its closest $F-1$ neighboring groups (using group reference points), the frequency id of 0 and assign its closest $F-1$ neighboring groups frequency ids 1 to $F-1$. And as long as some group is not assigned a frequency id, choose that unassigned group that has lowest average distance to its closest $F-1$ groups that have been assigned different frequencies and assign it the frequency id not assigned to these $F-1$ groups.

3.9.2 Discover CNs (at each time snapshot)

A Channelized Neighborhood (CN) is a collection of nodes that share the same frequency and are connected. If the number of groups in the scenario is greater than the number of frequency channels F, then as the scenario evolves neighboring groups change and hence two groups with the same frequency channel can become disconnected (i.e., there is no path from one group to the other that passes through other groups with the same frequency channel) or connected (either directly or through other groups with the same frequency channel). Hence it is necessary to find the CNs at each time instance of the scenario. At each time snapshot, groups assigned the same frequency channel and which are connected to each other (either directly or through other groups assigned the same frequency channel) form a single CN.

3.9.3 Find Artery Nodes within Channelized Neighborhood

Traffic within a Channelized Neighborhood (CN) is routed through a set of Artery Nodes (ANs). These artery nodes are chosen to form a Connected Dominating Set (CDS) so that any CN node not in this set is a neighbor of a node in the CDS. We use Algorithm I of Guha and Khuller [25] to form a CDS that approximates a Minimal Connected Dominating Set (MCDS). An MCDS is a CDS that has a minimum number of nodes. Algorithm I of Guha and Khuller constructs the CDS from one node outward and yields a CDS of size at most $2\,(1 + H(\Delta))\,|OPT|$, where H is the harmonic function, and OPT refers to an optimal solution, i.e., a MCDS.

3.9.4 Find Edge Nodes of Channelized Neighborhoods

If the artery nodes of neighboring connected CNs are not connected to each other, edge nodes need to be added to the CNs (as necessary) in order to route flows across the CNs. We use a simple heuristic to add the required edge nodes. Consider two neighboring channelized neighborhoods CN1 and CN2 that are connected via some nodes. If any artery node of CN1 is connected to any artery node of CN2, we do not add edge nodes. If the artery nodes of the two CNs are not directly connected but some artery node of CN1 (or CN2) is connected to some non-artery node of CN2 (or CN1), we add this non-artery node of CN2 (or CN1) as an edge node. Finally if none of the artery nodes of CN1 or CN2 are directly connected to any node of CN2 or CN1 respectively, we add the closest connected nodes of CN1 and CN2 as edge nodes.

3.9.5 Construct Multicast Routing Tree and Unicast Routing Path

Once artery nodes in each CN are chosen and edges nodes chosen (if necessary) to connect neighboring CNs, traffic from a source is routed through these nodes, either to a set of receivers (defining a multicast group) in the case of multicast traffic or to a single destination in the case of unicast traffic. For multicast traffic, we construct a Steiner tree from the source to the multicast group receivers using as Steiner points (i.e., intermediate nodes) the set of artery nodes and edges nodes selected. In the Steiner tree problem, given a graph $G(V, E)$, and a set $R \subseteq V$ of required nodes, we want to find a minimum cost tree connecting all nodes in R. The set of nodes R includes the source and the multicast group receivers while the set V includes R, the artery nodes and the edge nodes. We use the heuristic proposed in [26] (called the KMB heuristic) to construct the Steiner tree. The KMB heuristic has a performance guarantee of at most twice the size of the optimum Steiner tree. For unicast traffic, we use the shortest path between the source and the destination to route traffic using as intermediate nodes the chosen artery and edge nodes.

3.9.6 Modeling USAP Hard Scheduling Mode for Multicast and Unicast Traffic

For a source transmitting to a multicast group, the traffic is routed over a multicast tree (Steiner tree) with intermediate nodes being the artery nodes and edge nodes. For a source unicasting to a single destination, the traffic is routed over the shortest path going through the artery nodes and edge nodes. We can consider this shortest path for unicast traffic also as a tree with the root being the source and with only a single leaf comprising the single destination. Hence let there be G destination groups $M_1, \ldots, M_G$ (consisting of either a set of receiving nodes or a single destination). We assume that calls originate at source s for group M_g (routed via tree $T(s, M_g)$) as a Poisson process with rate $\lambda_{T(s,M_g)}$, with each call holding time having finite mean $1/\mu_{T(s,M_g)}$, and with the call demand being $b_{T(s,M_g)}$ cells per frame. The children of node i in tree $T(s, M_g)$ can be on different CNs and hence node i needs to transmit on all the frequencies corresponding to the CNs that its children belong to.

For convenience, we again list the USAP MDL reservation rules that have been specified in section 3.8. For intra-DOCk communication on the RBSs, the 2-hop reuse constraint specifies those slots that cannot be used by node i to broadcast (on the RBSs) to its neighbors on frequency k (*DOCk*) and is the following:

1. Node i cannot reserve slots that already have scheduled incoming and outgoing transmissions to and from itself and *all* its neighbors.

The FRS unicast Inter-DOCk reservation is based on the following three rules for 2 hop reuse that specify those slots that cannot be used by node i to transmit to node j on node j's DOCk channel:

1. i cannot reserve those time slots which already have scheduled incoming or outgoing transmissions to and from i and j.

2. i cannot reserve slots containing incoming call transmissions (on j's DOCk) to the neighbors of i.

3. i cannot reserve those slots containing outgoing call transmissions from the neighbors of j (on j's DOCk).

We do not model the priority of FRS over RBS and assume that when a node in a CN with frequency f_1 unicasts to a node in a neighboring CN with frequency f_2, the transmission uses the same time slots as the RBS with same priority. So as to have no traffic loss, we assume that all the source's neighboring nodes on the receiver's DOCk need to listen to this transmission.

These intra-DOCk and inter-DOCk reservation rules can be used to find those (upto 2 hop) neighboring nodes *j* transmitting on frequencies *h* that cannot share slots with node *i* transmitting on frequency *f*. We can think of a (node i, frequency f) pair as similar to a link. The calls in those (node, frequency) pairs that cannot transmit simultaneously have to share *M* slots. Thus for each node *i* transmitting on frequency *f*, we can build a conflict graph whose vertices are those transmitting nodes j on frequency h, i.e, those (node j, frequency h) pairs, that cannot share slots with (node i, frequency f) and with edges between those node-frequency pairs that cannot share slots with one another. From this conflict graph, we can find all the maximal cliques. The calls in (node i, frequency f) and the (node j, frequency h) pairs of each maximal clique should be such that the sum of the slots used is less than or equal to the total number of slots M. Thus the set of the number of valid calls (the state space $\mathcal{S}$) in the entire wireless network has to atleast satisfy these constraints for each (node i, frequency f) pair along each routing tree $T(s, M_g)$ ($g = 1 \ldots G$) in the network.

A node i belongs to a particular Channelized Neighborhood and let us denote the Channelized Neighborhood of node i as $CN(i)$. Using the intra-DOCk and inter-DOCk reservation rules, we can find those (node j, frequency h) pairs that cannot transmit simultaneously with (node i, frequency f). If node i on $CN(i)$ transmits on frequency $f = CN(i)$, then using the intra-DOCk and inter-DOCk reservation rules, node i transmitting on $CN(i)$ cannot share slots with the following node-frequency pairs:

- Neighbor j (of node i) transmitting on any frequency where node j's CN frequency is the same as $CN(i)$.

- Neighbors j (of node i) transmitting on $CN(i)$ where j's CN frequency is not the same as $CN(i)$.

- Strict 2-hop neighbor k (of node i) transmitting on $CN(i)$ where neighbor k is a neighbor of node j with node j being a neighbor of node i and j's CN frequency is the same as node i, i.e., $CN(j) = CN(i)$.

If node i on $CN(i)$ transmits on frequency f where $f \neq CN(i)$, then using the intra-DOCk and inter-DOCk reservation rules, node i transmitting on $f \neq CN(i)$ cannot share slots with the following node-frequency pairs:

- Neighbor j (of node i) transmitting on $CN(i)$.

- Neighbor j (of node i) transmitting on any frequency where node j's CN frequency is the same as f, i.e, $CN(j) = f$.

- Neighbor j (of node i) transmitting on f where node j's CN frequency is not equal to f but there is a common neighbor k of i and j with $CN(k) = f$.

- Strict 2-hop neighbor k (of node i) transmitting on f where neighbor k is a neighbor of node j with node j being a neighbor of node i and j's CN frequency is equal to f.

We have assumed that there are G destination groups $M_1, \ldots, M_G$. We assume that calls arrive at source s for group M_g (routed via tree $T(s, M_g)$) as independent Poisson processes with rates $\lambda_{T(s,M_g)}$. Let there be K routing trees $T(s, M_g)$. The holding times of the calls are independent of each other and independent of the Poisson arrival processes. The holding time of calls arriving at source s for group M_g are identically distributed with mean $1/\mu_{T(s,M_g)}$. The call demand of each call arriving at source s for group M_g is assumed to be $b_{T(s,M_g)}$ cells per frame. The offered load, $\rho_{T(s,M_g)}$ of a call arriving at source s for group M_g is $\lambda_{T(s,M_g)}/\mu_{T(s,M_g)}$. The number of calls in the wireless network form a Markov process (K independent birth death processes) with the state space $\mathcal{S}$ characterized as follows. For each (node i, frequency f) pair in the network that has calls going through it, find the sets of maximal cliques where each clique consists of (node j, frequency h) pairs (with calls going through it) that cannot share slots with calls in (node i, frequency f) pair and that cannot be active simultaneously. For each maximal clique, the number of calls in (node i, frequency f) pair and the (node j, frequency h) pairs of the clique cannot use more that M slots. Let $\mathcal{K}_{i(f)}$ be the set of call classes that go through node i transmitting on frequency f. Let $\mathcal{M}_{i(f)}$ (called closed maximum clique set) denote the union of (node i, frequency f) and a maximal clique node-frequency pair set around node i transmitting on frequency f. For each closed maximal clique set $\mathcal{M}_{i(f)}$ around (node i, frequency f), we have

$$\sum_{j(h)\in\mathcal{M}_{i(f)}} \sum_{k\in\mathcal{K}_{j(h)}} b_k n_k \leq M \tag{3.26}$$

Let there be $C_{i(f)}$ maximum clique sets around node i transmitting on frequency f (i.e, $i(f)$). Denote each closed maximum clique set as $\mathcal{M}^c_{i(f)}$ ($c = 1, \ldots, C_{i(f)}$). We can combine the state space constraints for node i transmitting on frequency f as

$$\sum_{k\in\mathcal{K}_{i(f)}} b_k n_k + \max_{c=1,\ldots,C_{i(f)}} \left(\sum_{j(h)\in\mathcal{M}^c_{i(f)}/\{i(f)\}} \sum_{k\in\mathcal{K}_{j(h)}} b_k n_k \right) \leq M \tag{3.27}$$

We use an extended reduced load loss network approximation modified to take into account multicast for computing the blocking probability of a class k (defined by a tree $T(s, M_g)$) call in the network. The extended reduced load loss network approximation assumes that the node-frequency pairs (similar to a link) block independently, that the calls arrive at a node-frequency pair along the route as independent Poisson processes, and that the cliques around a node-frequency pair block independently. Since the node-frequency pairs block independently, the blocking probability, $B_{T(s,M_g)}$ of a call traversing tree $T(s, M_g)$, is given by

$$B_{T(s,M_g)} = 1 - \prod_{i\in T(s,M_g), f\in[0,F-1]} (1 - B_{i(f),T(s,M_g)}) \tag{3.28}$$

where, $B_{i(f),T(s,M_g)}$ is the blocking probability of a call at node i of tree $T(s, M_g)$ with node i transmitting on frequency f.

We assume that there are $C_{i(f)}$ maximum clique sets around node i transmitting on frequency f. Since we assume that the cliques block independently, the blocking probability of a call at node i of tree $T(s, M_g)$ transmitting on frequency f is given by,

$$B_{i(f),T(s,M_g)} = 1 - \prod_{c \in 1,\ldots,C_{i(f)}} (1 - B_{i(f),T(s,M_g),c}) \tag{3.29}$$

where, $B_{i(f),T(s,M_g),c}$ is the blocking probability of a call traversing tree $T(s, M_g)$ at clique c of node i transmitting on frequency f. Thus,

$$B_{T(s,M_g)} = 1 - \prod_{i \in T(s,M_g), f \in [0,F-1]} \prod_{c \in 1,\ldots,C_{i(f)}} (1 - B_{i(f),T(s,M_g),c}) \tag{3.30}$$

To compute $B_{i(f),T(s,M_g),c}$, consider the closed maximal clique set $\mathcal{M}^c_{i(f)}$ ($c = 1, \ldots, C_{i(f)}$) around node i transmitting on frequency f. Each (node j, frequency h) pair in the closed maximal clique set $\mathcal{M}^c_{i(f)}$, has a closed maximal clique set around (node j, frequency h) with the same node-frequency pairs as in $\mathcal{M}^c_{i(f)}$. $\mathcal{M}^c_{i(f)}$ is a stochastic knapsack of capacity M with input traffic being all the calls that use any of the node-frequency pairs in $\mathcal{M}^c_{i(f)}$. Let the number of node-frequency pairs in $\mathcal{M}^c_{i(f)}$ that are also along tree $T(s, M_g)$ be $n_{T(s,M_g),\mathcal{M}^c_{i(f)}}$. The demand, $b_{T(s,M_g),\mathcal{M}^c_{i(f)}}$, of each of these calls along $T(s, M_g)$ in $\mathcal{M}^c_{i(f)}$ is the demand $b_{T(s,M_g)}$ multiplied by the number of node-frequency pairs $n_{T(s,M_g),\mathcal{M}^c_{i(f)}}$ of tree $T(s, M_g)$ in $\mathcal{M}^c_{i(f)}$. The offered load of each of these calls along tree $T(s, M_g)$ using any node-frequency pair in $\mathcal{M}^c_{i(f)}$ is the source offered load thinned due to blocking along all node-frequency pairs of tree $T(s, M_g)$ that are not in $\mathcal{M}^c_{i(f)}$ and also due to blocking by all other cliques at each of the node-frequency pairs in $\mathcal{M}^c_{i(f)}$ that are in tree $T(s, M_g)$. Thus

$$\begin{aligned}\rho_{T(s,M_g),\mathcal{M}^c_{i(f)}} &= \frac{\lambda_{T(s,M_g)}}{\mu_{T(s,M_g)}}\left(\prod_{j(h)\in T(s,M_g)-\mathcal{M}^c_{i(f)}} B_{j(h),T(s,M_g)}\right) \\ &\quad \left(\prod_{j(h)\in T(s,M_g)\cap\mathcal{M}_{i(f)^c}} \; \prod_{d\in\{1,\ldots,C_{j(h)}\}-a_{j(h)}} B_{j(h),T(s,M_g),d}\right) \qquad (3.31)\end{aligned}$$

where $a_{j(h)}$ is the position of the clique $\mathcal{M}^c_{i(f)}$ in the node-frequency pairs $j(h)$ belonging to $T(s, M_g) \cap \mathcal{M}_{i(f)^c}$. Since we assume that the node-frequency pairs along a route block independently and the node-frequency pairs of $T(s, M_g) \cap \mathcal{M}_{i(f)^c}$, i.e., node-frequency pairs of a route $T(s, M_g)$ in the same clique should block equally, we have

$$B_{i(f),T(s,M_g),c} = 1 - \left(\sum_{s=0}^{M-b_{T(s,M_g)}*n_{T(s,M_g),\mathcal{M}^c_{i(f)}}} q_M(s)\right)^{1/n_{T(s,M_g),\mathcal{M}^c_{i(f)}}} \qquad (3.32)$$

where the $q_M(s)'s$, are the probabilities of having exactly s slots occupied in a frame with a total of M slots (i.e., knapsack occupancy probabilities). These occupancy probabilities can be computed easily using Kaufman recursion.

3.10 MDL and USAP Hard Scheduling Results and Validation

3.10.1 Scenario with 60 nodes divided into 5 groups

The scenario considered is a time varying fast moving network of 60 vehicles divided into 5 groups (platoons) of 12 vehicles each that move towards a specific rendezvous point. Figure 3.19 shows the movement of the five groups over the entire scenario of 575 seconds. The vehicles move at speeds between 22 to 60 miles per hour. Initially all the 5 groups are connected. Groups 3 (nodes numbered from 25 to 36), 4 (nodes numbered from 37 to 48), and 5 (nodes numbered from 49 to 60) start moving immediately with groups 1 (nodes numbered from 1 to 12) and 2 (nodes numbered from 13 to 24) following groups 4 and 5 respectively after some initial amount of time (120 seconds). The groups have to go around two hills in their paths and hence groups 3, 4, and 5 lose connectivity with each other. Three Aerial Platforms (APs) (similar to MQ-1 predator (UAV)) then need to be brought in so that communication between the platoons is maintained at all times. We use a Deterministic Annealing algorithm [27] to determine the AP locations for full network connectivity.

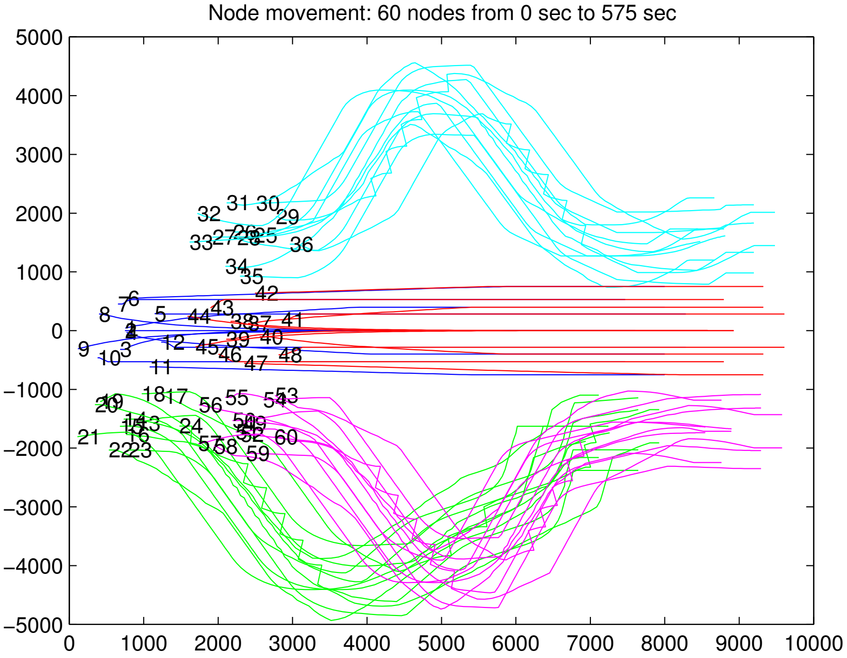

Figure 3.19: Movement of 5 groups (with 12 nodes each) for 575 seconds

The scenario is specified every 5 seconds (the ground nodes move an average of 100 meters in 5s). At every 5 second interval the following are input to the USAP Hard Scheduling model: ground node positions, traffic demands (offered load), traffic routes, environment conditions. All ground nodes and APs have identical omni-directional radios with receiver sensitivity of -95dBm, receiver threshold of 10dB, and transmit power of 5W. The environment is modeled as a fading channel with a $1/R^{\alpha}$ power attenuation. The radio specification and the path loss exponent α together determine a maximum connectivity distance between nodes. α is taken to be 4.5 between ground nodes, 3.9 between ground and aerial nodes, and 3.0 between the aerial nodes. The radio specification and path loss α result in a maximum connectivity distance of 857m between ground nodes, 2423m between ground-aerial nodes, and 25099m between aerial nodes. The maximum channel rate between any two nodes is set to 1 Mbps.

There are 6 multicast groups considered: 5 of them are intra-cluster/group and each multicast group include all the nodes of that cluster/group. The sixth multicast group spans all the 5 scenario groups and includes 2 nodes in each group for a total of 10 nodes. This multicast group models communication between commanders of the groups (platoons). All IERs, i.e., traffic flows in the scenario are assumed to be voice (using 1 cell per frame) with a holding time of 2 minutes. The traffic is chosen so that 70 percent of total offered traffic are from multicast flows while the rest are from unicast flows. There are 15 multicast IERs: 10 of them intra-group (with arrival rate of 1.5 calls/minute) and 5 inter-group (with arrival rate of 0.5 calls/minute). There are 5 unicast intra-group IERs with arrival rate of 1.5 calls/minute.

3.10.2 USAP Parameters

The USAP frame period is set to 125ms and the combined capacity of all the frequency channels is set to 1 Mbps. Only half of the USAP frame period is used for the

RBS/FRS slots. The RBS/FRS portion of the frame has a total of 50 cells. Based on the total number of cells in the RBS/FRS portion, the capacity of all channels and the fraction of the frame used for RBS/FRS, 1250 bits can be carried per cell. Hence for a connection to have a call demand ($n_{T(s,M_g)}$) of 1 reservation slot per frame, the call demand rate (for e.g., the voice coder rate) is 10 kbps. We assume that the voice coder rate is 10 kbps (hence voice calls use 1 reservation cell per frame) and the voice coder frame period is 125ms. All chosen traffic flows have a call demand of 1 reservation slot per frame.

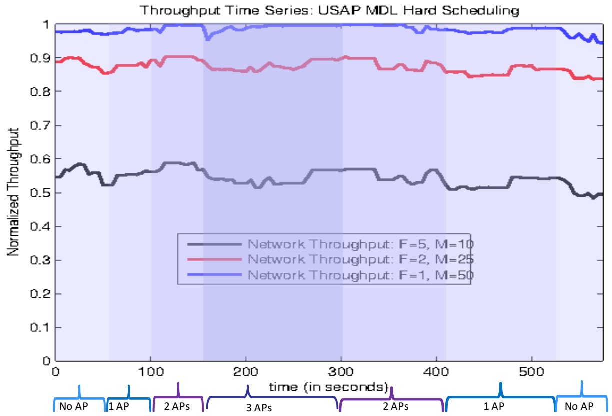

Figure 3.20: Total Network Throughput for various combinations of F and M

3.10.3 Results and Validation of USAP MDL Hard Scheduling Model

Figure 3.20 shows the total network throughput using the models developed for the scenario described previously as the total number of cells is held constant at 50 but the number of time slots M (and correspondingly the number of frequency channels F) is changed. We observe that since the traffic is mostly intra-group, the network throughput increases as the number of time slots M increases.

To find out the effect of offered load on throughput, we ran the scenario at time snapshot 0 but with all connections' offered load scaled by a common scale factor δ. Figure 3.21 shows the effect of offered load on total throughput using the developed models for various combinations of F and M with the total number of cells held constant at 50. We see that the total carried load in all cases saturates to some constant as offered load is increased. With most of our traffic being intra-cluster/group, we observe that the carried load increases as the the number of time slots M increases.

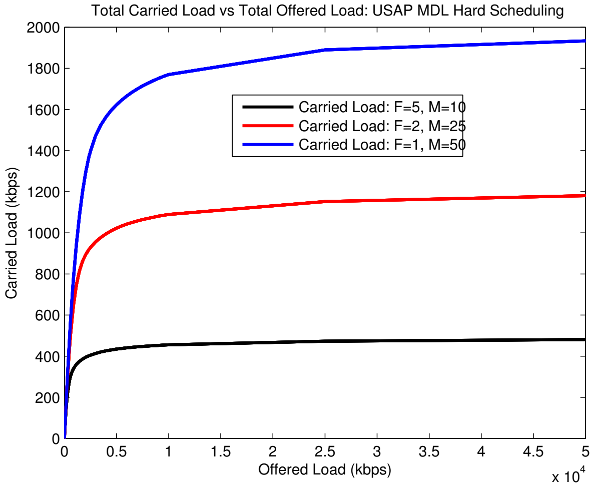

Figure 3.21: Total Carried Load vs Total Offered Load for various (F, M)

Figures 3.22 and 3.23 compare individual connection throughput of the model against simulation (developed in C++) for $F = 1$, $M = 50$ for scale factors 2.0 and 3.0 respectively. We see that the model matches the simulation well. Similarly figures 3.24 and 3.25 compare individual connection throughput of the model against simulation for $F = 2$, $M = 25$ for scale factors 1.0 and 2.0 respectively; and figures 3.26 and 3.27 compare individual connection throughput of the model against simulation for $F = 5$, $M = 10$ for scale factors 0.5 and 0.75 respectively. We see that the model matches the simulation well for different values of F and M.

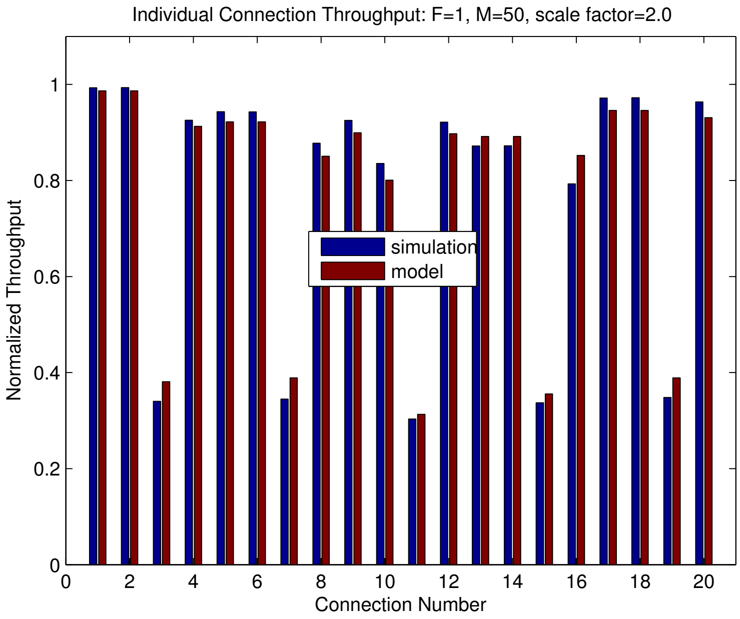

Figure 3.22: Simulation vs. Model: $F = 1$, $M = 50$ (scale factor = 2.0)

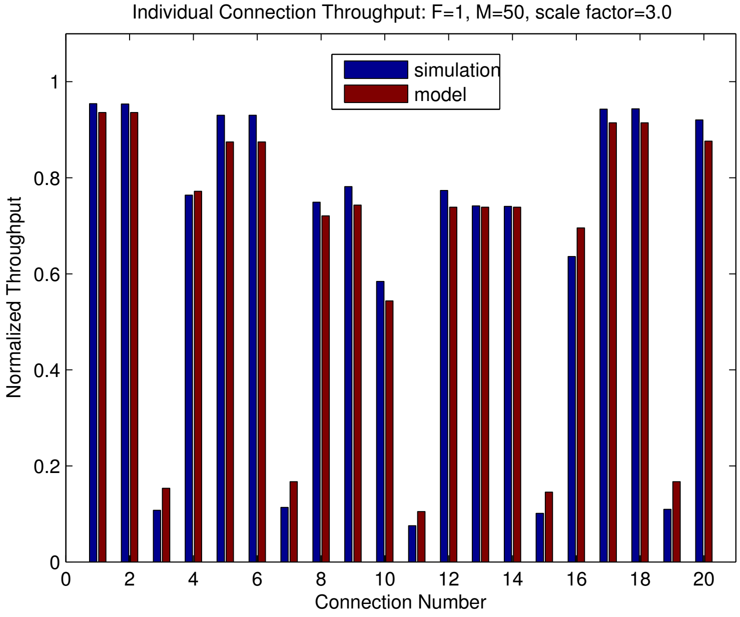

Figure 3.23: Simulation vs. Model: $F = 1$, $M = 50$ (scale factor = 3.0)

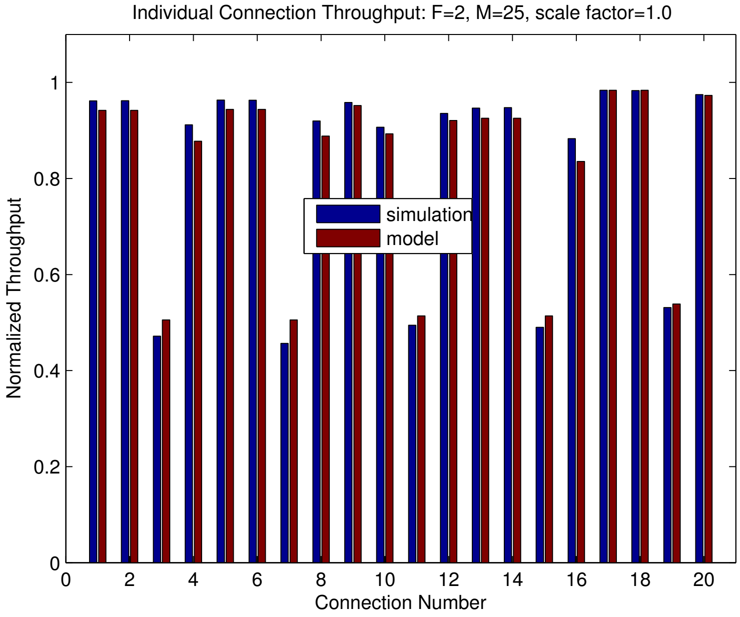

Figure 3.24: Simulation vs. Model: $F = 2$, $M = 25$ (scale factor = 1.0)

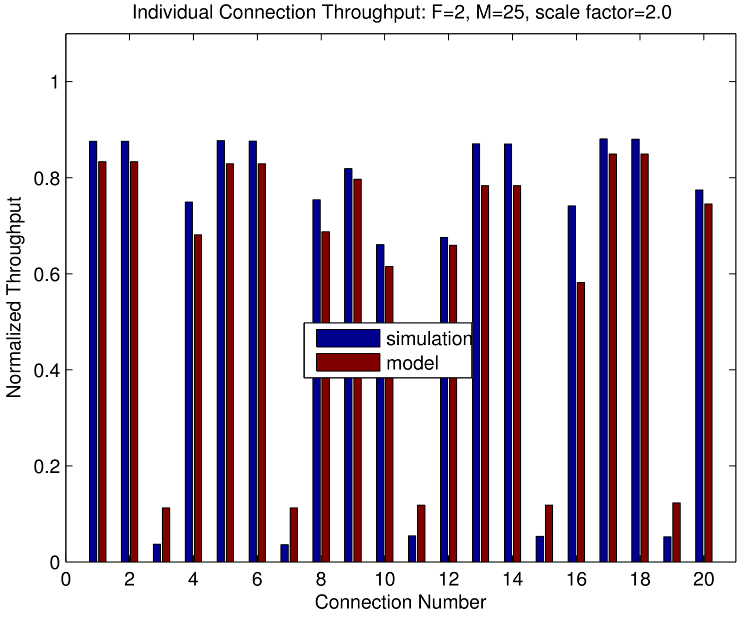

Figure 3.25: Simulation vs. Model: $F = 2$, $M = 25$ (scale factor = 2.0)

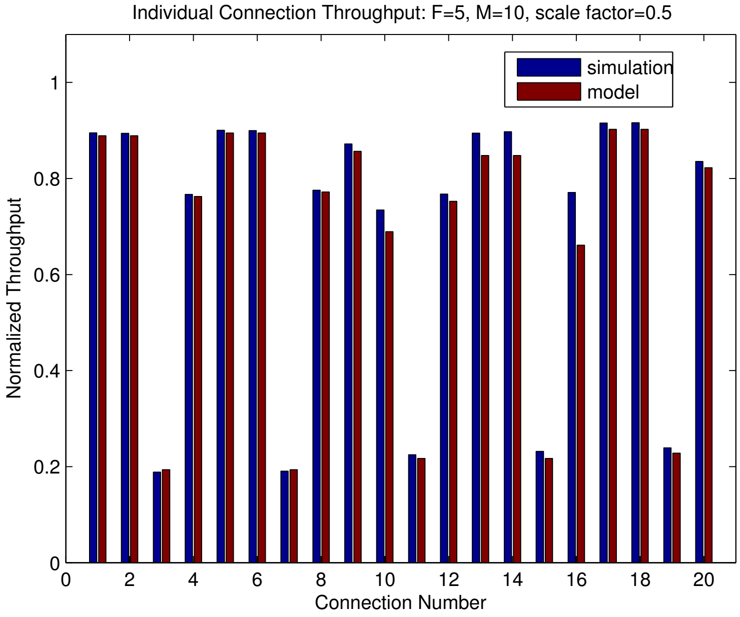

Figure 3.26: Simulation vs. Model: $F = 5$, $M = 10$ (scale factor = 0.5)

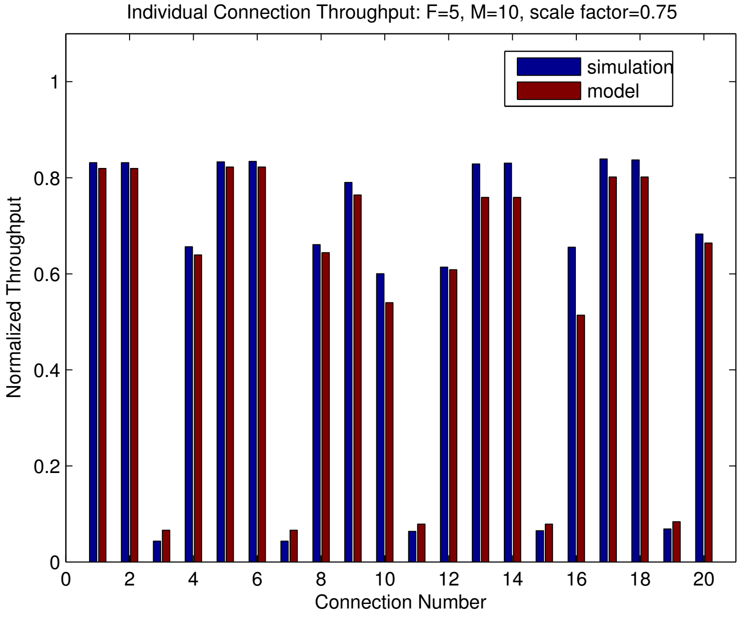

Figure 3.27: Simulation vs. Model: $F = 5$, $M = 10$ (scale factor = 0.75)

Chapter 4

Topology Control via Addition of Aerial Platforms

In this chapter, we look at the problem of adding a minimum number of Aerial Platforms (APs) to connect disconnected MANET clusters while also satisfying required traffic capacity between disconnected clusters. We also extend the connectivity solution in order to make the network single AP survivable.

4.1 Introduction and Related Work

There has been an enormous amount of research into wireless ad-hoc networks focusing on improving routing, scheduling, throughput, QoS, auto-configuration, etc. Most of these works assume that the nodes form a connected network. But this assumption may not always hold true. Connectivity among nodes depends on several factors including node density, transmission power, transmitter and receiver characteristics, propagation path loss, node mobility, area of deployment, etc. Hence it is highly probable that a MANET has nodes that are disconnected from each other. Also in the case of military or disaster relief networks, the operational scenario may be such that there are disconnected clusters of nodes (where the nodes within a cluster can communicate with each other but the different clusters are too far apart to be connected) but still there is need for communication between the different clusters. One of the methods suggested to improve connectivity, capacity, robustness, and survivability of MANETs is to use Aerial

Platforms (APs) as relays in the network [28–30]. While connectivity between different clusters can be bridged by adding additional ground resources to act as relays, this may not always be feasible due to the various factors including the nature of the terrain (steep hill or water body) or operational considerations. Furthermore, Aerial Platforms like Unmanned Aerial Vehicles (UAVs), helicopters, blimps, etc. have inherent advantages like rapid deployment to the theater of operation, range extension and beyond line of sight capability to provide connectivity between disconnected clusters, and the capability to support new services [31–34].

Addition of a network of APs can enable a fragmented communication network consisting of disconnected clusters to become connected and provide the capacity needed for communication between the various disconnected source-destination pairs. Even in the case when all the nodes can communicate with other other (either through single or multiple hops), APs can be used to increase the capacity of the network by providing additional communication pathways between source destination nodes and can potentially reduce delays in the ground network by reducing congestion in the ground network. The additional pathways through the APs also adds to the robustness and survivability of the ground network. In addition, a network of aerial platforms can also be used to reliably connect high priority nodes.

We first look at the problem of providing (basic) connectivity between disconnected ground clusters and satisfying required traffic capacity between these clusters by placing a number of APs at appropriate places to act as relay nodes. Since Aerial Platforms are scarce and expensive resources, the goal is to find the minimum number of APs and their locations so that the resultant network (both between ground nodes and APs and between the APs) is connected and there are enough pathways to support the required inter-cluster capacity (see Figure 4.1). In [35], the authors use a deterministic annealing (DA) clustering approach ([36]) to find near-optimal solutions to the problem of finding the minimum number of APs and their location so that at least one node from each ground cluster is connected to at least one AP. The minimum requirement for connecting each cluster to an AP is to have at least one node in each cluster communicating with at least one AP. In [35], the authors do not consider any communication distance constraints for AP-AP communications and do not consider any capacity constraints. We extend the approach of [35] in two ways: *a*.) include communication distance constraints between the APs so that not only are the clusters connected to the APs but the APs also form a connected network; *b*.) the APs connected to each cluster are capable of supporting the required capacity out of each cluster to other clusters with maximum AP-cluster link utilization. We then extend the basic connectivity solution in order to make the resultant network single AP survivable. Finally we propose a heuristic algorithm to connect high-priority clusters and a maximum number of non-priority clusters when the number of available APs is less than the minimum required to connect all the ground clusters.

The basic requirement for connectivity between the ground clusters and the APs is converted into a summation form distortion cost (just as in [35]). We include the communication distance constraints between the APs by adding a summation form constraint to the basic distortion function so that the addition of APs always results in a network of APs that is connected. In order to make sure that the required capacity out of each cluster to other clusters is supported by the APs connected to this cluster, we add another cost function that only depends on the assignment probabilities of the APs and results in

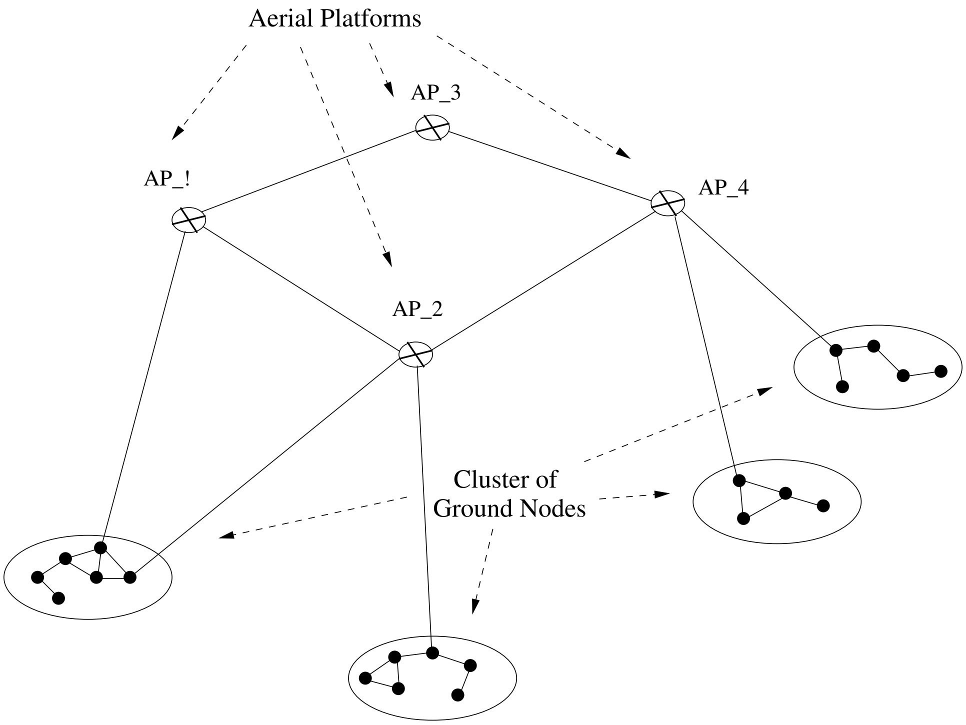

Figure 4.1: Network with four partitions and four connecting APs.

a load-balanced solution, i.e., the end resultant assignment probabilities of the APs are equal. By assigning the prior probabilities of each cluster to be proportional to the capacity required, the fact that in the end the assignment probabilities of the APs are almost equal translates to the fact that the capacities supported through each AP are nearly equal and hence we have maximum AP-cluster link utilization. In order to make the network single AP survivable, we extend the basic connectivity solution by adding another summation form constraint so that the connected AP network becomes a biconnected network and also by making sure that each ground cluster is connected to two APs by modifying the resultant optimal association probabilities of a cluster to various APs.

The basic AP location problem of connecting APs to ground clusters (without considering AP-AP connectivity but with constant AP altitude and constant AP-ground node communication range) is related to the *Euclidean disk-cover* problem where one tries to find the minimum number of circles with a given radius *r* to cover *n* given points on a plane. In fact the basic AP location problem is an extension of the *Euclidean disk-cover* problem if one considers trying to connect APs to any node in a ground cluster. Hence our placement problem is NP-hard using the NP-hardness results of the *disk-cover* problem ([37]). In order to avoid the computational complexity and difficulties of the *minimax* problems, we follow a clustering approach for finding near-optimal solutions to our problem. The clustering approach not only avoids the computational complexity of previous approaches but also provides a flexible framework to address other extensions (like AP-AP connectivity, satisfying capacity constraints, and making the resultant network single AP survivable) to the basic AP placement problem. Also in [35], the authors look at multiple nodes within a cluster connecting to a single AP (thereby providing robustness to node mobility and ground conditions and increasing capacity by reducing congestion and routing overhead) and also multiple APs connecting each cluster (thereby providing redundancy, increasing reliability and improving capacity). Since the particular form of our formulation results in a non-convex optimization problem we use the DA algorithm to avoid local minima and obtain near-optimal solutions.

This chapter is organized as follows. Section 4.2 describes our scenario and the assumption made. Section 4.3 explains our formulation of the basic connectivity problem and the inter-cluster capacity problem in the framework of a constrained clustering problem with complexity costs. Section 4.4 explains the Deterministic Annealing solution to the basic connectivity and inter-cluster problems and gives a brief review of the algorithm used. Section 4.5 extends the DA solution to make the network single AP survivable, describes an algorithm to connect high priority clusters when the number of available APs is less than the minimum number required as calculated by the DA algorithm, and describes how to call the algorithm in dynamic scenarios. Section 4.6 presents the results of the DA algorithm for basic connectivity, inter-cluster capacity, and single AP survivable network. We also compare our connectivity results with an exhaustive grid search algorithm.

4.2 Scenario and Assumptions

Let the ground nodes and the APs have identical omnidirectional radios with free space communication (where the signal decays as $1/R^2$, with R being the distance between radios) possible if the distance between two radios is less than R_2. Since the ground nodes communicate with one another in an environment (indirect reflections, etc.) where the signal decays as $1/R^\alpha$, where α is greater than 2 (suburban decay is as $1/R^4$), we assume that the ground nodes can communicate with each other if their distance is less that R_0 (with $R_0 < R_2$). Assume that the ground network has N nodes (with positions $G = \{g_i, i = 1, \ldots, N\}$) forming M clusters where the nodes within each cluster can communicate with each other and the nodes in different clusters cannot communicate with one another. Each cluster is represented by K_j, $j = 1$ to M. Also assume that all of the ground nodes, g_i ($i = 1, \ldots, N$), have the same altitude (of 0). This assumption basically keeps the problem in $\mathcal{R}^2$ and is a reasonable approximation for most practical cases.

Let each AP fly at a maximum cruising altitude of h in a holding pattern above the scenario. Since the AP-AP and AP-ground node communication can be modeled as that of free space, it is assumed that the AP-AP or AP-ground node communication can take place if the distance between the nodes is less than R_2. Since all APs fly at a constant altitude h, the connectivity problem can be reduced to $\mathcal{R}^2$, with the positions of the APs projected onto the ground and denoted by a_k (with $A = \{a_k, k = 1, \ldots, L\}$ assuming L APs). This results in a maximum AP-ground node communication distance of $R_1 = \sqrt{R_2^2 - h^2}$ with the AP-AP maximum communication distance being R_2.

Assume that the maximum link capacity between AP-AP and AP-ground node is C_{max}. Let the total capacity required from source cluster K_i to destination cluster K_j be C_{ij} with C_{ii} taken to be 0. Hence the total capacity (C_i) of the links going out and coming into cluster K_i is $C_i = \sum_{j=1}^{K}(C_{ij} + C_{ji})$. We need to have $C_i \leq C_{max}(\forall i = 1, \ldots, M)$ as the maximum AP-ground node capacity is C_{max}.

4.3 Formulation of Basic Connectivity and Inter-Cluster Capacity Problem

If the baseline ground scenario is disconnected, Aerial Platforms can be used to establish connectivity and provide required capacity. We formulate the basic connectivity problem as a constrained clustering problem ([38], [39]) with a summation form distortion function ($D(K, A)$) involving the distances between the ground clusters (K) and the APs (A) and a summation form cost function ($C_1(A)$) involving only the distances between the APs (A). The capacity constraints, including maximizing the AP-cluster link utilization, are handled by adding a complexity cost function $C_2(p(A))$ ([40]) that only depends on the *assignment probabilities* $p(a_i)$ of the APs; and relating the prior probabilities $p(K_i)$ of each cluster K_i to be proportional to C_i (the capacity required by this cluster to communicate with all other clusters). The resultant clustering problem is then solved using Deterministic Annealing (DA) to obtain near-optimal solutions. We first give a brief description of basic Deterministic Annealing in order to clarify the concepts of prior and assignment probabilities, then formulate the basic connectivity problem, and finally add the capacity constraints.

4.3.1 Deterministic Annealing

Deterministic Annealing ([36]) is a method for clustering where a large number of data points, denoted by x's, (in our problem, the various ground clusters) need to be assigned to a small number of centers, denoted by y's, (in our problem, the various APs) such that the average distortion function is minimized. The average distortion can be written as $D = \sum_x p(x)d(x, y(x))$, where $p(x)$ is the prior probability of data point x. The DA approach tries to avoid local minima by turning the hard clustering problem (where a data point is associated with only one center) into a soft/fuzzy clustering problem (where each data point can be associated to many centers via its *association probabilities* $p(y|x)$) and then minimizing the distortion at various levels of randomness measured by the Shannon entropy $H(X, Y)$. Hence the original distortion function is re-written as $D = \sum_x p(x) \sum_y p(y|x)d(x, y)$ where the *assignment probability* $p(y) = \sum_x p(x)p(y|x)$, measures the percentage of data points assigned to a center y. The objective function that DA minimizes is $F = D - TH$ or $F = D - TH(Y|X)$ at various values of temperature T starting from high temperature and then slowly decreasing the temperature.

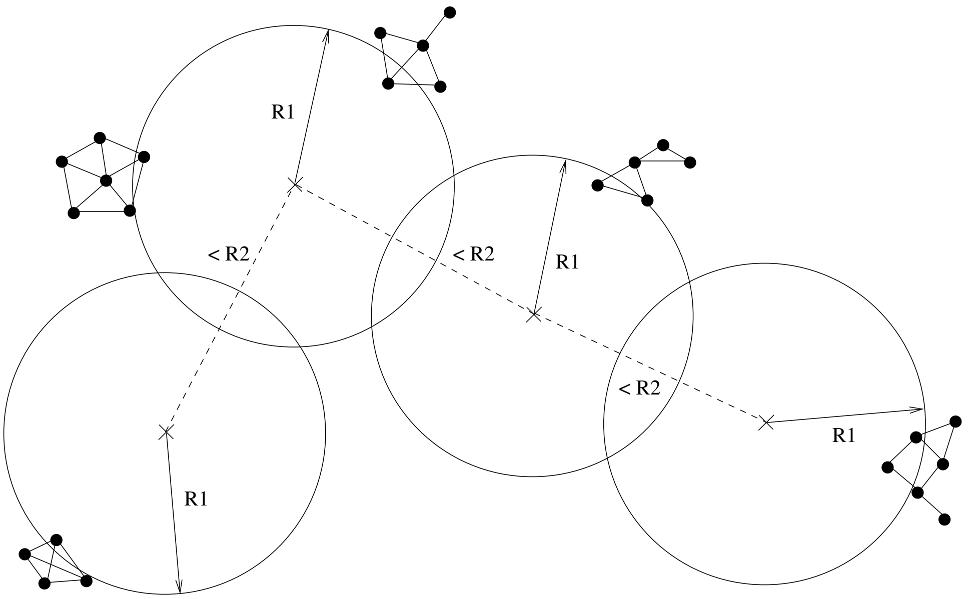

Figure 4.2: Aerial Platform Placement.

4.3.2 Basic Connectivity Problem

In order to connect the various ground clusters to the APs while ensuring that the APs form a connected network, we need to find the minimum number of APs L and their positions on the ground, a_k, (with $A = \{a_k, k = 1, \ldots, L\}$) such that:

- At least one node from each cluster is within a radius of R_1 from an AP (i.e., within a circle of radius R_1 from any AP position a_k, $k = 1, \ldots, L$; see Figure 4.2); and
- The AP locations a_k ($k = 1, \ldots, L$) are within a distance R_2 from each other (i.e., the APs form a connected graph; again see Figure 4.2).

Assuming that the APs are numbered from 1 to L, we can make sure that the APs form a connected network by ensuring that any AP numbered j is connected to atleast one lower numbered AP i, where $i < j$. This is used in the DA solution where when we add a new AP, we make sure that it is connected to at least one of the previously added APs. Hence the connectivity problem can be stated as:

$$Minimize\ L$$

$$subject\ \ to$$

$$\exists a_1, \ldots, a_L; \quad \max_{j \in \{1,\ldots,M\}} \min_{\substack{g \in K_j \\ i \in \{1,\ldots,L\}}} \parallel g - a_i \parallel \leq \ R_1$$

$$\text{and,} \qquad \max_{l \in 2,\ldots,L} \min_{m<l} \parallel a_l - a_m \parallel \leq \ R_2$$

where $\parallel \ g - a \ \parallel$ is the l^2-norm between points g and a on the ground. Finding the exact solution to the problem above involves an exhaustive search on the different ways in which nodes can be selected from each cluster and the ways clusters can be grouped together for coverage by a single AP all the while making sure that the APs are connected to each other. This problem is NP-hard as it is a generalization of the Euclidean *disk-cover* problem. Hence using the approximation,

$$\max(s_1, \ldots, s_n) \cong (s_1^\alpha + \ldots + s_n^\alpha)^{\frac{1}{\alpha}} \ \text{ for large } \alpha \tag{4.1}$$

we can convert the constraint between the ground nodes and the APs as well as the constraint between the APs into a summation form,

$$Minimize\ L$$

$$subject\ \ to$$

$$\begin{aligned} \exists a_1, \ldots, a_L; \quad \sum_{j=1}^{M} d_1(K_j, a_{u_1(j)}) \quad &\leq \quad R_1^\alpha \\ \text{and,} \qquad \sum_{l=2}^{L} d_2(a_l, a_{u_2(l)}) \quad &\leq \quad R_2^\beta \end{aligned}$$

for large α and β, where,

$$d_1(K_j, a_i) \quad = \quad \min_{g \in K_j} \parallel g - a_i \parallel^\alpha$$

$$
\begin{aligned}
d_2(a_l, a_m) &= \min_{m<l} \parallel a_l - a_m \parallel^{\beta} \\
u_1(j) &: \{1, \ldots, M\} \to \{1, \ldots, L\} \\
&\quad \text{is the function that assigns an AP to every cluster.} \\
u_2(l) &: \{2, \ldots, L\} \to \{1, \ldots, L-1\} \\
&\quad \text{is the function that assigns the closest lower numbered AP to an AP.}
\end{aligned}
$$

Constrained clustering problems of the above form are non-convex optimization problems except in special cases. Hence the Deterministic Annealing (DA) method is used to solve the constrained clustering problem for globally near-optimal solutions. Within the framework of constrained clustering ([38], [39]), the distortion function between the ground nodes and the APs is given by $D(K, A) = \sum_{j=1}^{M} d_1(K_j, a_{u_1(j)})$ and the cost function among the APs is given by $C_1(A) = \sum_{l=2}^{L} d_2(a_l, a_{u_2(l)})$.

4.3.3 Capacity Constraints

In order to ensure that the capacity required by a cluster K_i to communicate with other clusters (i.e., C_i, $\forall i = 1, \ldots, M$) is satisfied by the APs within communication range of the cluster, we need to ensure that the capacity supported by an AP, ($C_{ap}(j) \triangleq \sum_{i=1}^{M} C_i \, I(\text{AP } j \text{ is associated with cluster } i), \forall\, j = 1, \ldots, L$), is less than the maximum link capacity C_{max}. Since in the clustering formulation, we can have a single cluster K_i associated with different APs via its association probabilities $p(a_j|K_i)$, we rewrite the capacity supported by an AP, $C_{ap}(j)$ as

$$C_{ap}(j) = \sum_{i=1}^{M} C_i \, p(a_j|K_i) \leq C_{max}$$

Denoting the AP-cluster link utilization as $u(j) = C_{ap}(j)/C_{max}$, in order to maximize the sum of the AP-cluster link utilizations, we would like to $max \sum_{j=1}^{L} u(j)$.

In order to satisfy the capacity constraints from each cluster K_i to all other clusters, we let the cluster prior probability be set to $p(K_i) = C_i / \sum_{j=1}^{M} C_j$ and add a complexity cost function $C_2(p(a_k)) = 1/p(a_k)^s$ that only depends on the assignment probabilities of the APs. For high values of s, the cost value for small $p(a_k)$ (i.e., $1/p(a_k)^s$) blows up and the end resultant solution ([40]) tends to be load balanced, i.e., $p(a_k) = 1/L, \forall k = 1, \ldots, L$. But

$$\begin{aligned} p(a_k) &= \sum_{i=1}^{M} p(K_i)\, p(a_k|K_i) \\ &= \sum_{i=1}^{M} \left(C_i / \sum_{j=1}^{M} C_j \right) p(a_k|K_i) \\ \Rightarrow\ p(a_k) \sum_{j=1}^{M} C_j &= \sum_{i=1}^{M} C_i\, p(a_k|K_i) \ = \ C_{ap}(k) \end{aligned}$$

Hence we stop adding APs when the maximum of $p(a_k) \sum_{j=1}^{M} C_j$ over all the APs becomes less than the maximum link capacity C_{max}. Since all the $p(a_k)$'s are approximately equal, we also tend to maximize the sum of the AP-cluster link utilizations.

4.4 Deterministic Annealing Solution

The overall distortion function D including the AP-AP connectivity constraints and the cluster capacity constraints is given by:

$$D = \sum_{i=1}^{M} p(K_i) \sum_{j=1}^{L} p(a_j|K_i) \left[d_1(K_i, a_j) + \eta\, C_2(p(a_j)) \right] + \ \lambda \sum_{l=2}^{L} d_2(a_l, a_{u_2(l)})$$

The deterministic annealing algorithm tries to minimize the objective function $F = D - T\,H(A|K)$ where

$$H(A|K) = -\sum_{i=1}^{M} p(K_i) \sum_{j=1}^{L} p(a_j|K_i) \log p(a_j|K_i).$$

Minimizing F with respect to the association probabilities $p(a_j|K_i)$ with the additional constraints that $p(a_j) = \sum_{i=1}^{M} p(K_i) p(a_j|K_i)$ and $\sum_{j=1}^{L} p(a_j|K_i) = 1$ gives the Gibbs distribution:

$$p(a_j|K_i) \quad = \quad \frac{\exp\left(-\frac{d_1(K_i,a_j)+\eta C_2(p(a_j))+\eta p(a_j)\frac{dC_2(p(a_j))}{dp(a_j)}}{T}\right)}{Z_{K_i}}$$

where

$$Z_{K_i} \quad = \quad \sum_{j=1}^{L} \exp\left(-\frac{1}{T}(d_1(K_i, a_j) + \eta C_2(p(a_j)) + \eta p(a_j)\frac{dC_2(p(a_j))}{dp(a_j)})\right)$$

The corresponding minimum F^* of F is obtained by plugging the value for the association probabilities $p(a_j|K_i)$ into $F = D - T\,H(A|K)$ to obtain:

$$F^* \quad = \quad -T\sum_{i=1}^{M} p(K_i) \log Z_{K_i} - \eta \sum_{j=1}^{L} p^2(a_j) \frac{dC_2(p(a_j))}{dp(a_j)} + \lambda \sum_{l=2}^{L} d_2(a_l, a_{u_2(l)})$$

The optimal AP locations a_k are given by minimizing F^* leading to the following expression involving the gradients of a_k that needs to be set to zero:

$$\sum_{j=1}^{M} p(K_j, a_k)\, \nabla_{a_k} \left(d_1(K_j, a_k)\right) + \lambda\, \nabla_{a_k} \left(\sum_{l=2}^{L} d_2(a_l, a_{u_2(l)})\right)$$

This leads to two equations, one for the x coordinate of a_k (i.e., x_{a_k}) and another for the y coordinate of a_k (i.e., y_{a_k}):

$$\begin{aligned} x_{a_k} &= \frac{\alpha \sum_{j=1}^{M} d1X_{numr}(K_j, a_k) + \lambda\,\beta\,(d2X_{numr}(a_k))}{\alpha \sum_{j=1}^{M} d1_{denr}(K_j, a_k) + \lambda\,\beta\,(d2_{denr}(a_k))} \\ y_{a_k} &= \frac{\alpha \sum_{j=1}^{M} d1Y_{numr}(K_j, a_k) + \lambda\,\beta\,(d2Y_{numr}(a_k))}{\alpha \sum_{j=1}^{M} d1_{denr}(K_j, a_k) + \lambda\,\beta\,(d2_{denr}(a_k))} \end{aligned}$$

where

$$\begin{aligned}
d1X_{numr}(K_j, a_k) &= x_{K_j}\, p(K_j)p(a_k|K_j)d_1(K_j, a_k)^{1-2/\alpha} \\
d1Y_{numr}(K_j, a_k) &= y_{K_j}\, p(K_j)p(a_k|K_j)d_1(K_j, a_k)^{1-2/\alpha} \\
d1_{denr}(K_j, a_k) &= p(K_j)p(a_k|K_j)\, d_1(K_j, a_k)^{1-2/\alpha} \\
d2X_{numr}(a_k) &= x_{a_{u_2(k)}}\ d_2(a_k, a_{u_2(k)})^{1-2/\beta} + \sum_{l>k} I(u_2(l) = k)\ x_{a_l}\ d_2(a_l, a_k)^{1-2/\beta} \\
d2Y_{numr}(a_k) &= y_{a_{u_2(k)}}\ d_2(a_k, a_{u_2(k)})^{1-2/\beta} + \sum_{l>k} I(u_2(l) = k)\ y_{a_l}\ d_2(a_l, a_k)^{1-2/\beta} \\
d2_{denr}(a_k) &= d_2(a_k, a_{u_2(k)})^{1-2/\beta} + \sum_{l>k} I(u_2(l) = k)\ d_2(a_l, a_k)^{1-2/\beta}
\end{aligned}$$

For $k = 1$, $d_2(a_k, a_{u_2(k)}) = 0$, so that the first term in $d2X_{numr}(a_k)$, $d2Y_{numr}(a_k)$, and $d2_{denr}(a_k)$ is not present.

4.4.1 Algorithm

We start with an initial temperature $T = T_{init}$ and $\lambda = 0$ to get the unconstrained clustering solution for that T (i.e., taking into account only the connectivity between the APs and ground clusters). At a given T, we then gradually increase λ and optimize until the maximum of the minimum inter-node distance between an AP and its lower numbered APs is just less than R_2. We then reduce the temperature T and repeat the procedure of increasing λ from 0. The temperature T is progressively reduced until all the clusters are covered by at least one AP and the capacity constraints are satisfied, i.e., $\max_{k=1}^{L} p(a_k) \sum_{j=1}^{M} C_j \leq C_{max}$.

At each iteration (i.e., fixed T and fixed λ), the association probabilities $p(a_i|K_j)$ are first calculated, then the assignment probabilities $p(a_i)$ are calculated, and finally the optimal AP locations a_i are determined until there is convergence. If after a fixed number of temperature reduction iterations, either all the clusters are not covered or the cluster capacity constraints are not satisfied, then the number of APs is increased. This is done by choosing the AP center i with either farthest associated groups or maximum $p(a_i)$ and adding a small perturbation to its current location, and then dividing its probability $p(a_i)$ equally between the new and old center.

$$p(a_i) = p(a_i)/2; \quad p(a_{L+1}) = p(a_i); \quad L = L + 1$$

If a new center is really needed, then the two centers move apart from each other, else they merge again after a few steps. This is checked by finding the distance between the new and old centers after a couple of temperature reduction iterations and merging them if the distance is less than a threshold.

4.5 Extensions: AP Survivability, Ground Cluster Priority, Dynamic Scenarios

4.5.1 Single AP Survivable Network

APs flying above the ground can fail (e.g., crashing, being shot down) and hence we would like to have a network of APs such that even if one of the APs fail, the resultant network of APs and ground clusters is still connected. A connected network is called a single node survivable network or a biconnected network or a 2-connected network if given that a node and all links to it are removed, a path still exists between any two remaining nodes, i.e., the network is still connected. Our goal is to design a single AP survivable network, i.e., design a network of connected APs that connect all the ground clusters together such that the network is still connected even if any single AP is removed from the network.

Our basic connectivity formulation made sure that the resultant network of APs and ground clusters was connected by dividing the problem into two parts: 1) make sure that each ground cluster is connected to atleast one AP; and 2) make sure that the APs form a connected network. We can extend this basic connectivity formulation to make the network single AP survivable by the following definition of a single AP survivable network. A connected network of APs and ground clusters becomes single AP survivable if the following two conditions hold:

1. Each ground cluster is connected to atleast two APs.

2. The connected AP network forms a biconnected network.

Condition 1 is necessary since for a particular ground cluster to maintain connectivity with the AP-AP network despite a single AP failure, it needs to be connected to atleast two APs. Condition 2 is necessary since if a single AP fails, we still want the network of APs to be connected to each other. Thus the two conditions together guarantee that the network of APs and ground clusters remains connected despite the failure of any single AP. one of the AP fails. The following two sections describe the methods used satisfy condition 1 and 2 respectively so as to make the network single AP survivable. The final section explains the changes necessary to the DA algorithm.

4.5.1.1 Ground Cluster Connected to Two APs

In order to connect a ground cluster to two APs, we extend the DA algorithm as in [35] by modifying the optimal association probabilities $p(a_i|K_j)$ calculated during each iteration of the DA algorithm (see section 4.4). The association probability $p(a_i|K_j)$ indicates the influence of cluster K_j in determining the AP position a_i. In the DA algorithm, the association probabilities of a cluster start from a uniform distribution at high temperatures (where a cluster equally influences every AP) and converge at low temperatures to a vector with a one and all zeros (hard clustering), where, each cluster affects only one AP. Hence in order for a cluster to be connected to L_i (here $L_i = 2$) APs, at low temperatures, the association probabilities of this cluster to the L_i APs should be large enough to influence the location of the L_i APs. In order to achieve this goal, at each iteration of the DA algorithm, the calculated association probabilities $\mathbf{p}(A|K_i)$ of a cluster K_i to the different AP centers ($A = [a_1, \ldots, a_L]$) are adjusted so that the highest L_i probabilities are made equal. This is done by ordering the values of the association probability vector $\mathbf{p}(A|K_i)$ from largest to smallest probability values ($\mathbf{p}(A|K_i) = [p_1, p_2, \ldots, p_L]$) and then adjusting the largest $L_i = 2$ probabilities to be the average of the first L_i different probabilities, i.e.,

$$\mathbf{p}(A|K_i) = \left[\frac{1}{L_i} \sum_{j=1}^{L_i} p_j, \ldots, \frac{1}{L_i} \sum_{j=1}^{L_i} p_j,\, p_{L_i+1}, \ldots, p_L \right]$$

4.5.1.2 Biconnected AP Network

In order to ensure that the network of connected APs form a biconnected network, we extend the connectivity formulation of section 4.3.2 (where each AP is constrained to connect to the nearest previously added AP) by adding additional AP-AP distance constraints that make sure that each AP is additionally connected to the second nearest previously added AP. The additional AP-AP distance constraint for an AP l (where $l > 2$) is of the form:

$$\min_{m<l,\, m\neq u_2(l)} \| a_l - a_m \| \leq R_2 \qquad \forall l > 2$$

where $u_2(l)$ as defined previously is the function that assigns the closest lower numbered AP to an AP l. Thus to ensure that the resultant AP network is biconnected, we make sure that as each AP is added, it is connected to two previously added APs not just one. A proof that the resultant AP network is biconnected is as follows. Suppose an AP l is removed and suppose that it not be the first AP to be added in the DA solution, i.e., $l > 1$. All APs m, where $m < l$, are still connected to each other and form a connected subnetwork since each AP is connected to atleast one previously added AP. If l is the last AP to be added, we have shown that the remaining AP network is still connected or that the original AP network is biconnected. If l is not the last AP to be added, consider AP $l + 1$. It is connected to atleast one of the APs m where $m < l$, since each AP is connected to atleast two previously added APs. Hence the set of APs $[1, \ldots, l-1, l+1]$ form a connected network. Similarly AP $l+2$ will be connected to atleast one of the APs in the set comprising remaining APs numbered lower than itself, i.e., $[1, \ldots, l-1, l+1]$ and hence the set $[1, \ldots, l-1, l+1, l+2]$ form a connected network. We can similarly argue for all APs numbered greater than l. Now if AP to be removed is the first one added in the DA solution, i.e., $l = 1$, then the network of remaining APs is still connected as each higher numbered AP is connected to two lower numbered APs (except for AP numbered 2). So AP numbered 3 is connected to AP numbered 2. AP 4 is connected to one of APs 2 or 3 and so on. Hence we have proved that if each AP is connected to atleast two previously added APs, the resultant AP network is biconnected.

In order to make sure that the AP network is biconnected, we want the following to hold true:

$$\begin{aligned} \max_{l \in 2,\ldots,L} \quad \min_{m<l} \quad \| a_l - a_m \| &\leq R_2; \quad \text{and} \\ \max_{l \in 3,\ldots,L} \quad \min_{m<l,\, m \neq u_2(l)} \quad \| a_l - a_m \| &\leq R_2 \end{aligned}$$

Using the approximation for the maximum as in equation 4.1, we convert the connectivity constraints into a summation form as given below:

$$Minimize\ L$$

$$subject\ \ to,\ \exists a_1, \ldots, a_L;$$

$$\begin{aligned} \sum_{j=1}^{M} d_1(K_j, a_{u_1(j)}) &\leq R_1^\alpha \\ \text{and,} \quad \sum_{l=2}^{L} d_2(a_l, a_{u_2(l)}) + \sum_{l=3}^{L} d_3(a_l, a_{u_3(l)}) &\leq R_2^\beta \end{aligned}$$

for large α and β, where the functions $d_1(K_j, a_i)$, $d_2(a_l, a_m)$, $u_1(j)$, and $u_2(l)$ are as defined in section 4.3.2 and,

$$\begin{aligned} d_3(a_l, a_m) &= \min_{m<l,\, m \neq u_2(l)} \| a_l - a_m \|^\beta \\ u_3(l) &: \{3, \ldots, L\} \rightarrow \{1, \ldots, L\} - \{u_2(l)\} \end{aligned}$$

is the function that assigns the second nearest lower numbered AP (which will not be equal to $u_2(l)$) to an AP.

Starting from the third AP added and beyond, we make sure that it is connected to two previously added APs via constraints $d_2(a_l, a_{u_2(l)})$ and $d_3(a_l, a_{u_3(l)})$.

4.5.1.3 Changes to DA Solution

The overall distortion function D for the AP to be connected to two lower numbered neighboring APs (with the cluster capacity constraints) is modified from that in section 4.4 and is given by:

$$\begin{aligned} D &= \sum_{i=1}^{M} p(K_i) \sum_{j=1}^{L} p(a_j|K_i) \left[d_1(K_i, a_j) + \eta\, C_2(p(a_j)) \right] \\ &\quad + \lambda \left[\sum_{l=2}^{L} d_2(a_l, a_{u_2(l)}) \;+\; \sum_{l=3}^{L} d_3(a_l, a_{u_3(l)}) \right] \end{aligned}$$

The equation for the optimal association probabilities remains the same as in section 4.4 but the equation for the optimal AP locations a_k changes to include the additional AP-AP distance constraints $d_3(a_l, a_{u_3(l)})$ leading to the following expression for the gradients of a_k that need to be set to zero.

$$\begin{aligned} \sum_{j=1}^{M} p(K_j, a_k)\, \nabla_{a_k} \left(d_1(K_j, a_k) \right) + \lambda\, \nabla_{a_k} \left(\sum_{l=2}^{L} d_2(a_l, a_{u_2(l)}) \right) \\ + \lambda\, \nabla_{a_k} \left(\sum_{l=3}^{L} d_3(a_l, a_{u_3(l)}) \right) \end{aligned}$$

This leads to an extra term $\lambda\beta(d3X_{numr}(a_k))$ in the numerator and an extra term $\lambda\beta(d3_{denr}(a_k))$ in the denominator of the solution for the x coordinate of a_k (x_{a_k}, see section 4.4). Similarly y_{a_k} has an extra term $\lambda\beta(d3Y_{numr}(a_k))$ in the numerator and an extra term $\lambda\beta(d3_{denr}(a_k))$ in the denominator. The values for the extra terms are as below:

$$\begin{aligned} d3X_{numr}(a_k) &= x_{a_{u_3(k)}}\, d_3(a_k, a_{u_3(k)})^{1-\frac{2}{\beta}} + \sum_{l \neq k} I(u_3(l) = k)\; x_{a_l}\; d_3(a_l, a_k)^{1-\frac{2}{\beta}} \\ d3Y_{numr}(a_k) &= y_{a_{u_3(k)}}\, d_3(a_k, a_{u_3(k)})^{1-\frac{2}{\beta}} + \sum_{l \neq k} I(u_3(l) = k)\; y_{a_l}\; d_3(a_l, a_k)^{1-\frac{2}{\beta}} \\ d3_{denr}(a_k) &= d_3(a_k, a_{u_3(k)})^{1-\frac{2}{\beta}} + \sum_{l \neq k} I(u_3(l) = k)\; d_3(a_l, a_k)^{1-\frac{2}{\beta}} \end{aligned}$$

The DA algorithm proceeds as in section 4.4.1 except for two changes. At a given T, the value of λ is increased till the distance from each AP to atleast two lower numbered APs is less than R_2. Also at each iteration (i.e., for fixed T and fixed λ), the association probabilities $p(a_i|K_j)$ are first calculated according to the formula given in section 4.4 and are then changed as per section 4.5.1.1 so that a ground cluster is connected to two APs.

4.5.2 Cluster Priority and Insufficient APs

In this section, we describe a heuristic algorithm for the scenario when certain ground clusters of high priority need to be connected together and the number of APs available (L_a) is less than the minimum number of APs (L) required to connect all the ground clusters. Certain ground clusters can contain nodes of high value and these ground clusters have to be given preference when connecting various clusters together using APs. In case the number of APs available, L_a, is greater than or equal to the minimum number of APs, L, required to connect all the ground clusters as given by the DA solution, then the output of the DA algorithm can be used to place the APs. But if $L_a < L$, then the following algorithm can be used to give preference to the high priority clusters and connect as many of the other clusters as possible. The algorithm basically involves running the DA algorithm with only the high priority clusters and then adding as many of the non-priority clusters as possible. The steps are as follows:

1. Run DA solution on the scenario with only high priority clusters to obtain minimum number of APs required, L_{hp}.

2. If $L_a = L_{hp}$, then add all non-priority clusters that are within range (i.e., within R_1) of the APs.

3. If $L_a < L_{hp}$, then remove the high priority cluster that is farthest from any of the APs and run DA solution. Repeat this step till $L_a = L_{hp}$.

   (a) Add all non-priority clusters that are within range (i.e., within R_1) of the APs.

4. If $L_a > L_{hp}$, then add all non-priority clusters that are within range of the APs.

   (a) Among the remaining non-priority clusters, add the closest one (to any AP) to the scenario and run DA solution till number of APs required is equal to L_a.

   (b) Add any remaining non-priority clusters that are within range of any AP.

4.5.3 Dynamic Scenarios

The ultimate goal for the AP Placement algorithm is to use the algorithm in scenarios with changing topology. In this case, we assume that the algorithm is called periodically (depending on ground node movements) to determine the new number of APs and their locations to make the resultant network connected. It is then desirable that the algorithm converges as quickly as possible to respond to topology changes. In such situations we can take advantage of the fact that the new node locations will not be drastically different from their previous values. We can start the algorithm with a lower temperature T and the previous number and locations of APs as the new starting number of APs and their locations so that the convergence time is reduced.

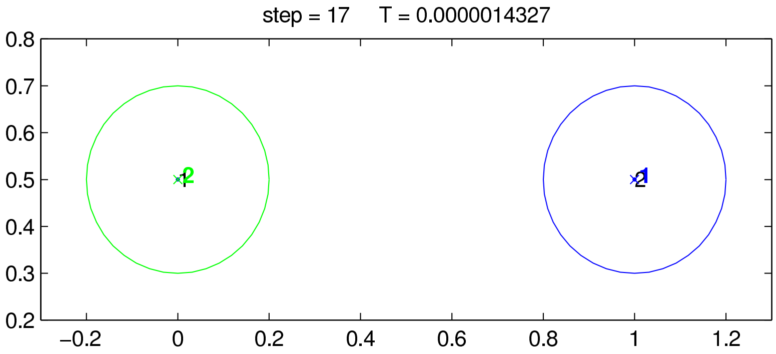

Figure 4.3: Simple 2 node scenario: AP Placement with only AP-ground node connectivity.

4.6 Results

4.6.1 Basic Connectivity Constraints

To test the basic connectivity solution, we fix the inter-ground node communication distance R_0 to 0.1 and AP-ground node communication distance R_1 to 0.2. We assume that the APs are connected if $R_2 = 2 * R_1$, i.e., two APs are connected if circles of radius R_1 drawn around each AP intersect. Using a simple scenario of 2 nodes at a distance of 1.0 from each other forming 2 clusters, we see that the minimum number of APs needed taking into account only the connectivity between the APs and ground clusters is 2 as shown in Figure 4.3 (the dots indicate ground nodes and crosses indicate APs; here the APs are directly above the ground nodes). The addition of basic AP-AP connectivity constraints necessitates a minimum of 3 APs to form a connected network as seen in Figure 4.4 with AP 2 acting as a relay between AP 1 and AP 3.

To test a bigger scenario, we used the 170 node example in [35] where the nodes form 17 clusters. R_0 and R_1 are the same as before, i.e., 0.1 and 0.2 respectively. We again

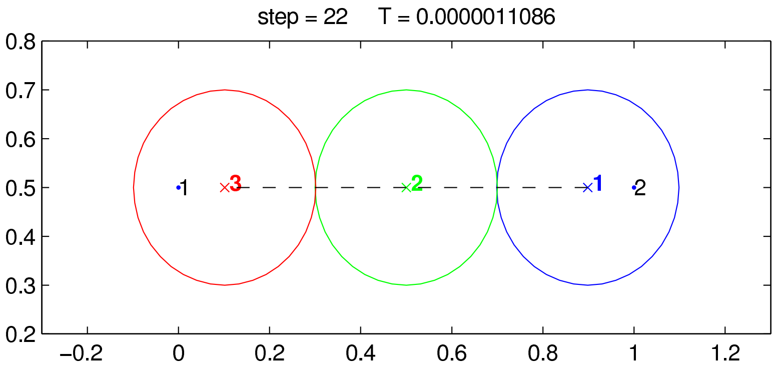

Figure 4.4: Simple 2 node scenario: AP Placement with AP-ground node connectivity and AP-AP connectivity (showing need for relay AP).

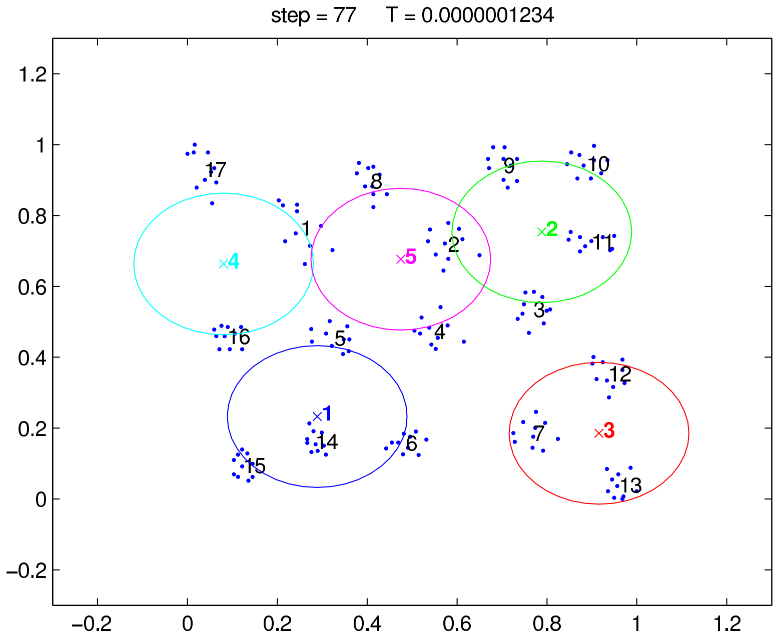

Figure 4.5: Complex Scenario: AP Placement with only AP-ground node connectivity.

choose R_2 so that $R_2 = 2 * R_1$. In [35], the authors show that the minimal number of APs to cover the various clusters without taking into consideration connectivity between the APs is 5 as shown in Figure 4.5. We see that AP 1 and AP 3 are not connected. Using the constrained clustering formulation taking into account the inter-AP connectivity, we

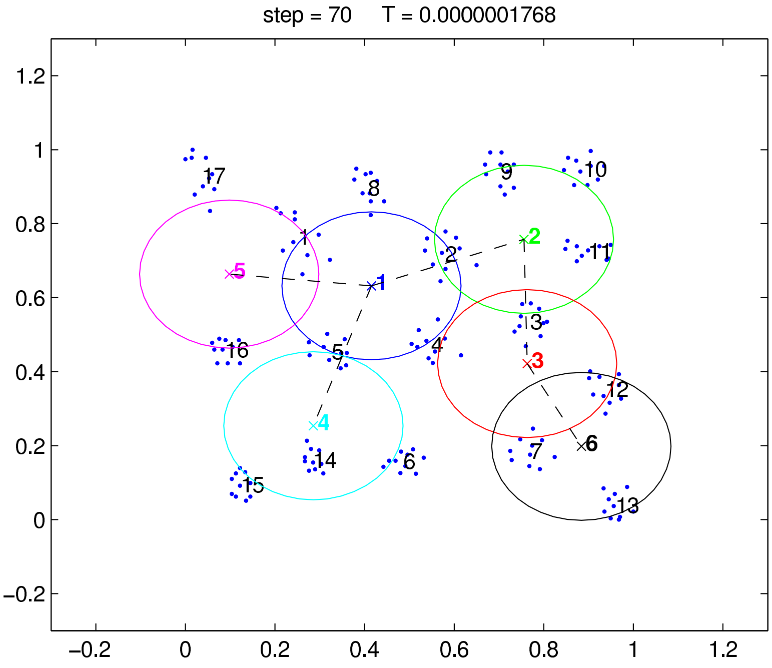

Figure 4.6: Complex Scenario: AP Placement with AP-ground node connectivity and AP-AP connectivity.

obtain the output shown in Figure 4.6. We see that 6 APs are necessary for connecting the APs with one another and ensuring that all clusters are connected to at least one AP.

The results of the constrained clustering formulation with inter AP connectivity are compared with a Grid algorithm that performs an exhaustive search over the ground node area to find the minimum number of APs required to connect the different clusters and also have connectivity among themselves. The Grid algorithm divides the area into a grid with a granularity of 0.02 and then performs an exhaustive search over all the different AP locations till it finds a solution. The algorithm starts with a single AP and then increments the number of APs until a solution is found. Obviously, this procedure is not scalable and can only be used in relatively small scenarios. The Grid algorithm when run with the

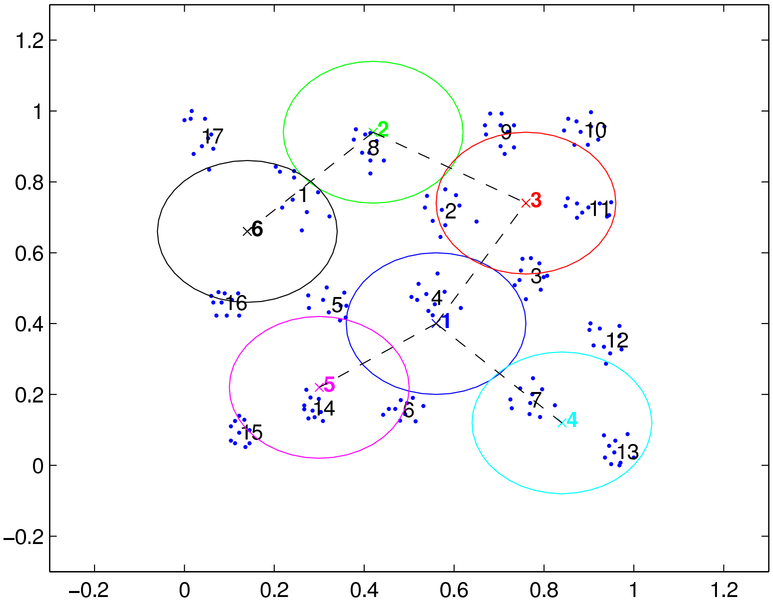

Figure 4.7: Grid Algorithm (with $R_2 = 0.4$): Number and location of APs.

same 170 node scenario also requires a minimum of 6 APs to ensure connectivity both among the ground clusters and among each other. The output of the grid algorithm with $R_2 = 0.4$ is shown in figure 4.7.

4.6.2 Capacity Constraints

To test the inclusion of capacity constraints, we use a simple scenario of 4 nodes arranged on the corners of a square with sides 0.35 forming 4 clusters (see figures 4.8 and 4.9). R_0, R_1, and R_2 are the same as in the previous section. C_1 (total capacity out of node 1 to all other nodes) and and C_2 are set to 0.4 Mbps each. The corresponding capacity for nodes 3 and 4 is set to 0.8 Mbps. C_{max} is set to 1.0 Mbps. If capacity constraints are taken into account, a single AP can support both nodes 1 and 2 while nodes 3 and 4 need a separate AP each. Thus the minimum number of APs taking into account capacity

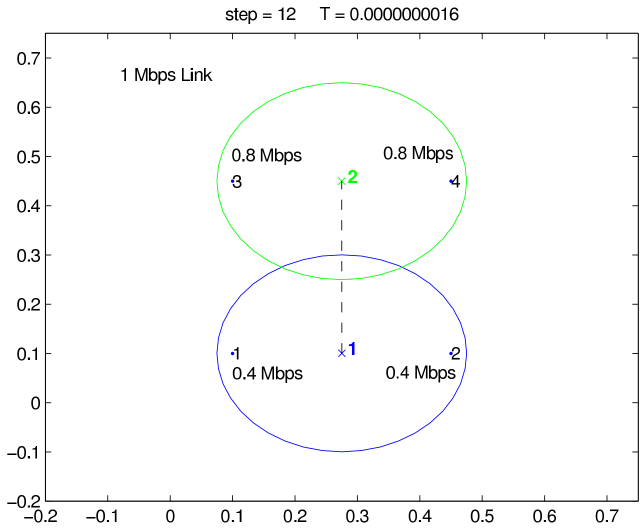

Figure 4.8: Simple 4 Node Scenario: AP Placement for connectivity without capacity constraints.

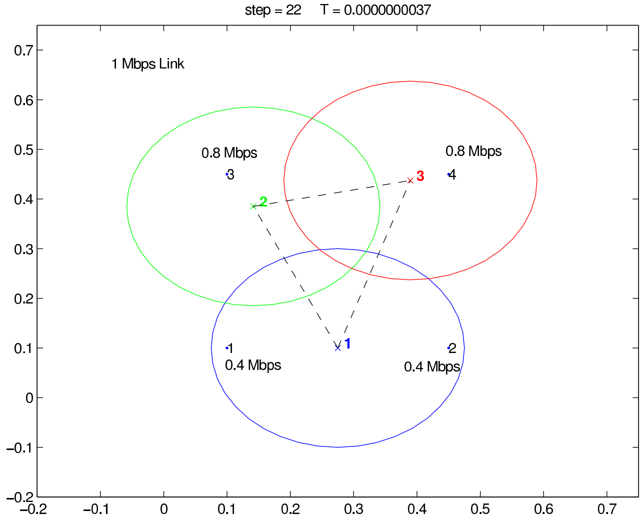

Figure 4.9: Simple 4 Node Scenario: AP Placement for connectivity with capacity constraints.

constraints is 3 and this is shown in figure 4.9. The solution without taking into account capacity constraints requires 2 APs for connectivity as seen in figure 4.8. More work

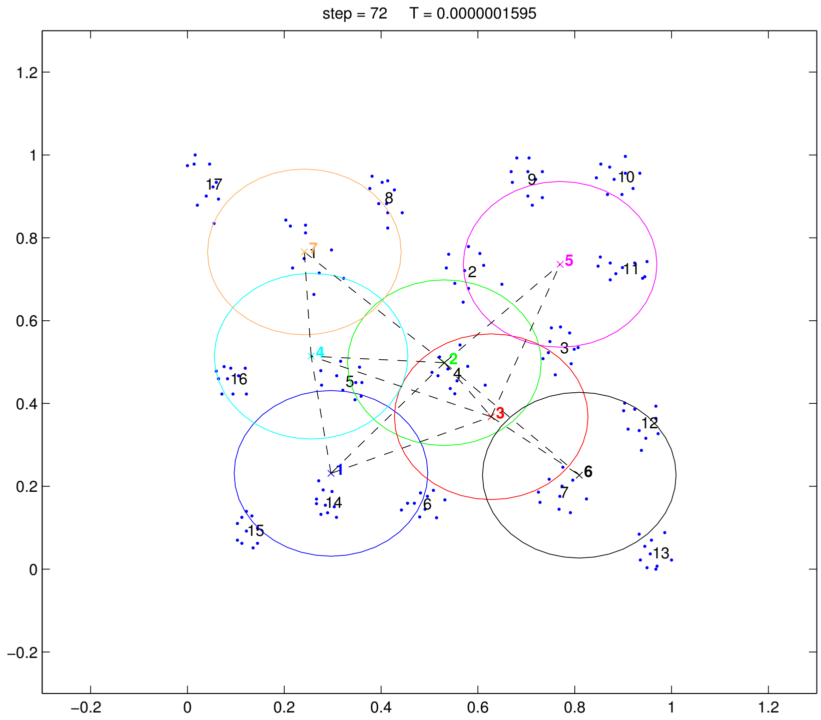

Figure 4.10: AP Placement for biconnected AP network (each AP is connected to 2 lower numbered APs): 7 APs needed.

needs to be done on the annealing schedule taking into account different η values.

4.6.3 Single AP Survivable Network

We run the same 170 node example as used in section 4.6.1 with the same values of R_0 $(= 0.1)$, R_1 $(= 0.2)$, and R_2 $(= 0.4)$. For basic connectivity between the APs, we showed in section 4.6.1 (figure 4.6) that 6 APs are necessary to connect all the ground clusters. Figure 4.10 shows the result of AP placement to make the AP network biconnected where each AP is constrained to connect to atleast two previously added APs. We

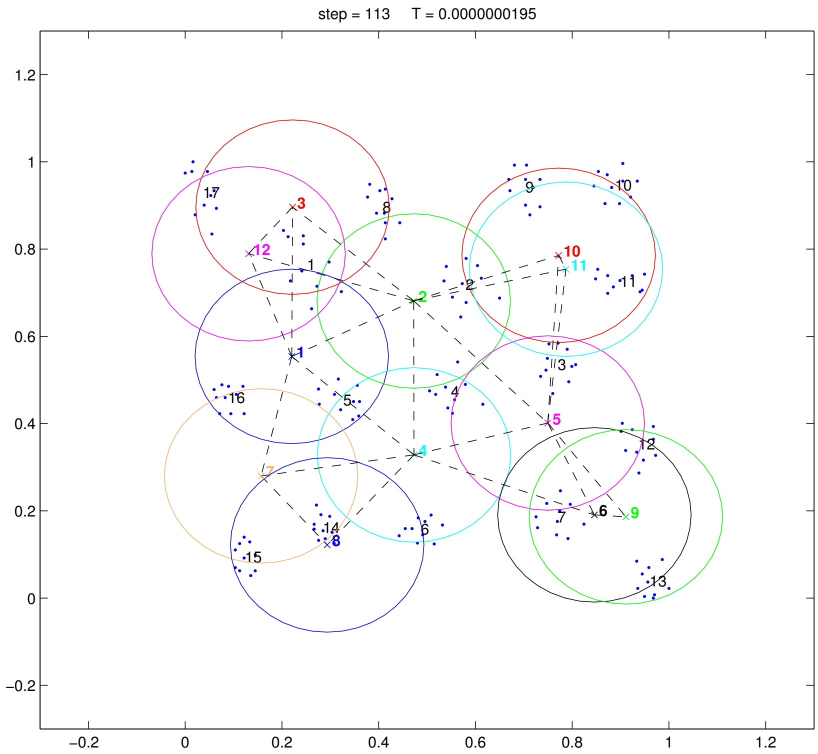

Figure 4.11: AP Placement for single AP survivable network: 12 APs needed.

see that we need one additional AP, i.e., a total of 7 APs for this enhanced connectivity amongst APs resulting in a connected AP network even if one of the APs is removed. We can now trivially make the network of APs in figure 4.10 single node survivable by placing an additional AP above each of the 7 APs. But this requires a total of 14 APs. Figure 4.11 is the result of the DA extension to make the network single node survivable. From the figure, we see that 12 APs are required (less than the trivial solution of 14 APs) to make the network single AP survivable. We see that each ground cluster is covered by atleast 2 APs and the AP network is biconnected.

Chapter 5

Conclusions and Future Work

In this thesis, we have looked at the general problem of modeling and designing a mobile ad-hoc wireless network, with emphasis on performance analysis and design of reservation based wireless networks and on making a disconnected network connected via addition of Aerial Platforms.

In particular, we provide a systematic methodology for the analysis and design of MANETs. The methodology is based on approximating the various network components like the MAC protocol and the physical layer via approximate reduced load models that link the various layers via a set of interdependent equations that is valid for the entire network. These equations are solved via fixed point iterations to obtain a set of consistent values for the performance metrics of interest like throughput, delay and packet loss. In order to design a network to satisfy performance metrics, we rely on sensitivity analysis using explicit equations or Automatic Differentiation. We apply this methodology for the 802.11 DCF MAC model of [10] for a realistic scenario obtained from a field exercise performed as part of the CBMANET project to obtain performance estimates of throughput. We extend this 802.11 DCF model to estimate end-to-end delay.

We focus on using the modeling and design methodology for a generic reservation based wireless MAC protocol called Unifying Slot Assignment Protocol (USAP). We develop reduced load loss network models extended to a wireless network to estimate blocking probability and throughput for the Hard Scheduling (Virtual Circuit) mode of USAP. We also use an implied cost formulation to come up with explicit equations to compute throughput sensitivities and use them to maximize network throughput.

We also look at the problem of topology control via dynamic addition of Aerial Platform (AP) nodes in order to connect a wireless ad-hoc network composed of disconnected clusters and also to improve the performance of such networks. We transform the connectivity problem into a summation form clustering problem with summation form constraints and use Deterministic Annealing to obtain near-optimal solutions for the minimum number and location of APs so that a disconnected ground network can become connected. We then extend the connectivity formulation so that the traffic between ground clusters is supported by the AP to ground cluster links. We also extend the basic connectivity solution so that the resultant network after the loss of a single AP is still connected.

5.1 Modeling and Design of Reservation Based MAC

We develop models to estimate throughput and blocking probability for source-destination connections using a generic reservation based virtual-circuit MAC protocol called Unifying Slot Assignment Protocol (USAP) and use these models to calculate throughput sensitivities in order to maximize total network throughput. USAP divides the communication channel into periodic frames and each frame is divided into orthogonal time-frequency cells for data and control traffic. USAP uses a generic set of rules in order to reserve end-to-end connections. For USAP Hard Scheduling as described in [4], we develop reduced load loss network models extended to a wireless network to estimate the blocking probability of each source-destination connection. The sharing of the wireless channel between a link and its neighboring links is modeled by reduced available link capacity distribution which is calculated using USAP reservation rules and the traffic among neighboring nodes. We show the validity of this approach of using a distribution on the link capacity by matching results with simulation when using the extended reduced load loss network models but with available link capacity distribution obtained from simulation. But our current available capacity estimation model does not estimate the available link capacity distribution well and a better method to estimate the distribution needs to be developed. One method could be to replace the link capacity estimation model with an estimate of the simulation's free capacity distribution via a parametric multinomial or other distribution where learning methods can be used to estimate the parameters of the distribution. The throughput sensitivities are calculated using an implied cost formulation and are used to maximize total throughput.

For the simpler case of the USAP frame with only a single frequency channel, we develop better models to estimate the link blocking probabilities for the reduced load loss network approximation by analysis of the link neighborhood directly using cliques instead of first estimating the available link capacity distribution and then estimating the link blocking probability for each value of available link capacity. The blocking probability of a call at a particular link is calculated by considering cliques of neighboring interfering links that cannot transmit simultaneously and assuming that these cliques block independently. We show that this method matches simulation results well. One can improve the link blocking probability estimation by perhaps only considering the most dissimilar cliques.

We also develop models for USAP as used in the Mobile Data Link (MDL) layer of the Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW). These models estimate throughput for both multicast and unicast traffic. The models again use reduced load loss network approximations extended for a multicast wireless network with sharing of the wireless medium between a node and its (upto 2 hop) neighbors modeled by considering cliques of interfering nodes around a particular node and assuming that these cliques block independently. We compare the results of our modeling with simulation and obtain good results using our extended reduced load loss network models.

5.2 Connecting Disconnected Ground Clusters using APs

MANETs are not always connected and in some situations there is need for these disconnected clusters to be connected to one another. We develop fast near optimal algorithms using Deterministic Annealing (DA) for placement of a minimum number of Aerial Platforms (APs) in order to connect a wireless network composed of disconnected clusters. We further develop DA algorithms to place APs so that the the connected network remains connected even if a single AP fails as well as an extension to make sure that the connected ground and AP network is such that the traffic between ground clusters is supported by the AP to ground cluster links.

In a dynamic scenario that is divided into time snapshots, the DA algorithm is run at each time snapshot to obtain the number and location of the APs at each snapshot. Since the ground nodes do not drastically change their location in consecutive snapshots, the DA algorithm is called at the latter snapshot with the new ground node locations but with previous number and locations of the APs and with a lower temperature T (than that used initially). This reduces the convergence time of the DA algorithm. But work needs to be done to find a valid trajectory for the APs between the time snapshots given constraints on the speed, angular velocity, and minimum turning radius.

When choosing the number and location of the APs, while we have considered an extension to the DA algorithm so that the connected ground-AP network is such that the traffic between ground clusters is supported by the AP to ground cluster links, we have not looked into satisfying the capacity constraints on the AP-AP links and have not looked into the routing of the inter-cluster traffic in the AP-AP network.

Bibliography

[1] T. Krout, “Real world manets,” Presentation at DARPA ITMANET Workshop, March 7 2006.

[2] A. Swami, “Mobile multi-hop military ad hoc wireless networks,” Presentation at DARPA ITMANET Workshop, March 7 2006.

[3] G. Bianchi, “Performance analysis of the ieee 802.11 distributed coordination function,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[4] C. Young, “USAP: a unifying dynamic distributed multichannel TDMA slot assignment protocol,” in *Military Communications Conference, 1996. MILCOM ’96, Conference Proceedings, IEEE*, Oct. 1996.

[5] ——, “The Mobile Data Link (MDL) of the Joint Tactical Radio System Wideband Networking Waveform,” in *Military Communications Conference, 2006. MILCOM ’06, Conference Proceedings, IEEE*, 23-25 Oct. 2006.

[6] A. Kumar, E. Altman, D.Miorandi, and M. Goyal, “New insights from a fixed point analysis of single-cell ieee 802.11 wlans,” in *Proceedings of IEEE Conference of Communications (INFOCOM)*, 2005.

[7] K. Medepalli and F. A. Tobagi, “Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications,” in *Proceedings of the 25th IEEE Conference on Computer Communications, (INFOCOM 2006)*, Barcelona, Spain, Apr. 2006.

[8] M. M.Hira, F. A. Tobagi, and K. Medepalli, “Throughput analysis of a path in an IEEE 802.11 multihop wireless network,” in *Proceedings of the IEEE Wireless Communications and Networking Conference, (WCNC 2007)*, Hong Kong, China, Mar. 2007.

[9] M. Garetto, T. Salonidis, and E. W. Knightly, “Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 864–877, Aug. 2008.

[10] J. S. Baras, V. Tabatabaee, G. Papageorgiou, and N. Rentz, “Modelling and Optimization for Multi-hop Wireless Networks Using Fixed Point and Automatic Differentiation,” in *Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt’08), Berlin, Germany*, March 31 - April 4 2008.

[11] F. Kelly, “Loss networks,” *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, 1991.

[12] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*. Springer Telecommunications Networks and Computer Systems, 1995.

[13] F. Kelly, “Blocking probabilities in large circuit switched networks,” *Adv. Appl. Prob.*, vol. 18, pp. 473–505, 1986.

[14] D. Mitra, J. Morrison, and K. Ramakrishnan, “Atm network design and optimization: a multirate loss network framework,” *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, pp. 531–543, Aug. 1996.

[15] A. Kashper, S. Chung, and K. Ross, “Computing approximate blocking probabilities with state-dependent routing,” *IEEE/ACM Trans. Netw.*, vol. 1, no. 1, pp. 105–115, Feb. 1993.

[16] R. Srikanth and A. Greenberg, “Computational techniques for accurate performance evaluation of multirate, multihop communications networks,” *IEEE J. Sel. Areas Commun.*, vol. 5, no. 2, pp. 266–277, Feb. 1997.

[17] M. Liu and J. Baras, “Fixed point approximation for multirate multihop loss networks with state-dependent routing,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 361–374, Apr. 2004.

[18] M. Bcker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, *Automatic Differentiation: Applications, Theory and Implementations*. Birkhuser, 2006.

[19] “ADOL-C,Automatic Differentiation by Overloading in C++,” http://www.coin-or.org/projects/ADOL-C.xml.

[20] G. M. Louth, “Stochastic networks: Complexity, dependence and routing,” University of Cambridge, Ph.D. Thesis, 1990.

[21] G. D. Mulligan and D. G. Corneil, “Corrections to bierstone’s algorithm for generating cliques,” *Journal of the ACM*, vol. 19, no. 2, pp. 244–247, April 1972.

[22] C. Bron and J. Kerbosch, “Finding all cliques of an undirected graph,” *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, September 1973.

[23] C. Young and J. A. Stevens, “Artery nodes,” United State Patent No: US 7397810 B1, 8 July 2008.

[24] ——, “Heuristics for combining inter-channel and intra-channel communications in a wireless communications environment,” United State Patent No: US 7385999 B1, 10 June 2008.

[25] S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets,” *Algorithmica*, vol. 20, no. 4, pp. 374–387, April 1998.

[26] L. Kou, G. Markowsky, and L. Berman, “A fast algorithm for steiner trees,” *Acta Informatica*, vol. 15, pp. 141–145, 1981.

[27] S. Perumal, J. S. Baras, C. J. Graff, and D. G. Yee, "Aerial platform placement algorithms to satisfy connectivity, capacity and survivability constraints in wireless ad-hoc networks," in *Proceedings of the Military Communications Conference (Milcom) 2008*, Nov. 2008.

[28] M. Pinkey, D. Hampel, and S. DiPierro, "Unmanned aerial vehicle (uav) communications relay," in *MILCOM '96, Conference Proceedings, IEEE*, Oct. 1996.

[29] G. Djuknic, J. Freidenfelds, and Y. Okunev, "Establishing wireless communications services via high-altitude aeronautical platforms: a concept whose time has come?" *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 128–135, Sep. 1997.

[30] T. Tozer, D. Grace, J. Thompson, and P. Baynham, "Uavs and haps-potential convergence for military communications," in *IEE Colloquium on Military Satellite Communciations*, June 2000.

[31] D. Gu, G. Pei, H. Ly, M. Gerla, B. Zhang, and X. Hong, "Uav aided intelligent routing for ad-hoc wireless network in single-area theater," in *In Proceeding of IEEE WCNC 2000*, 2000, pp. 1220–1225.

[32] J. Hillmand, S. Jones, R. Nichols, and I. Wang, "Communications network architectures for the army future combat system and objective force," in *MILCOM 2002, Conference Proceedings, IEEE*, Oct. 2002, pp. 1417–1421 vol. 2.

[33] I. Rubin and R. Zhang, "Performance behavior of unmanned vehicle aided mobile backbone based wireless ad hoc network," in *Proceedings of IEEE Vehicular Technology Conference VTC 2003*, April 22-25 2003.

[34] P. Basu, J. Redi, and V. Shurbanov, "Coordinated flocking of uavs for improved connectivity of mobile ground nodes," in *MILCOM 2004, Conference Proceedings, IEEE*, 31 Oct - 3 Nov 2002, pp. 1628–1634 vol. 3.

[35] M. Raissi-Dehkordi, K. Chandrashekar, and J. Baras, "Uav placement for enhanced connectivity in wireless ad-hoc networks," University of Maryland, College Park, CSHCN Technical Report 2004-18, 2004.

[36] K. Rose, "Deterministic annealing for clustering, compression, classification, regression and related optimization problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210–2239, Nov. 1998.

[37] R. J. Fowler, M. Paterson, and S. L. Tanimoto, "Optimal packing and covering in the plane are np-complete," *Information Processing Letters*, vol. 12, no. 3, pp. 133–137, Jun. 1981.

[38] K. Rose, E. Gurewitz, and G. Fox, "Constrained clustering as an optimization method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 8, pp. 785–794, Aug. 1993.

[39] K. Rose and D. Miller, “Constrained clustering for data assignment problems with examples of module placement,” in *Proceedings of the IEEE Int. Symp. Circuits and Systems*, vol. 4, San Diego, CA, May 1992, pp. 1937–1940.

[40] J. Buhmann and H. Kuhnel, “Vector quantization with complexity costs,” *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1133–1145, Jul. 1993.