UNIVERSITY OF CALIFORNIA, SAN DIEGO

Capacity Consideration of Wireless Ad Hoc Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Yusong Tan

Committee in charge:

Professor Anthony Acampora, Chair
Professor Massimo Franceschetti
Professor Srikanth Krishnamurthy
Professor Joseph Pasquale
Professor Curt Schurgers

2008

The dissertation of Yusong Tan is approved, and is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2008

To my parents, Xiaodan Tan and Baoling Ma,

and my grandmother, Zhufen Yang.

In memory of my grandfather, Xiuxian Ma

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would first and foremost like to thank my advisor Professor Anthony Acampora for his patience, encouragement, and valuable advice in the course of my doctoral research at UCSD. I would like to acknowledge the Center for Wireless Communications and the Department of Electrical and Computer Engineering for their financial support throughout my studies as well as the great academic environment they provided.

I would like to express gratitude and appreciation to members of my doctoral committee, Professor Joseph Pasquale, Professor Massimo Franceschetti, Professor Curt Schurgers, and Professor Srikanth Krishnamurthy for their attendance, approval, time and effort. I would like to thank Professor Rene Cruz as well for his presence at my preliminary examination.

I would like to thank John Minan, Karol Previte, M'Lissa Michelson, and the rest of the Department of Electrical and Computer Engineering staff for their support and advice. I would also like to thank my fellow graduate students for their friendship and support along the way. In particular, I would like to thank Sumit Bhardwaj.

Finally, I would like to thank my parents and grandparents for the unconditional love and support they have showed over the years.

Chapter 3, in part, is a reprint of the material in the following papers: M.Y. Tan and A. Acampora, "Capacity Estimation of Peer-to-Peer Networks Based on a Flow Deviation Approach," *IEEE International Symposium on Personal, Indoor, Mobile Radio Communications* (PIMRC), Athens, Sept. 2007; paper to be submitted to *IEEE Transactions on Wireless Communications*. The dissertation author was the primary investigator and author of this paper, and the co-author listed in these publications directed and supervised the research which forms the basis of this dissertation.

Chapter 4, in part, is a reprint of the material to be submitted to *IEEE Transac-*

tions on Wireless Communications. The dissertation author was the primary investigator and author of this paper, and the co-author listed in these publications directed and supervised the research which forms the basis of this dissertation.

Chapter 5, in part, is a reprint of the material to be submitted to *IEEE Transactions on Wireless Communications*. The dissertation author was the primary investigator and author of this paper, and the co-author listed in these publications directed and supervised the research which forms the basis of this dissertation.

VITA

1981	Born, Chengdu, China
2001-2003	Research Assistant Center for Research in Wireless Communications Clemson University, Clemson, SC
2003	Bachelor of Science, Summa Cum Laude Electrical Engineering Clemson University
2006	Master of Science Electrical Engineering (Communication Theory and Systems) University of California, San Diego
2004-2008	Research Assistant Center for Wireless Communications University of California, San Diego
2006-2008	Teaching Assistant Department of Electrical and Computer Engineering University of California, San Diego
2008	Doctor of Philosophy Electrical Engineering (Communication Theory and Systems) University of California, San Diego

ABSTRACT OF THE DISSERTATION

Capacity Consideration of Wireless Ad Hoc Networks

by

Yusong Tan

Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California, San Diego, 2008

Professor Anthony Acampora, Chair

The focus of this dissertation is on the fundamental capacity bounds of wireless ad hoc networks. We establish upper and lower bounds on the capacity to shed light on what is theoretically possible and what is currently achievable.

In the first part of the dissertation, we introduce and describe a method to find an upper bound on the capacity of wireless networks with arbitrary topology, size and traffic demands. The upper bound not only provides a yardstick against which the throughput of an existing wireless ad hoc network scheme can be gauged, it also provides insight into how to design better routing and medium access control protocols for wireless networks. Using the upper bound, we examine the behavior of networks of different size, under different channel conditions, and with different traffic patterns. Numeric results indicate that, when the channel conditions are known precisely, shadow and multipath

fading increase capacity; and that the capacity increases with network size when full traffic patterns are considered but decreases when directional traffic patterns are considered.

In the second part, we obtain the performance of an optimistic protocol based on CSMA/CA and compare it against the upper bound. There is a significant gap between the two results, especially when considering large networks operating in the high SNR region.

In the third part, we describe a new time-division scheduling scheme derived from the upper bound. In addition, we make an improvement to the schedules that increases the capacity significantly, even for small to medium size networks. Our schedules perform better than the protocol based on CSMA/CA, for medium to high SNR regions. Moreover, they also perform well against the upper bound when there is a directional traffic pattern, but not as well when there is a full traffic pattern. Finally, we examine the effects of time-varying fading and mobility on the schedules. We conclude that our schedules perform well in an environment where the channel changes slowly relative to the schedule update rate.

CHAPTER 1

Introduction

1.1 Wireless Ad Hoc Networks

A wireless ad hoc network is a collection of wireless mobile nodes that self-configure to form a network without the aid of any established infrastructure (Fig. 1.1). Without an inherent infrastructure, nodes are responsible for network control and management themselves. A node may communicate with any other node by establishing peer-to-peer connections. Depending on the distance between two nodes, their connection may either be a direct connection that is consisted of a single hop or a multihop connection, where data is relayed to the destination through a series of intermediate nodes; thus, nodes must cooperate to store-and-forward data on behalf of other nodes. Wireless ad hoc networks are very appealing for many reasons. Nodes may be distributed rapidly and randomly without requiring the establishment of a central control, which can prove expensive, such as the construction of a cellular tower, or infeasible, such as in a hostile environment. In addition, node redundancy and the lack of single points of failure provide the robustness that infrastructure-based networks lack. These properties are especially important for military applications, and the study of wireless ad hoc networks became an intense research topic under the name of ”packet radio network” [1, 2, 3], supported by the Defense Advanced Research Projects Agency (DARPA) through the 1970’s and 1980’s. By the early 1990’s, research on wireless ad hoc network has intensified due to the proliferation of inexpensive and widely available wireless devices and

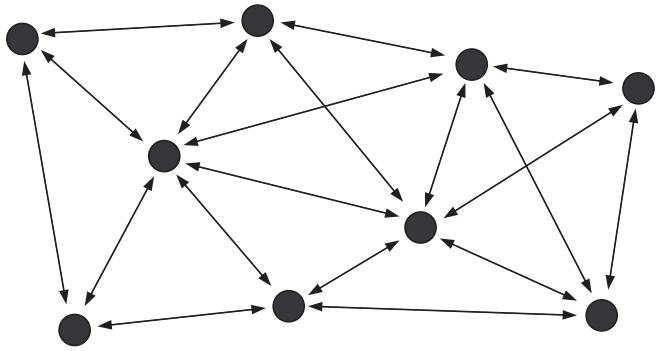

Figure 1.1 *An example of a wireless ad hoc network.*

the network community's interest in mobile computing [4, 5, 6, 7]. The robustness of a wireless ad hoc network makes it flexible enough to be tailored towards many different types of applications. They include data networks [8], home networks [9, 10], device networks [11], sensor networks [12, 13, 14] and distributed control systems.

These different applications have a wide range of network requirements. In order to meet the various application requirements, there have been extensive research directed towards different levels of the network stack. More specifically, designs in wireless ad hoc network protocols are largely based on a layered approach, such as according to the Open Systems Interconnection (OSI) model. Each layer is reasonably self-contained so the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers. Examples of research and design for some of the layers include multiple antennas, coding, power control and adaptive techniques at the physical layer, power control and scheduling at the medium access control (MAC) layer, energy-constrained and delay-constrained routing at the network layer, and application adaptation at the application layer [15].

The focus of this dissertation is on the fundamental capacity limit of a wireless ad hoc network, which itself cannot be categorized under any one layer in the OSI model;

instead, the goal of finding the capacity limit is to shed light on good design principles in the different layers, e.g., routing, scheduling. In the next section, we summarize some of the current research on capacity of wireless ad hoc networks.

1.2 Current Research on Capacity of Wireless Ad Hoc Networks

Current research on the capacity of wireless ad hoc networks can be divided into two main categories. In one category, one seeks the asymptotic bounds by allowing the number of nodes in a network to grow to infinity, with the goal of obtaining insights into the behavior of large networks. In the other category, one seeks numeric bounds of networks of some fixed set of parameters, e.g., network size, traffic pattern, topology, usually using computer simulation involving integer linear programming (ILP) or Monte Carlo methods.

Asymptotic Approach

In the seminal work by Gupta and Kumar [16], the authors have shown that, for the protocol model, the total throughput of a network of arbitrarily located nodes scales as $\sqrt{n}$, and the total throughput of a random network scales as $\sqrt{n/\log n}$ with high probability. It is shown that similar results hold for the physical model. Other works that produced similar results are [17, 18]. Moreover, in [18], the author described schemes that achieved total throughput that scales as $n^{\frac{1}{2}}(\log n)^{-\frac{3}{2}}$ and $n^{\frac{d+1}{2}}(\log n)^{-\frac{5}{2}}$, under a general model of fading without and with mobility, respectively, for $0 < d < 1$. In [19], the authors have shown, by using percolation theory, that the total throughput of a random network scales as $\sqrt{n}$ instead of $\sqrt{n/\log n}$.

It must be noted that the aforementioned works are decidedly non-information theoretic. The open question is whether the results obtained there, more specifically the upper bound, can be confirmed from a information theoretic point of view. A first con-

firmation was made by Xie and Kumar [20], but they assumed the signals are strongly attenuated over distance, i.e., an attenuation function of $\frac{1}{r^\alpha}$ with $\alpha > 6$. In [21], Lévêque and Telatar provided another information theoretic proof, but with only minimal assumption on the attenuation function ($\alpha > 2$). An alternate proof of Lévêque and Telatar's results was given in [22].

The bounds by Gupta and Kumar showed that, for a network with immobile nodes, the per-node throughput tends to zero as the number of nodes in the network increases. These results are a pessimistic indication of the scalability of wireless ad hoc networks. However, Grossglauser and Tse [23] have shown that mobility actually improves network capacity, at the sacrifice of large delays, and that the per-node throughput is $\Theta(1)$. This result holds true even when mobility is constrained to a randomly placed line segment [24].

Numeric Approach

While asymptotic bounds for wireless ad hoc networks provide us the insight into the behavior of large networks, it does not shed light on the behavior of small to medium sized networks. In addition, given a set of network parameters, we may wish to obtain numeric bounds on the network capacity, which would provide us with a benchmark against which current performance is compared. The general approach to obtain numeric bounds is to use either ILP or Monte Carlo methods.

In [25], the authors proposed a constructive method to obtain the capacity region of a wireless ad hoc network by finding the convex hull of the basic rate matrices representing the network. Using their approach, one can numerically find the capacity of the network given the traffic and channel gain matrix. However, that method becomes impractical for networks with more than 15 nodes because the number of basic rate matrices increases very rapidly with n.

In [26], the authors addressed the inefficiency in resource utilization of Time

Division Multiple Access (TDMA) by proposing a Spatial TDMA scheme in which time slots are reused by nodes that are sufficiently far apart. The authors presented set covering formulations to model the resource optimization problem for both node-oriented and link-oriented allocation strategies. The method is able to simulate networks of up to 40 nodes; however, it does not consider the traffic pattern.

In [27], a probabilistic approach to the calculation of capacity bounds is considered. The authors modeled the effective throughput of a random network as a random variable and calculated its expected value using Monte Carlo methods. The approach is more scalable than the methods based on ILP and can simulate networks of up to 200 nodes.

1.3 Contributions

In this dissertation, we study the capacity of wireless ad hoc networks. Given a set of network parameters, the methods described herein will produce an upper and lower bound on the capacity of the network. Our contributions per chapter are as follows:

- In Chapter 3, we introduce and describe the Flow Deviation (FD) method to calculate an upper bound on the network capacity. The upper bound, however, is unachievable with high probability as the number of nodes increases. We describe this limitation of the FD method and present a proof that at least 1/3 of this upper bound is achievable, in the absence of interference. We conclude the chapter with some results and present the computational complexity of the FD algorithm, which allows for the simulation of networks consist of 100 nodes or more, depending on the traffic pattern.

- In Chapter 4, we describe and simulate a protocol similar to Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), but under ideal conditions.

The goal of this chapter is to gauge the performance of a realistic protocol, albeit very optimistic, in the presence of interference by comparing it against the FD upper bound. We conclude by noting the difference between the FD upper bound and the protocol performance, and shedding light on areas where improvements can be made.

- In Chapter 5, we describe a new time-division (TD) scheduling scheme that is derived from the FD results. We also describe an improvement to the TD schedules that can increase the network capacity significantly. We follow by comparing the performance of our TD schedules against the FD upper bound and the CSMA/CA performance. Finally, we conclude by examining the impact of time-varying fading and mobility on the effectiveness of our TD schedules.

CHAPTER 2

System Model

Consider n nodes denoted by $N_1, N_2, \ldots, N_n$ distributed over a square area of $[0, \sqrt{n}-1] \times [0, \sqrt{n}-1]$. Each node has a transmitter, receiver, and infinite buffer. Each node may wish to transmit to some or all of the other nodes, either directly or via multihop routing. We assume nodes cannot simultaneously transmit and receive; thus, only half-duplex communication may be established. We assume all nodes transmit at some maximum power level P and all transmissions occupy the full bandwidth W of the channel. Let the traffic matrix be defined as $\boldsymbol{T} = \{T_{ij}\}$, where T_{ij} specifies the relative traffic between nodes N_i and N_j, $i, j \in \{1, 2, \ldots, n\}$. The diagonal elements of $\boldsymbol{T}$ are unimportant, so we can set them to zero. Let the channel gain matrix be defined as $\boldsymbol{H} = \{H_{ij}\}$, where H_{ij} specifies the channel gain between nodes N_i and N_j. The diagonal elements of $\boldsymbol{H}$ are unimportant, so they are set to zero. We also assume symmetric channels so $H_{ij} = H_{ji}$[1]. When N_i transmits, N_j receives the signal at power level $H_{ij}P$. The signal is corrupted by the presence of thermal noise, background interference and interference from other users. We model the thermal noise and background interference jointly as additive white Gaussian noise (AWGN) with one-sided power spectral density N_0.

Let's define the set of concurrent transmitters at an instance in time as $\{N_t, t \in \mathcal{T}\}$. When N_j is receiving a transmission from N_i, its signal to interference and noise

[1] In general, symmetry of the channel gain matrix is not required for the FD method

ratio (SINR) is defined as

$$\zeta_{ij} = \frac{H_{ij}P}{WN_0 + \sum_{k\in\mathcal{T},k\neq i} H_{kj}P} = \frac{\gamma_{ij}}{1 + \sum_{k\in\mathcal{T},k\neq i} \gamma_{kj}}, \tag{2.1}$$

where

$$\gamma_{ij} = \frac{H_{ij}P}{WN_0}$$

is the signal-to-noise ratio (SNR) at N_j.

The rate R at which N_i transmits to N_j is a function of SINR and, for our purposes, we use the Shannon capacity function

$$f(\zeta_{ij}) = W \log_2 (1 + \zeta_{ij}). \tag{2.2}$$

Under the Shannon assumption, bit error probability can be made arbitrarily small as long as R is less than or equal to the capacity $c_{ij} = f(\zeta_{ij})$. We assume a receiver N_j can always correctly decode a packet, regardless of its SINR, as long as the transmitter N_i is transmitting at a rate equal to the capacity of the channel, i.e., $R = c_{ij}$. This implies that the transmitter can continuously adapt its transmission rate to meet the SINR requirement at the receiver.

In the following chapters, instead of explicitly specifying the transmit power P, we specify the normalizing SNR γ^{norm}. If we consider the area of distribution a square grid and place the nodes unit distance from each other, then ignoring the effects of shadowing and multipaths, the SNR between neighboring nodes is defined by γ^{norm}. We do this so that our results can be appropriately scaled to apply to a network with an arbitrary number of nodes distributed over an arbitrary area. For example, results for a network with 100 nodes distributed over an area of 81 square unit distance with $\gamma^{norm} = 20$dB can be applied to an actual network of 100 nodes distributed over, say, 8100 square-meters as long as the SNR between nodes 10 meters apart is 20dB. An example of a 9-node constellation is shown in Fig. 2.1.

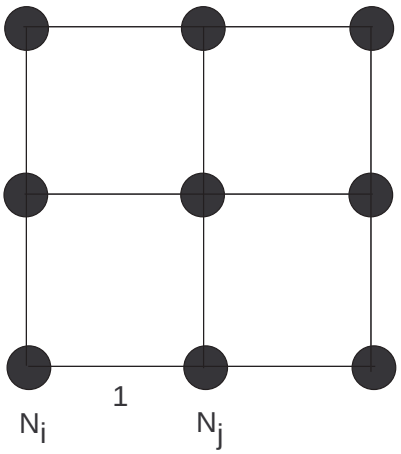

Figure 2.1 *The square grid for a 9-node network on which the coverage area and* γ^{norm} *are defined. The area is equal to* $4 = (\sqrt{9} - 1) \times (\sqrt{9} - 1)$*, and* γ^{norm} *is the SNR between any two nodes unit distance from each other. For example,* γ_{ij}^{norm} *is the SNR between* N_i *and* N_j*.*

CHAPTER 3

Upper Bound on the Capacity of Wireless Ad Hoc Networks

3.1 Introduction

In this chapter we describe a method to find an upper bound on the capacity of wireless networks with arbitrary topology, size and traffic demands. The upper bound is of interest because it not only provides a yardstick against which the throughput of an existing wireless ad hoc network scheme can be gauged, it also provides insight into how to design better routing and medium access control protocols for wireless networks.

The chapter is organized as follows: In Section 3.2, we introduce the Flow Deviation (FD) method, which we will use to find an upper bound on the capacity of wireless ad hoc networks. In Section 3.3, we describe how to formulate the capacity problem into a form that can be solved using the FD method. In Section 3.4, we describe a limitation of the FD method, which causes the resulting upper bound to become increasingly "loose" as the size of the network grows. We rectify this by providing a mathematical proof that an achievable upper bound is within a constant factor of the FD upper bound. In Section 3.5, we present and analyze some simulation results. In Section 3.6, we describe some computational issues. We conclude in Section 3.7.

3.2 The Flow Deviation Method

Suppose we have a network of n nodes, $(N_1, N_2, \ldots, N_n)$, that needs to route a quantity r_{ij} of type (i, j) commodity from N_i to N_j. The multicommodity (m.c.) flow problem consists of finding routes for all such commodities, which minimizes (or maximizes) a well-defined performance function (e.g., total cost, delay or capacity), while satisfying a set of constraints (e.g., channel capacity constraint). The most general m.c. flow problem can be expressed in the following way:

Given: A network of n nodes and b arcs

An $n \times n$ matrix $\boldsymbol{R} = \{r_{ij}\}$, called the requirement matrix,

whose entries are non-negative

Minimize: (or maximize) $P(\Phi)$

where Φ is the flow configuration

and P is a well-defined performance function

Constraints: 1. Conservation of flow must hold for each node:

$$\sum_{k=1}^{n} f_{kl}^{(ij)} - \sum_{m=1}^{n} f_{lm}^{(ij)} = \begin{cases} -r_{ij}, & \text{if } l = i \\ +r_{ij}, & \text{if } l = j \\ 0, & \text{else} \end{cases} \tag{3.1}$$

2. Non-negativity of flow:

$$f_{kl}^{(ij)} \geq 0, \forall i, j, k, l. \tag{3.2}$$

where $f_{kl}^{(ij)}$ is the amount of commodity (i, j)

flowing on arc (k, l).

3. Additional constraints on Φ (3.3)

We call any flow configuration F that satisfies Equations (3.1)-(3.3) a feasible m.c. flow for the requirement $\boldsymbol{R}$.

The FD method was introduced in 1973 by Fratta, Gerla, and Kleinrock [28] to solve the flow problem where the goal is to minimize (or maximize) some well-defined performance function for a given topology and external flow requirements. It is used to find minima (or maxima) of unconstrained [1], non-linear, differentiable m.c. flow problems. We present an overview of the FD method next.

The (i, j) *commodity flow* $\boldsymbol{f}^{(ij)}$ is defined as:

$$\boldsymbol{f}^{(ij)} \triangleq \left(f_1^{(ij)}, f_2^{(ij)}, \ldots, f_b^{(ij)}\right)$$

where $f_k^{(ij)}$ is the portion of (i, j) commodity flowing in arc k, and the *global flow* $\boldsymbol{f}$ is defined as:

$$\boldsymbol{f} = \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{f}^{(ij)}$$

The performance function $P(\Phi)$ can be any generally well-defined function, but for our purpose it is sufficient for us to define it as a function of the global flow only. Thus, we have $P(\Phi) = P(\boldsymbol{f})$.

A flow $\boldsymbol{f}$ is defined as *stationary* if, for any infinitesimal perturbation $\delta\boldsymbol{f}$ (such that $\boldsymbol{f} + \delta\boldsymbol{f}$ is also an m.c. flow), we have

$$P(\boldsymbol{f} + \delta\boldsymbol{f}) \geq P(\boldsymbol{f}).$$

The most general perturbation around $\boldsymbol{f}$ can be obtained as a convex combination of $\boldsymbol{f}$ with any m.c. flow $\boldsymbol{v}$. The result is expressed as:

$$\boldsymbol{f'} = (1 - \lambda)\boldsymbol{f} + \lambda\boldsymbol{v} = \boldsymbol{f} + \lambda(\boldsymbol{v} - \boldsymbol{f})$$

[1] This means that there are no additional constraints in Equation (3.3)

where

$$\boldsymbol{v} \in F,\ 0 \leq \lambda \leq 1$$

For $\lambda = \delta\lambda \ll 1$, we have

$$\delta P(\boldsymbol{f}) \triangleq P(\boldsymbol{f'}) - P(\boldsymbol{f}) \cong \delta\lambda \sum_{k=1}^{b} l_k(v_k - f_k) \tag{3.4}$$

where

$$l_k = \frac{\partial P}{\partial f_k}$$

From Equation (3.4) and from the definition of stationarity, $\boldsymbol{f}$ is stationary if

$$\sum_{k=1}^{b} l_k(v_k - f_k) \geq 0,\ \forall \boldsymbol{v} \in F \tag{3.5}$$

Let's denote the perturbation around $\boldsymbol{f}$ with an operator, $\mathrm{FD}(\boldsymbol{v}, \lambda)\odot$. It maps an m.c. flow $\boldsymbol{f}$ into another m.c. flow $\boldsymbol{f'}$ and is defined as:

$$\mathrm{FD}(\boldsymbol{v}, \lambda) \odot \boldsymbol{f} \triangleq (1 - \lambda)\boldsymbol{f} + \lambda\boldsymbol{v} = \boldsymbol{f'}$$

where $\boldsymbol{v}$ is a properly chosen feasible m.c. flow and $0 \leq \lambda \leq 1$ is the step size.

Finally, assuming a feasible initial flow $\boldsymbol{f}^0 \in F$ can be found, the following general algorithm for the FD method will allow us to find a pair $(\boldsymbol{v}, \lambda)$ in such a way that the repeated application of $\mathrm{FD}(\boldsymbol{v}, \lambda)\odot$ produces a sequence $\{\boldsymbol{f}^n\}$ which converges to a stationary flow.

1. Find a feasible starting flow $\boldsymbol{f}^0$
2. Let $m = 0$
3. $\boldsymbol{f}^{m+1} = \mathrm{FD}(\boldsymbol{v}^m, \lambda^m) \odot \boldsymbol{f}^m$

4. If $\{P(\boldsymbol{f}^m) - P(\boldsymbol{f}^{m+1})\} < \epsilon$, (or if $\sum_{k=1}^{b} l_k(f_k^m - v_k^m) < \epsilon'$)

   where ϵ and ϵ' are acceptable positive tolerances, stop.

   Otherwise, let $m = m + 1$ and go to 3.

The algorithm converges to stationary points; however, the only stationary points of stable equilibrium are the local minima, so we can assume that the algorithm converges to local minima.

3.3 Capacity Problem Formulation

In the previous section we presented an overview of the FD algorithm, which can be used to find local minima of unconstrained, non-linear, differentiable m.c. flow problems. We now show that we can formulate the capacity problem into the form that the FD method requires.

We want to minimize a performance function that captures the most basic notion of the capacity of a wireless ad hoc network; that is, a network is at full capacity if at least one of its nodes is fully utilized (i.e., the sum of the fractions of time a node is in transmission and reception is 1). Let us defined the utilization function for N_i as

$$g^i(\boldsymbol{f}) = \sum_{j=1}^{n} \frac{f_{ij}}{c_{ij}} + \sum_{k=1}^{n} \frac{f_{ki}}{c_{ki}} \tag{3.6}$$

where f_{ij} is the amount of flow on link (i, j) in bits/sec. Note that the first sum corresponds to the fraction of time that N_i is transmitting and that the second sum corresponds to the fraction of time that N_i is receiving. Thus, $g^i(\boldsymbol{f})$ is the fraction of time N_i is actively transmitting or receiving. If $g^i(\boldsymbol{F}) = 1$, we say N_i is critical. If $g^i(\boldsymbol{f}) > 1$, we say it is overly-critical. Otherwise, we say it is not critical.

One way to express the capacity problem is as follows:

Given: $\boldsymbol{T}$ and $\boldsymbol{H}$

Objective: maximize k

Constraint: be able to deliver traffic $k\boldsymbol{T}$ without any nodes becoming overly-critical.

Thus, we seek to find the maximum factor by which $\boldsymbol{T}$ may be scaled. If we define

$$\psi(\boldsymbol{f}) = \max_{i \in \boldsymbol{n}} g^i(\boldsymbol{f}) \tag{3.7}$$

where $\boldsymbol{n} = \{1, 2, \ldots, n\}$, then the above objective is equivalent to first minimizing $\psi(\boldsymbol{f})$ and then scaling $\boldsymbol{T}$ so that $\psi(\boldsymbol{f}) = 1$. Mathematically, we want to solve the unconstrained optimization problem

$$\min_{\text{feasible } \boldsymbol{f}} \psi(\boldsymbol{f}) \tag{3.8}$$

and assign $k = 1/\psi(\hat{\boldsymbol{f}})$, where $\hat{\boldsymbol{f}}$ is the global flow that minimizes $\psi(\cdot)$. Thus, we seek to find the flow, for a given $\boldsymbol{T}$, for which the utilization of the maximally utilized node is minimized. We can then scale $\boldsymbol{T}$ by a factor k such that this utilization equals 100%. If we were to scale $\boldsymbol{T}$ by a larger factor, then at least one of the nodes would be utilized more than 100% of the time, regardless of the flow.

In order to find a local minimum to the problem in Equation (3.8), we must find a stationary flow such that Equation (3.5) is satisfied. To achieve that goal, we must first determine what $l_k = \frac{\partial P}{\partial f_k}$ and v_k are for our problem. Since a $max(\cdot)$ function operates on a subset of functions within its arguments, we substitute the definition of partial derivatives with the definition of directional derivatives. We call the following theorems and corollary from [29]:

Theorem 1. *Consider the function* $\psi(x) = \max_{j \in \boldsymbol{m}} f^j(x)$*, with* $f^j : \mathbb{R}^n \to \mathbb{R}$ *continuously differentiable. Then the directional derivative* $d\psi(x;h)$ *exists for all* x, $h \in \mathbb{R}^n$ *and is given by*

$$d\psi(x;h) = \max_{j \in I(x)} \langle \nabla f^j(x), h \rangle. \tag{3.9}$$

where $I(x) \triangleq \{j \in \boldsymbol{m} | \psi(x) = f^j(x)\}$ is the maximizing set, ∇ is the gradient operator, and $\langle\rangle$ is the dot product operator .

Theorem 2 (Danskin)**.** *Suppose that the functions* $f^j : \mathbb{R}^n \to \mathbb{R}$ *in* $\psi(x) = \max_{j \in \boldsymbol{m}} f^j(x)$ *are continuously differentiable and that* $\hat{x}$ *is a local minimizer of* $\psi(\cdot)$*. Then*

$$d\psi(\hat{x};h) \geq 0, \forall h \in \mathbb{R}^n \tag{3.10}$$

Corollary 1. *Suppose that the functions* $f^j : \mathbb{R}^n \to \mathbb{R}$ *in* $\psi(x) = \max_{j \in \boldsymbol{m}} f^j(x)$ *are convex and continuously differentiable. Then* $\hat{x}$ *is a global minimizer of* $\psi(\cdot)$ *if and only if Equation* (3.10) *holds.*

It is easy to see that the functions $g^j(\boldsymbol{f})$ in Equation (3.7) is convex and continuously differentiable since it is a linear function of $\boldsymbol{f}$.

We define our *general search direction* $\hat{\boldsymbol{h}}$ as follows:

1. For each link $(i,j),$ where $i,j \in \boldsymbol{n},$ set its cost to $1/c_{ij}$.
2. For each element T_{ij} of the traffic matrix,

   if $i,j \notin I(\boldsymbol{f})$, then go to Step 2a.

   If $i \in I(\boldsymbol{f}), j \notin I(\boldsymbol{f})$, then go to Step 2b.

   If $j \in I(\boldsymbol{f}), i \notin I(\boldsymbol{f})$, then go to Step 2c.

   Else if $i,j \in I(\boldsymbol{f})$, then go to Step 2d.

2a. Delete all N_k, $k \in I(\boldsymbol{f}),$ and route T_{ij} using shortest-route.

2b. Delete all N_k, $k \in I(\boldsymbol{f}),$ except N_i.

Delete all N_i's non-smallest cost egress links.

Route T_{ij} using shortest-route.

2c. Delete all N_k, $k \in I(\boldsymbol{f}),$ except N_j.

Delete all N_j's non-smallest cost ingress links.

Route T_{ij} using shortest-route.

2d. Delete all N_k, $k \in I(\boldsymbol{f}),$ except N_i and N_j.

Delete all N_i and N_j's non-smallest cost ingress and egress links.

Route T_{ij} using shortest-route.

3. Let $\tilde{\boldsymbol{f}}$ be the resulting new flow after Step 2, and let $\bar{\boldsymbol{f}}$ be the old flow resulted from the previous search.

Then $\hat{\boldsymbol{h}} = \tilde{\boldsymbol{f}} - \bar{\boldsymbol{f}}$.

In Step 1, we set the link costs equal to the elements of $\nabla g^i(\boldsymbol{f})$. In Step 2a, we disallow traffic to be routed through any maximally utilized nodes. In Steps 2b, the only maximally utilized node active is N_i because it is generating the traffic. In Step 2c, the only maximally utilized node active is N_j because it is receiving the traffic. In Step 2d, the only maximally utilized nodes active are N_i and N_j because the traffic is transported between them. We delete non-smallest cost links and route using shortest-route to minimize $\langle \nabla g^i(\boldsymbol{f}), \hat{\boldsymbol{h}} \rangle$ in Equation (3.9). Note that Equations (3.1) and (3.2) hold for any flow obtained by a shortest-route algorithm such as Dijkstra's algorithm [30]. Therefore, $\tilde{\boldsymbol{f}}$ is a feasible flow.

If we find a global flow $\hat{\boldsymbol{f}}$ such that $d\psi(\hat{\boldsymbol{f}}; \hat{\boldsymbol{h}}) \geq 0$, then we can conclude that $d\psi(\hat{\boldsymbol{f}}; \boldsymbol{h}) \geq 0, \forall \boldsymbol{h} \in \mathbb{R}^{n^2}$. This is easy to see because the only time N_i, $i \in I(\hat{\boldsymbol{f}})$, is

active is when it is transmitting traffic generated by itself or receiving traffic intended for itself. Also, when N_i is active, it is only using the links with the highest capacity. Therefore, if the utilization of N_i is not lowered in the flow direction $\hat{\boldsymbol{h}}$, then it cannot be lowered in any flow direction $\boldsymbol{h}$. We can now present our FD algorithm to solve the new objective given in Equation (3.8):

1. Find a feasible starting flow $\boldsymbol{f}^0$ by routing all traffic
   using shortest-route
2. Let $m = 0$
3. Determine $I(\boldsymbol{f}^m)$ using utilization functions $g^i(\boldsymbol{f}^m)$
4. Find $\tilde{\boldsymbol{f}}^m$ and $\hat{\boldsymbol{h}}^m$
5. If $d\psi(\boldsymbol{f}^m; \hat{\boldsymbol{h}^m}) > \epsilon'$
   where ϵ' is an acceptable negative tolerance, stop.
   Else, go to Step 6.
6. $\boldsymbol{f}^{m+1} = \mathrm{FD}(\tilde{\boldsymbol{f}}^m, \lambda^m) \odot \boldsymbol{f}^m$
7. Let $m = m + 1$ and go to Step 3.

Finally, if $\hat{\boldsymbol{f}}$ is the resulting global minimum flow, then we can obtain k with equation $k = 1/\psi(\hat{\boldsymbol{f}})$. The capacity of the network would be given as:

$$C = k \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij} \tag{3.11}$$

3.4 Flow Deviation Limitation

Wireless ad hoc networks are "packet radio networks," where flows are gathered in packets before being sent across the network. However, from Section 3.2 we see that the FD algorithm produces continuous flows instead of packetized flows. As a

consequence, in order to achieve the flow satisfying Equation (3.8), some nodes may be required to communicate with multiple nodes at any given time. For example, in a simple single-hop network with only three nodes where the traffic matrix and capacity matrix are respectively given as

$$\boldsymbol{T} = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{C} = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 0 \end{pmatrix}$$

the resulting flow from the FD algorithm would look like

$$\hat{\boldsymbol{f}} = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 2 & 0 & 0 \end{pmatrix}$$

This corresponds to each node having a utilization of 2, meaning that all of the traffic given in $\boldsymbol{T}$ can be delivered in 2 seconds, giving an upper bound on capacity of 3 bits/sec. However, if each node can only transmit or receive (but not both) from one other node, then the network actually requires 3 seconds to deliver the same traffic, giving an upper bound of only 2 bits/sec. Therefore, the flow provided by the FD algorithm gives a loose upper bound on the capacity of a network.

Definition 1. *A matrix $\boldsymbol{R}$ is called a rate matrix*[2] *if the following holds:*

$$R_{ii} = 0; R_{ij} \geq 0, i \neq j. \textit{ If } R_{ij} \textit{ is non-zero, then}$$
$$R_{ik} = 0, \forall k \neq j; \; R_{lj} = 0, \forall l \neq i; \; R_{mi} = R_{jm} = 0, \forall m$$

Each rate matrix specifies a valid TD schedule in the absence of interference. The non-zero elements in $\boldsymbol{R}$ are the corresponding link capacities (e.g., if R_{12} is non-

[2]From this point forward, $\boldsymbol{R}$ denotes rate matrix and not the requirement matrix of m.c. flow.

zero, then $R_{12} = c_{12}$).

Definition 2. *An achievable capacity C_a is one that satisfies:*

$$C_a = \sum_i \alpha_i \boldsymbol{R}_i, \qquad \sum_i \alpha_i \leq 1$$

where α_i is the fraction of time $\boldsymbol{R}_i$ is used.

Let us define C_u as the highest achievable capacity that can be obtained, possibly by using exhaustive search. We now make the following claim:

Claim 1. *The highest achievable capacity C_u cannot be less than 2/3 of the FD upper bound in special cases or less than 1/3 of the FD upper bound in the worst case.*

Proof. First, we introduce the normalized flow matrix $\boldsymbol{B}$ whose elements are defined as $B_{ij} = \hat{f}_{ij}/c_{ij}$, where $\hat{f}_{ij}$ is the element of the global minimum flow $\hat{\boldsymbol{f}}$ that corresponds to the flow on link (i, j). Next, we rewrite $\boldsymbol{B}$ as $\boldsymbol{B} = \beta\tilde{\boldsymbol{B}}$, where β is the largest row or column sum of $\boldsymbol{B}$. $\tilde{\boldsymbol{B}}$ has the property that there exists some $\hat{j}$ such that either the $\hat{j}$th column sum or $\hat{j}$th row sum or both equal to one. Now we pad the elements of $\tilde{\boldsymbol{B}}$ so that all of its column and row sums equal to one and call this matrix $\tilde{\boldsymbol{B}}^{pad}$. This is always possible because the padding problem can be constructed into a transportation problem where the total supply equals the total demand [31]. $\tilde{\boldsymbol{B}}^{pad}$ is a doubly stochastic matrix and can be decomposed according to the following theorem [32]:

Theorem 3 (Birkhoff-von Neumann)**.** *An $n \times n$ matrix over $\mathbb{R}$ is doubly stochastic if and only if it is a convex combination of permutation matrices.*

Let us digress for a second and present the following claim:

Claim 2. *A permutation matrix $\boldsymbol{P}$ with $m > 0$ off-diagonal and non-zero elements is the sum of at most three maximum size rate matrices.*

Proof. Ignoring the diagonal elements, $\boldsymbol{P}$ can be divided into $l \leq \lfloor \frac{m}{2} \rfloor$ disjoint rings. For each ring i, number the links consecutively. If ring i has an even number of links, then it can written as a sum of two rate matrices, $\boldsymbol{R}_1^i$ and $\boldsymbol{R}_2^i$. $\boldsymbol{R}_1^i$ contains all the odd links, and $\boldsymbol{R}_2^i$ contains all the even links. However, if ring i has an odd number of links, then it can be written as a sum of three rate matrices, $\boldsymbol{R}_1^i$, $\boldsymbol{R}_2^i$ and $\boldsymbol{R}_3^i$. $\boldsymbol{R}_1^i$ contains all the odd links, except the last link, $\boldsymbol{R}_2^i$ contains all the even links and $\boldsymbol{R}_3^i$ contains the last link. Because all the rings are disjoint, the three maximum size rate matrices can be written as:

$$\boldsymbol{R}_1 = \sum_{i=1}^{l} \boldsymbol{R}_1^i; \quad \boldsymbol{R}_2 = \sum_{i=1}^{l} \boldsymbol{R}_2^i; \quad \boldsymbol{R}_3 = \sum_{i=1}^{l} \boldsymbol{R}_3^i$$

□

From the Birkhoff-von Neumann theorem we know that $\tilde{\boldsymbol{B}}^{pad} = \sum_i \alpha_i \boldsymbol{P}_i$, where $\sum_i \alpha_i = 1$. If we can create valid TD schedules according to $\boldsymbol{P}_i$, then we can remove all the flows given in $\tilde{\boldsymbol{B}}^{pad}$ in one second. However, this is impossible. In fact, from the previous claim we know that each $\boldsymbol{P}_i$ can be decomposed into at most three maximum size rate matrices, each of which is a valid TD schedule. Therefore, the amount of time, t_{lb}, it takes to remove all the flows in $\tilde{\boldsymbol{B}}^{pad}$ is at most three seconds. On the other hand, the amount of time, t_{ub}, to remove the flows in $\tilde{\boldsymbol{B}}$, according to the FD upper bound, is equal to $\hat{j}$th column sum plus $\hat{j}$th row sum, and it is between one and two seconds[3]. Therefore, we have $1/3 \leq t_{ub}/t_{lb} \leq 2/3$, which implies $1/3 \leq k_{lb}/k_{ub} \leq 2/3$. □

[3]From this point forward, unless specified otherwise, whenever we mention the time it takes to remove a set of flows, we are always referring to the flow $\tilde{\boldsymbol{B}}$

3.5 Simulation Results and Analysis

We simulated our network in MATLAB, and the parameters we used are n, $\boldsymbol{T}$, $\boldsymbol{H}$ and γ^{norm}. The channel gains H_{ij} depend on the channel model we use. The propagation losses for a radio link are usually divided into three categories: path-loss, shadowing, and multipath fading [33, 34, 35]. When considering path-loss only, H_{ij} is given by

$$H_{ij} = \frac{1}{|X_i - X_j|^{\alpha}}, \tag{3.12}$$

where α is the path-loss coefficient and is set to $\alpha = 4$. When considering path-loss and shadowing, H_{ij} is given by

$$H_{ij} = S_{ij}\frac{1}{|X_i - X_j|^{\alpha}}, \tag{3.13}$$

where S_{ij} is a lognormal random variable with mean 0 and standard deviation $\sigma =$ 6 dB. So $S_{ij} = 10^{\eta_{ij}/10}$, where η_{ij} has Gaussian distribution with mean 0 and standard deviation $\sigma_N = 6$. Finally, when considering path-loss, shadowing and multipath fading, H_{ij} is given by

$$H_{ij} = M_{ij}S_{ij}\frac{1}{|X_i - X_j|^{\alpha}}, \tag{3.14}$$

where M_{ij} is an exponential random variable with mean 1.

We simulated networks of various sizes and different traffic patterns, with $\gamma^{norm} =$ 5, 10, or 20dB. For each scenario, we repeated the simulation 100 times, each time with node locations randomly chosen. An average FD upper bound on the network capacity is obtained, where the capacity is defined as the sum of all the elements of $k\boldsymbol{T}$. An average TD lower bound on the average highest achievable capacity C_u is obtained according to the proof of Claim 1. The lower bounds are calculated according to the formula t_{ub}/t_{lb}, where t_{ub} is the amount of time it takes to remove the traffic $\boldsymbol{T}$ according to the FD algorithm, and t_{lb} is the amount of time it takes to remove the same traffic using valid TD schedules. Equivalently, $k_{lb}/k_{ub} = t_{ub}/t_{lb}$, where $k_{lb}\boldsymbol{T}$ is the maximum amount of

traffic the network can deliver in one second using valid TD schedules, and $k_{ub}\boldsymbol{T}$ is the maximum amount of traffic the network can deliver in one second according to the FD algorithm.

3.5.1 Full Traffic Pattern

In this subsection, we evaluate networks of various sizes under heavy traffic; that is, we consider full traffic matrices, where each element, except for those on the diagonal, is Poisson distributed with mean 1. This type of traffic matrix characterize a general peer-to-peer network where any two nodes can communicate. Results are shown in Fig. 3.1. It can be seen from the figure that when there are few nodes, the FD upper bound and TD lower bound are tight, meaning that the highest achievable capacity is close to the upper bound. However, as the number of nodes increases, the TD lower bound approaches close to 2/3 the FD upper bound. This is a direct result of Claim 1. We have approximately 2 in the numerator because the traffic matrix we used is approximately symmetrical around the diagonal; therefore, the fraction of time a node is transmitting and the fraction of time it is receiving are approximately equal. We have 3 in the denominator because the larger the network the more likely that most of the permutation matrices must be decomposed into 3 maximum size rate matrices.

In Fig. 3.2, the FD upper bounds for all three channel models (Equations (3.12), (3.13) and (3.14)) are shown. It can be seen that, overall, shadowing and multipaths improves the capacity of the system, which agrees with a similar conclusion drawn in [36]. This is expected because the FD algorithm favors channels with high capacity and disfavors those in deep fade. When there are only a few nodes, the benefit of shadowing is small ($\sim 10\%$ gain), and the presence of multipaths actually negates any benefit from shadowing. When there are a large number of nodes, the benefit of shadowing and multipath becomes more significant ($\sim 78\%$ gain at 81 nodes). Also, at even moderate number of nodes, benefits from shadowing dominates the benefits from multipath. The

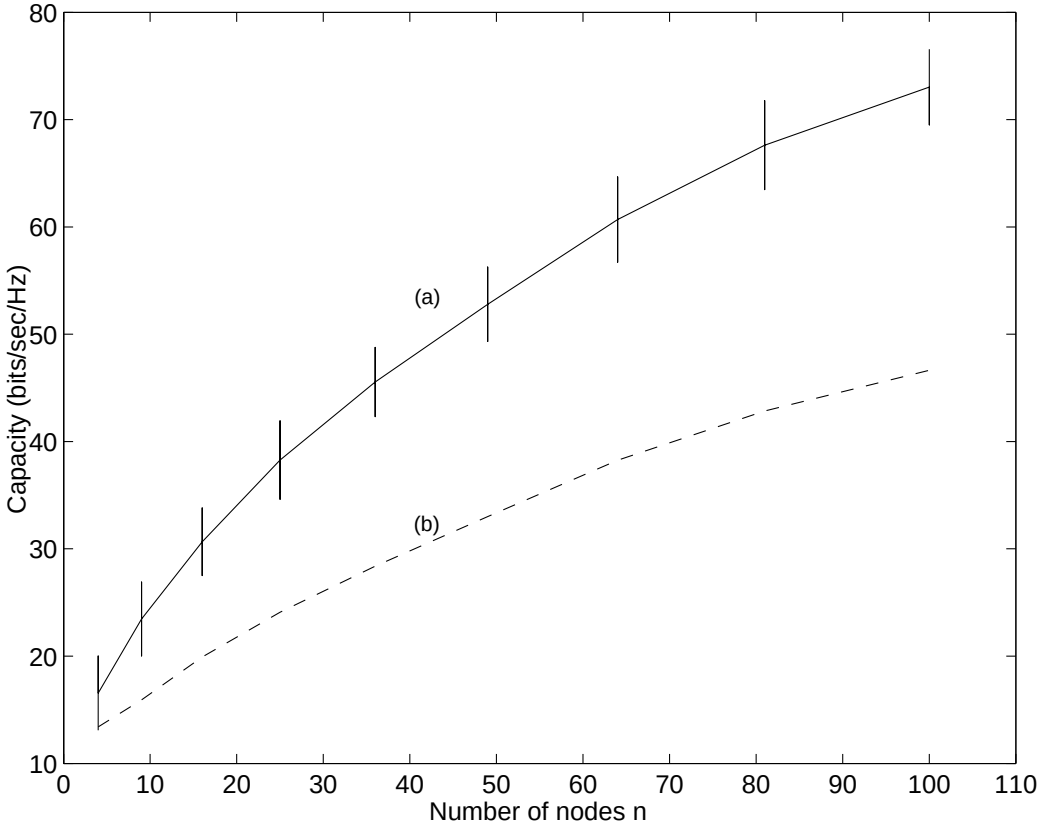

Figure 3.1 *Full traffic matrix with elements Poisson distributed with mean 1.* $\gamma^{norm} = 20$*dB. Channel model considers path-loss and shadowing. (a) Average FD upper bound and its standard deviation. (b) Average TD lower bound.*

TD lower bounds are not shown because similar trends shown in Fig. 3.1 can be assumed for each upper bound curve.

A plot of capacity versus normalizing SNR γ^{norm} for different network sizes is shown in Fig. 3.3. We can see from the figure that small networks ($n \leq 9$) have a fairly linear increase in capacity as a function of SNR in dB. The slope of a straight line approximating curve (a) is about 1bits/sec/Hz per 1dB of SNR. If we want to increase the capacity by 10bits/sec/Hz, we will have to increase the SNR by ten-fold. On the other hand, a moderate to large network ($n \geq 36$) display a polynomial curve, which means that as the number of nodes and SNR increase, less relative SNR change is required to produce a large capacity gain. Consequentially, this leads to the observation that as

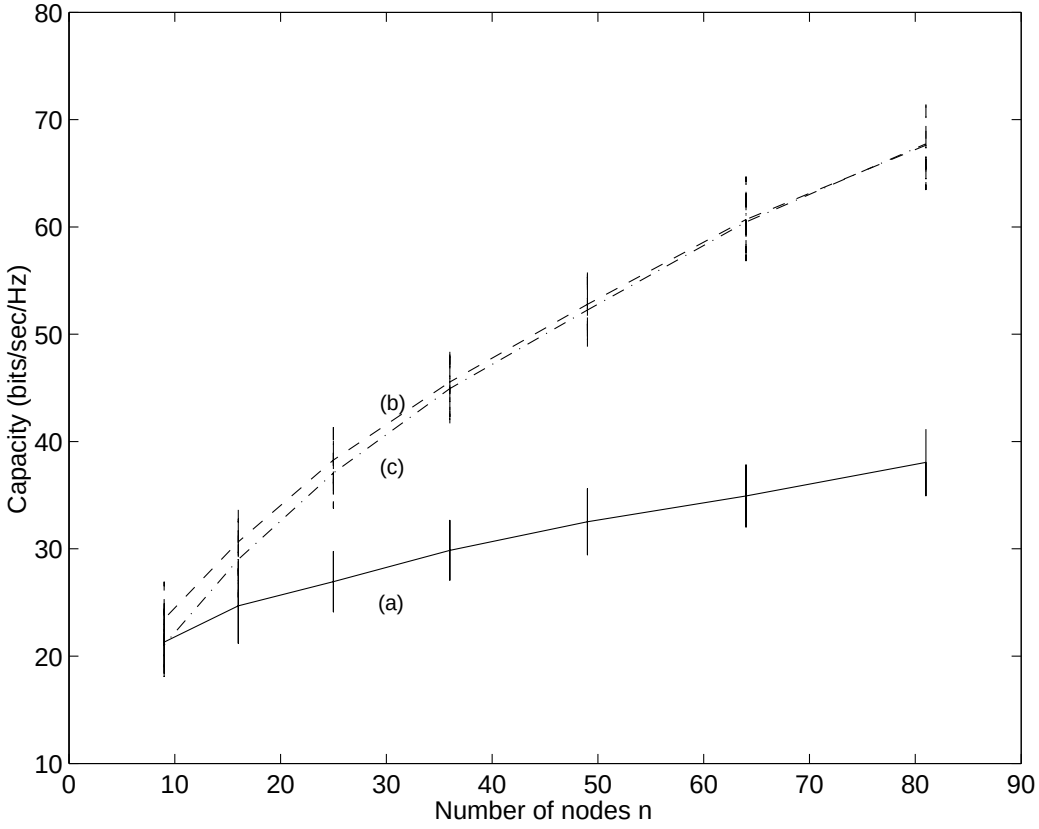

Figure 3.2 *Full traffic matrix with elements Poisson distributed with mean 1.* $\gamma^{norm} = 20dB$*. (a) Average FD upper bound and standard deviation with channel model considering path-loss only. (b) Average FD upper bound and standard deviation with channel model considering path-loss and shadowing. (c) Average FD upper bound and standard deviation with channel model considering path-loss, shadowing and multipaths.*

a network becomes very large, at high SNR, even a small relative node increase can produce a large capacity gain. As an example, in Fig. 3.3, at SNR = 5dB, we have to increase the network size from 9 nodes to 81 nodes in order to double the capacity. However, at SNR = 20dB, we only have to increase the network size from 9 nodes to 36 nodes in order to double the capacity.

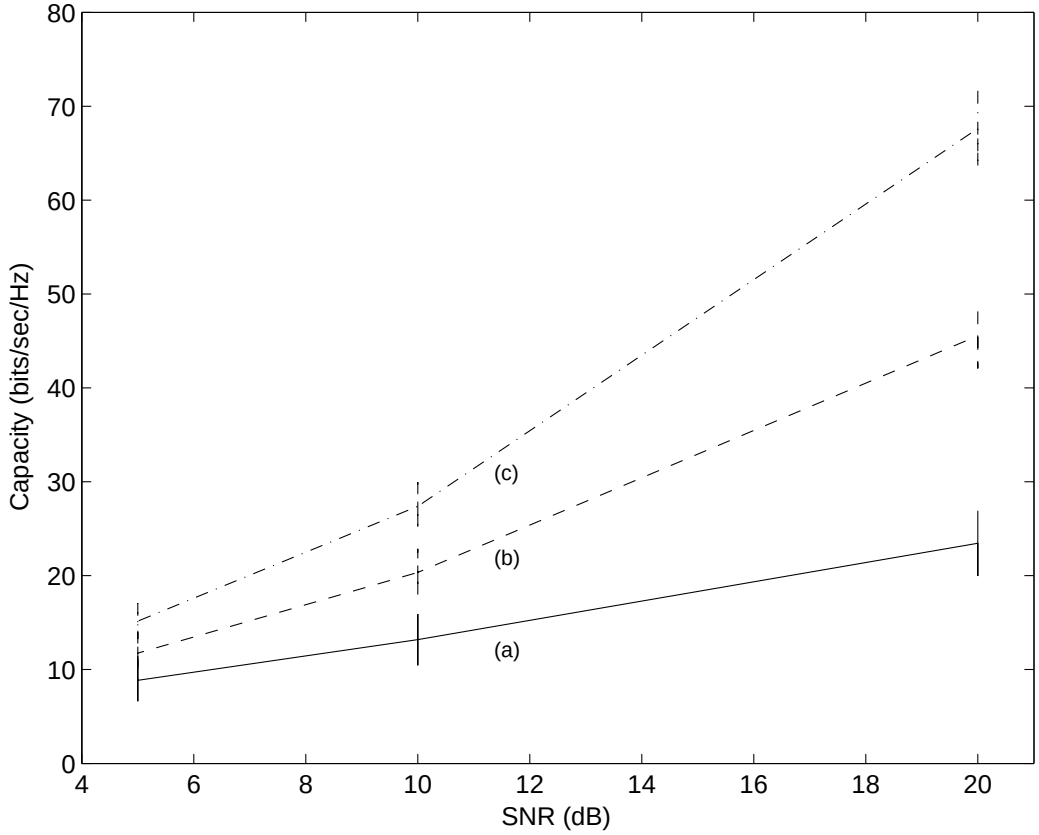

Figure 3.3 *Full traffic matrix with elements Poisson distributed with mean 1. Channel model considers path-loss and shadowing. SNR is the normalizing SNR γ^{norm}. (a) Average FD upper bound and standard deviation for $n = 9$. (b) Average FD upper bound and standard deviation for $n = 36$. (c) Average FD upper bound and standard deviation for $n = 81$.*

3.5.2 Ring Traffic Pattern

In this subsection, we evaluate networks of various sizes and traffic matrices with a ring structure, where all the non-zero elements are 1. This means that each node transmits to only one node and receives from only one node, but not the same node. A

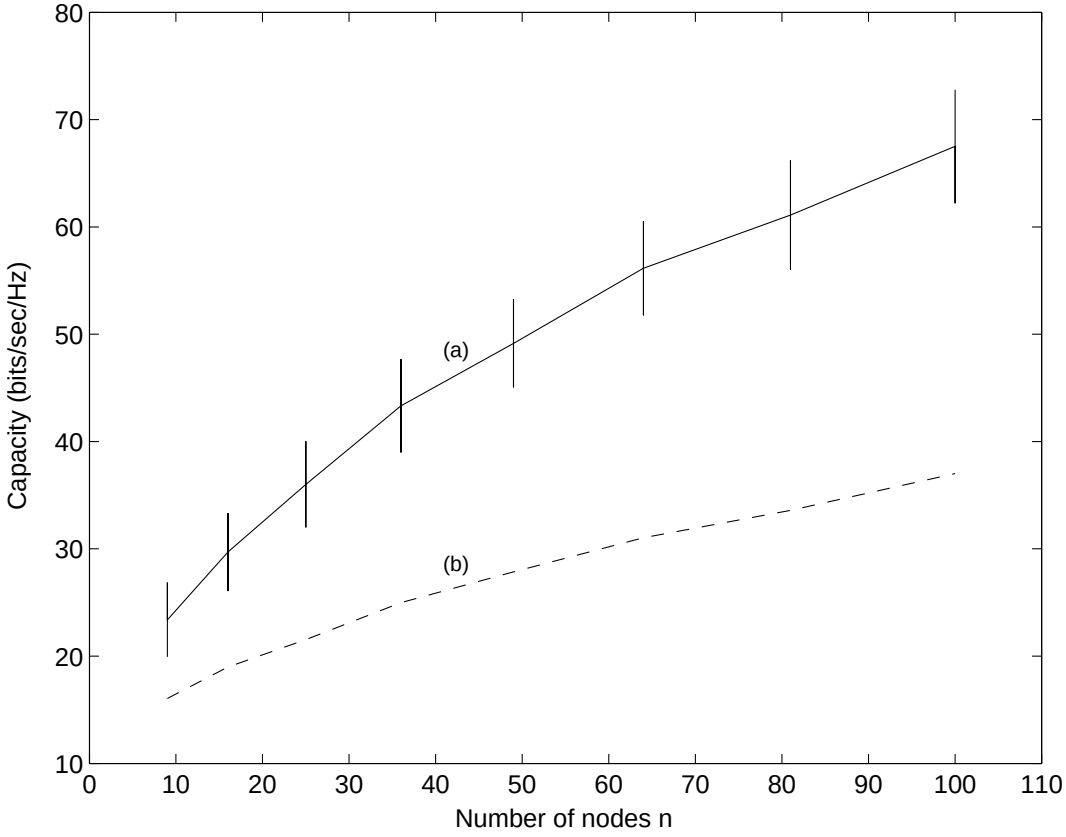

Figure 3.4 *Ring traffic matrix. $\gamma^{norm} = 20dB$. Channel model considers path-loss and shadowing. (a) Average FD upper bound and standard deviation. (b) Average TD lower bound.*

ring traffic pattern for a 4-node network may look like:

$$\boldsymbol{T} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

This type of traffic pattern characterizes a network with very light traffic load where each node only communicates with few other nodes. In Fig. 3.4, we show the FD upper bound and TD lower bound for $\gamma^{norm} = 20$dB and different network sizes. The FD

upper bound is very close to the FD upper bound of Fig. 3.1, which is not surprising because in each case the respective traffic matrix is scaled so at least one node in the network is at 100% utilization. However, the FD upper bound in Fig. 3.4 is slightly lower than its counterpart in Fig. 3.1. This is because a full traffic matrix makes more efficient use of the network resources than a sparse matrix would. In other words, on average, there are more fully utilized nodes under heavy traffic than light traffic. Another thing we can see is that, for a large network, the TD lower bound in Fig. 3.4 is about 1/2 of the FD upper bound, as opposed to 2/3 shown in Fig. 3.1. This is the result of the shape of the ring traffic pattern. Since the traffic matrix is clearly not symmetric about its diagonal, we would not expect a node to spend the same amount of time in transmission and reception; therefore, the amount of time the FD algorithm predicts will take to remove the traffic $\tilde{\boldsymbol{B}}$ is somewhere between 1 to 2 seconds, whereas it will very likely take 3 seconds for TD to remove the same traffic.

In Fig. 3.5, we show the resulting FD upper bounds for different channel models, which exhibit similar behaviors as those shown in Fig. 3.2. That is, shadowing improves the capacity, although not as much as in the case with full traffic matrix, but it is still significant ($\sim 64\%$ gain); and the presence of multipaths does not significantly impact the capacity of the network when shadowing is present.

Finally, we look at how different normalizing SNR affect the capacity for a select few network sizes. The results are shown in Fig. 3.6. Similar to the results for a full traffic matrix, we can see from the figure that small networks ($n \leq 9$) have a fairly linear increase in capacity as a function of SNR in dB, with a slope of approximately 1bits/sec/Hz per 1dB of SNR. On the other hand, a moderate to large network ($n \geq 36$) display a polynomial curve, which means that as the number of nodes and SNR increases, less relative SNR change is required to produce a large capacity gain. Consequentially, this leads to the observation that as a network becomes very large, at high SNR, even a small relative node increase can produce a large capacity gain.

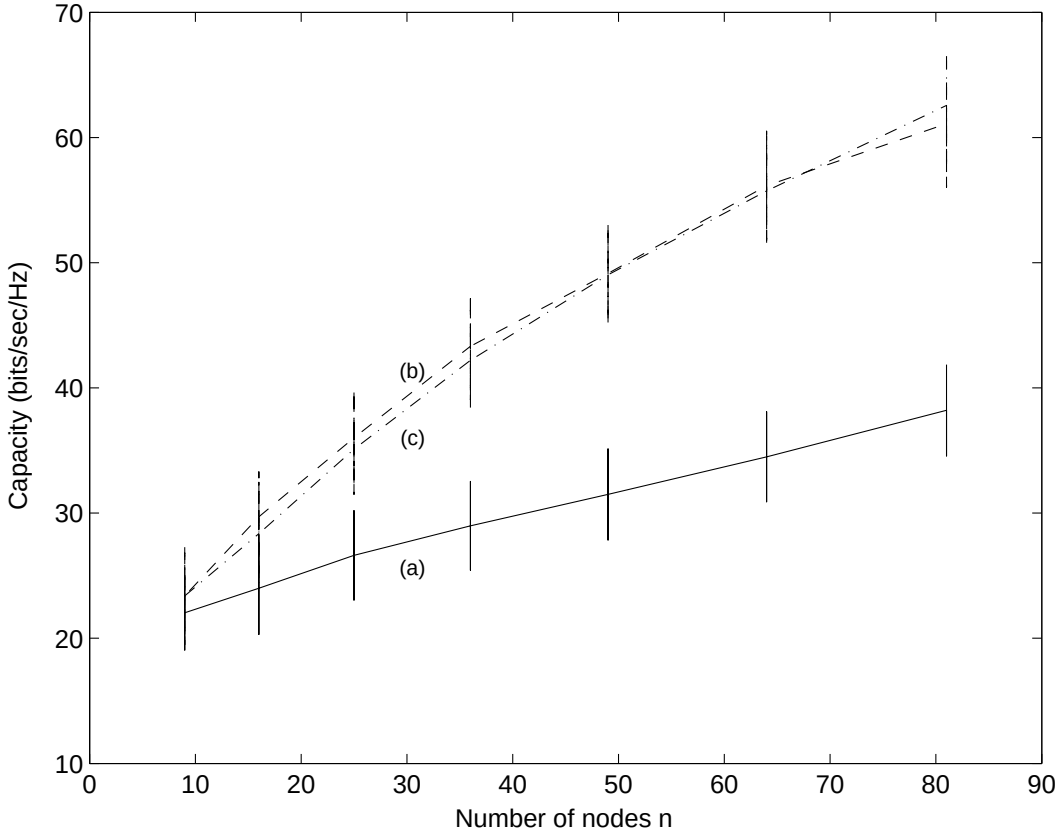

Figure 3.5 *Ring traffic matrix. $\gamma^{norm} = 20dB$. (a) Average FD upper bound and its standard deviation with channel model considering path-loss only. (b) Average FD upper bound and its standard deviation with channel model considering path-loss and shadowing. (c) Average FD upper bound and its standard deviation with channel model considering path-loss, shadowing and multipaths.*

3.5.3 Skewed Traffic Pattern

In this subsection, we evaluate the capacity of networks with traffic matrix in a skewed pattern. A skewed traffic matrix is one where all the traffic originate from and destine to only a select few nodes. It can be pictured in a tree form where there are only a few root nodes, and all traffic flow from the root nodes to the leaf nodes and vice versa. Let's call the root nodes the base-stations and all the remaining nodes the mobiles. An example of a network with 9 nodes composed of 3 base-stations and 6 mobiles is shown in Fig. 3.7. The base-stations are represented by solid circles, and the mobiles

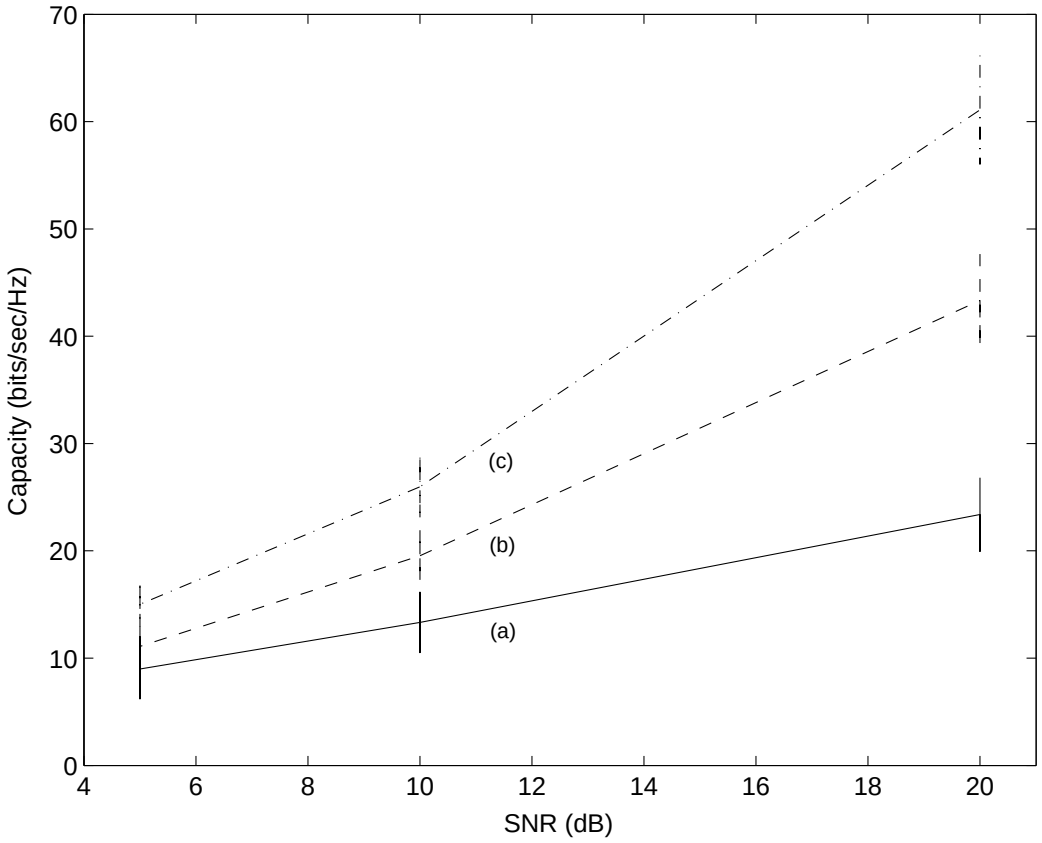

Figure 3.6 *Ring traffic matrix. Channel considers path-loss and shadowing. SNR is the normalizing SNR γ^{norm}. (a) Average FD upper bound and standard deviation for $n = 9$. (b) Average FD upper bound and standard deviation for $n = 36$. (c) Average FD upper bound and standard deviation for $n = 81$.*

and represented by white circles. Solid lines represent wired links, which we assume to have infinite capacity, and the dotted lines represent wireless links, which have finite capacity. The corresponding traffic matrix would have only 3 rows and columns with non-zero elements. We assume base-stations do not have traffic for each other, so, in general, a skewed traffic matrix would have $T(i, j) = 1$ if one of the indices is a base-station and the other is a mobile; otherwise, $T(i, j) = 0$.

A skewed traffic matrix can represent many different types of networks that are composed of different types of nodes. In one scenario, this type of traffic matrix can characterize a wireless metropolitan area network (WMAN) [37], where there are 3

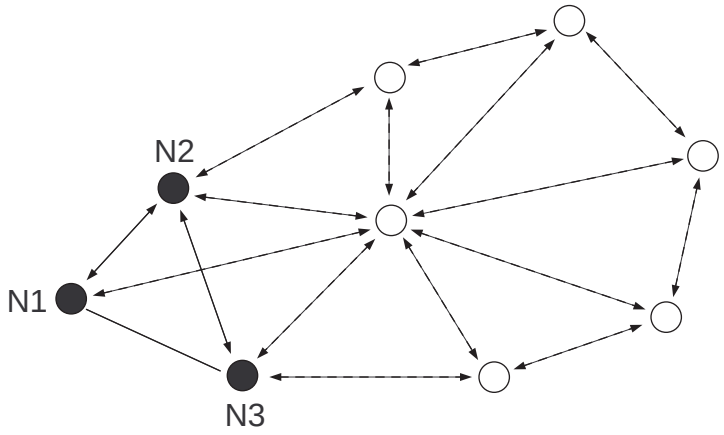

Figure 3.7 *A 9-node network composed of 3 base-stations (solid circles) and 6 mobiles (white circles). Solid lines represent wired links, which are assumed to have infinite capacity. Dotted lines represent wireless links, which have finite capacity. Traffic matrix are skewed, meaning the demands are only between base-stations and mobiles.*

base-stations and various number of access nodes and mobiles. If we assume that the communication channels between mobiles and access nodes and the communication channels between access nodes and base-stations are separate and non-interfering, then the skewed traffic matrix specifies the traffic between access nodes and base-stations. In another scenario, this type of traffic matrix can represent a multihop cellular network [38, 39], where mobiles that are far from a base-station can use mobiles that are closer to relay their traffic. The node distribution shown in Fig. 3.7 more realistically portrait one quadrant of a cellular network that is divided into four sectors, where all sector implements frequency reuse, so a node from one sector can only route traffic from another node in the same sector.

In our simulation, we assigned the 3 nodes closest to an arbitrary corner of the distribution area as the base-stations, and the rest as mobiles. The performance of networks with skewed traffic matrices are shown in Fig. 3.8. First of all, it can be seen from the figure that the capacity of the network decreases as a function of the total number of nodes. This is because, for small number of nodes, the bottlenecks of the network are

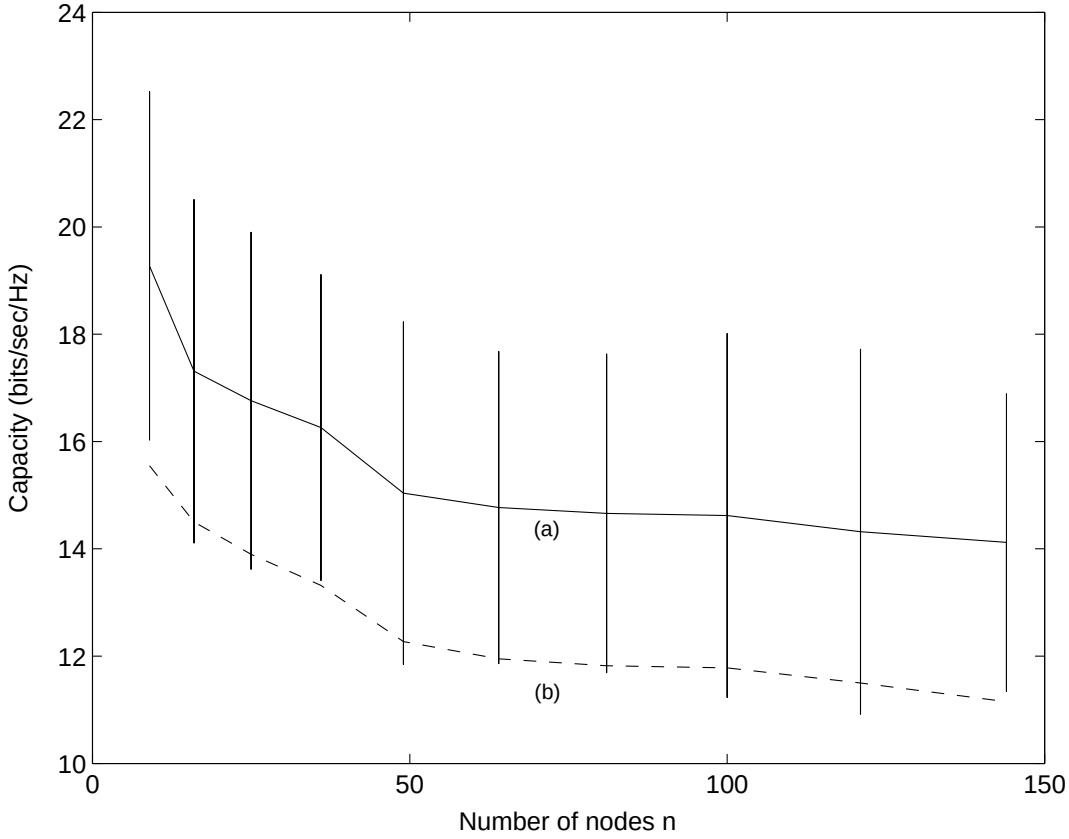

Figure 3.8 *Skewed traffic matrix. $\gamma^{norm} = 20dB$. Channel model considers path-loss and shadowing. (a) Average FD upper bound and standard deviation. (b) Average TD lower bound.*

often the base-stations. However, as the network grows, nodes close to the base-stations usually also become bottlenecks, and the average distances between the base-stations and the mobiles who are bottlenecks also increase. This factor along with a high path-loss attenuation factor of $\alpha = 4$ caused the total capacity to decrease with n. Secondly, the TD lower bound is around 80% of the FD upper bound, which is higher than the respective results for a full or ring traffic matrix. This is largely a result of the highly directional flows to and from one corner of the network caused by the skewed traffic pattern.

We also plotted the capacity versus the normalizing SNR, for a selective few

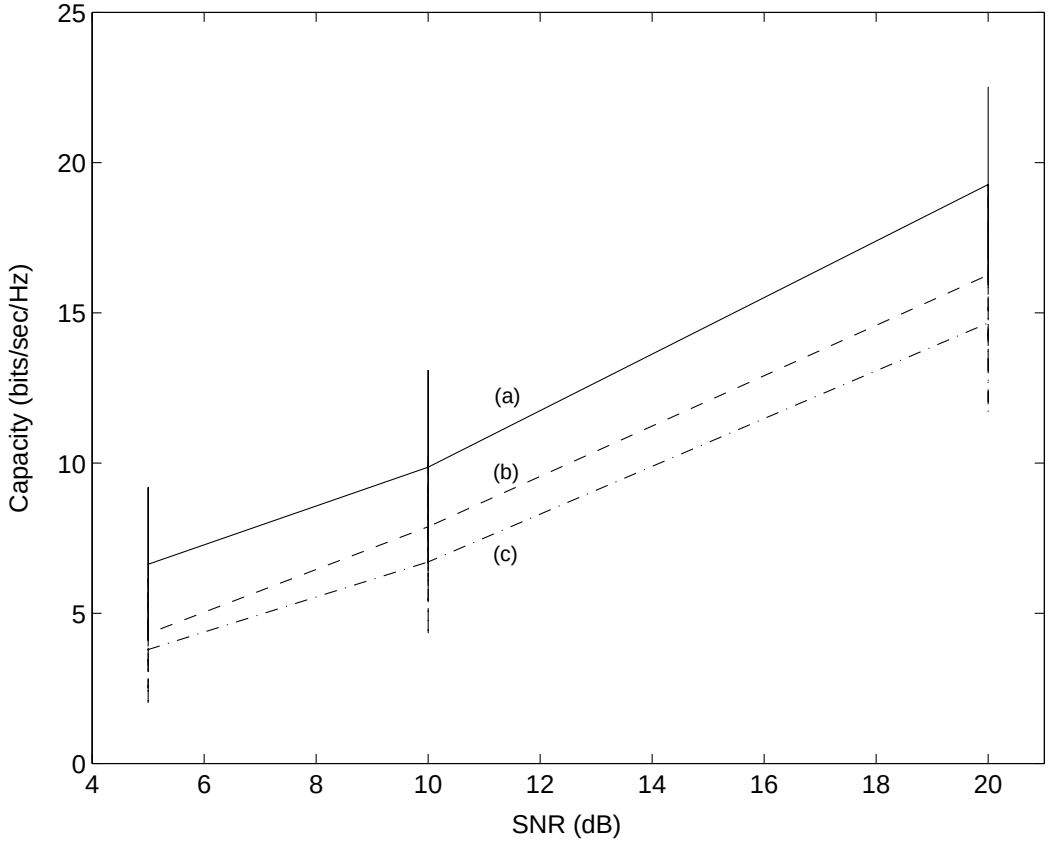

Figure 3.9 *Skewed traffic matrix. Channel model considers path-loss and shadowing. SNR is the normalizing SNR* γ^{norm}*. (a) Average FD upper bound and standard deviation for* $n = 9$*. (b) Average FD upper bound and standard deviation for* $n = 36$*. (c) Average FD upper bound and standard deviation for* $n = 81$*.*

network sizes. The results are shown in Fig. 3.9. As opposed to the results (Fig. 3.3 and Fig. 3.6) shown for the other traffic patterns, where the capacity increases more quickly for larger n, the results for the skewed traffic matrix show the complete opposite. Fig. 3.9 shows that not only is the capacity a decreasing function of n, the capacity function maintains approximately the same shape regardless of the size of the network. In other words, the major difference between the capacity curve for a 9-node network and a 81-node network is some constant offset. This means that once we have a full curve for a particular network size up to some normalizing SNR, we can approximate the performance of any other network size up to the same SNR, as long as we know the

Table 3.1 *Average simulation time (sec) of FD algorithm for* $\gamma^{norm} = 20dB$

$n =$	9	16	25	36	49	64	81
Full traffic	0.12	0.88	4.4	18	65	214	650
Ring traffic	0.06	0.40	1.5	5	11	26	48
Skewed traffic	0.06	0.26	0.8	2	3	10	17

offset value at any particular SNR.

3.6 Computational Issues

The simulations were performed on a computer with Intel Core™2 Duo processors at 2.00 GHz and 2 gigabyte of RAM. The software we used was MATLAB. For evaluation of simulation speed, two sets of run times are saved. One is the run time in the calculation of the FD upper bound, and the other is the run time in the decomposition of the flow into TD schedules. Recall that when we set up the FD algorithm for our problem in Section 3.3 that the number of iterations to be run is dependent on ϵ', which controls the threshold of directional derivatives above which the algorithm terminates. In our simulations, $\epsilon' = -0.1$. The average run times for the FD upper bounds is shown in Tables 3.1 for $\gamma^{norm} = 20$dB and various traffic patterns.

We can see that the run time for the FD algorithm under a full traffic matrix increases very fast as the size of the network grows. This is due to the fact that the number of shortest-path calculations grow as n^2. On the other hand, the run times for both ring traffic and skewed traffic increase much more slowly as a function of the network size. This is because the number of shortest-path calculations only grows linearly with n. The run times in skewed traffic is smaller than that for a ring traffic is because the FD algorithm converges much faster for a skewed traffic due to the directionality of the flows, in which case certain nodes, either the base-stations themselves or some mobiles close to the base-stations, unavoidably become congested quickly.

Table 3.2 *Average simulation time (sec) of TD decomposition*

$n =$	9	16	25	36	49	64	81
Avg. time =	0.01	0.05	0.15	0.4	1	2	5

The run time of the FD algorithm can be decreased in many ways. The simplest way is to increase the processing power of the computers. Another is to streamline the algorithm to take advantage of certain aspects of computer processing (e.g., parallel processing, using C++/MATLAB mixture of code). Also, there are a few ways to increase the convergence rate of the FD algorithm if one is only interested in the ball-park value of the upper bound. One way is to decrease the directional derivative threshold ϵ', which will decrease the number of iterations of the FD algorithm before converging. Another is to delete certain links of the network that fall below some capacity threshold, thus reducing the total number of alternate paths to route the traffic.

The run times to calculate the TD lower bound from the flows of an n-node network is shown in Table 3.2. In obtaining the values for Table 3.2, We did not specify the type of traffic patterns or the value of the normalizing SNR used because the TD decomposition depends primarily on the size of the flow matrix. During each iteration of the TD decomposition, at least one of the elements of the flow matrix is removed; therefore, there are a maximum of $n(n-1)$ iterations of the TD decomposition of an n-node network. The run times for TD decompositions may increase as n^2, but it is much less compared to the run times of the FD algorithm for all three types of traffic patterns.

3.7 Conclusions

In this chapter, we introduced and described the FD method. Using this method, we found an upper bound on the capacity of wireless ad hoc networks when there is no interference. The FD upper bound is not necessarily the tightest upper bound, but

it does provide a yardstick against which the throughput of a conventional hop-by-hop ad hoc network may be gauged. In fact, the FD upper bound is unachievable with high probability as the number of nodes in the network increases, but we proved that, under special cases, 2/3 of it is achievable, and under the worst case, 1/3 of it is achievable. The FD method also allowed us to examine the behaviors of networks under different channel conditions and with different traffic patterns. We saw that if the channel conditions are known, then shadow and multipath fading actually improves the overall throughput because the network can take advantage of the links that are bolstered by the fading, while avoiding those that are degraded. For networks with full or ring traffic patterns, it is not surprising to find the network capacity to increase as a function of n. Under the same traffic patterns, we also observed that, for small networks ($n = 9$), the capacity increases approximately linearly with γ^{norm}, and for medium to large networks ($n > 36$), the capacity increases polynomially with γ^{norm}. One surprising observation is that, under skewed traffic pattern, the total capacity actually decreases with n, and that the capacity-γ^{norm} curves for different network sizes only differ by some constant offsets. These phenomena are results of the way base-stations are distributed in one corner of the coverage area and bottlenecks being created along the path between sources and destinations as the network size grows. Finally, we examined the computational complexity of the FD algorithm and found that it scales better than the methods in [25, 26].

Chapter 3, in part, is a reprint of the material in the following papers: M.Y. Tan and A. Acampora, "Capacity Estimation of Peer-to-Peer Networks Based on a Flow Deviation Approach," *IEEE International Symposium on Personal, Indoor, Mobile Radio Communications* (PIMRC), Athens, Sept. 2007; paper to be submitted to *IEEE Transactions on Wireless Communications*. The dissertation author was the primary investigator and author of this paper, and the co-author listed in these publications directed and supervised the research which forms the basis of this dissertation.

CHAPTER 4

Carrier Sense Multiple Access with Collision Avoidance

4.1 Introduction

In Chapter 3, we found an upper bound on the capacity of wireless ad hoc networks in the absence of co-channel interference. In this chapter, we consider an existing MAC protocol and compare its performance, in the presence of interference, against the upper bound. The MAC protocol we will evaluate is based on CSMA/CA.

The chapter is organized as follows: In Section 4.2, we give an overview of the CSMA/CA protocol. In Section 4.3, we describe the model that is used to simulate the protocol. In Section 4.4, we show how different queueing disciplines affect the performance of CSMA/CA. In Section 4.5, we present and analyze some simulation results. We conclude in Section 4.6.

4.2 Background

Under CSMA/CA [40, 41, 42, 43], a source node that wish to transmit data first sense the channel. If the channel is busy, then the node defers transmission. If the channel is free, then it sends a short *Request To Send (RTS)* packet. After receiving the *RTS* packet and upon determining the channel is free, the destination node will

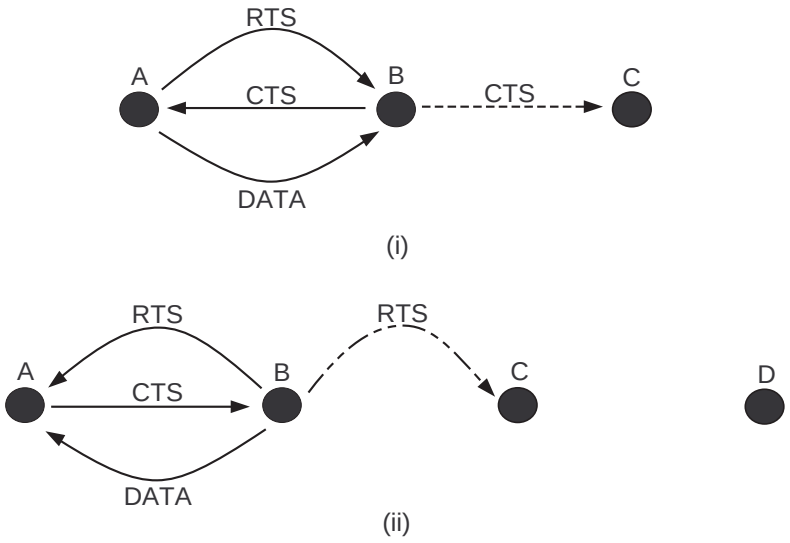

Figure 4.1 *The hidden and exposed node problems. (i) A wishes to transmit to B, and C, hidden from A's communication, is notified of this intention upon hearing the* CTS *packet so it will defer until after B receives the* DATA *packet. (ii) B wishes to transmit to A, and C, exposed to B's communication, is notified of this intention by hearing the* RTS *packet but it will only have defer until after B receives the* CTS *packet. If B and C coordinate their control packets, C can also transmit to D, thus resulting in higher utilization of the channel.*

respond with a short *Clear To Send (CTS)* packet, giving the source node permission to send data. The *CTS* packet also acts as a mechanism to silence the hidden nodes, which are nodes that are within the destination node's communication range but outside the source node's communication range. A node hearing the *CTS* packet will refrain from transmitting for the entire duration of the data packet so it will not cause packet collision at the destination node. On the other hand, a node hearing only the *RTS* packet will simply defer long enough for the source node to receive the *CTS*, after which it may begin its own transmission, as long as its receiver is out of the range of the source node, thus effectively solving the exposed terminal problem (Fig. 4.1).

CSMA/CA is an improvement over CSMA, but it does not completely eliminate the hidden node problem. First of all, *CTS* packets may still suffer collision. Secondly,

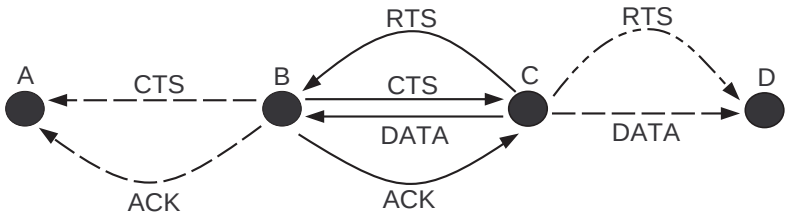

Figure 4.2 *The* RTS-CTS-DATA-ACK *handshake. C wishes to transmit to B, and both A and D must defer until after the* ACK *packets is received. D cannot transmit early because it will interfere with the reception of* ACK *at C.*

interference range is different from *communication range*, meaning a node that does not hear the *CTS* packet can still potentially cause interference at the receiver. For example, in Fig. 4.1(ii), if the channel matrix is asymmetric, then C may be outside B's communication range but B is inside C's communication range, and C can still cause collision at B. However, this fact is usually overlooked in evaluating protocols based on CSMA/CA as it leads to simpler graph models [44, 45].

The *RTS/CTS* handshake is also coupled with a back-off mechanism that ensures nodes do not transmit immediately after the channel becomes available, but rather wait a random back-off time, in order to reduce the probability of collision. In some variations [40, 43], an *Acknowledgment (ACK)* packet is returned by the destination node upon successful reception of the data packet. Under these settings, all nodes hearing the *RTS* and/or *CTS* will refrain from transmitting until the complete data packet and ACK packet have been received. An example of the *RTS-CTS-DATA-ACK* handshake is shown in Fig. 4.2.

4.3 Simulation Model

The throughput of the CSMA/CA medium access control mechanism in IEEE 802.11 has been studied extensively [46, 47, 48, 49]. We consider the optimistic perfor-

mance of a simple protocol based on CSMA/CA by assuming that *RTS* and *CTS* packets use negligible time to transmit, and no *ACK* is sent so the channel will be better utilized. We also assume that concurrent transmissions are possible because not all nodes will hear the *RTS-CTS* exchange, and that each transmitter knows the total amount of interference so it can adjust its transmission rate accordingly to meet its receiver's SINR requirement.

We first define the following terms. $\boldsymbol{t}$ and $\boldsymbol{r}$ are vectors containing the set of active transmitters and receivers, respectively. $\bar{\boldsymbol{t}}$ and $\bar{\boldsymbol{r}}$ are vectors containing the set of nodes that cannot be transmitters and receivers, respectively, as a result of overhearing the *RTS-CTS* exchange.

A node N_t not in the sets $\boldsymbol{t}$, $\bar{\boldsymbol{t}}$ and $\boldsymbol{r}$ looks at its queue for the next packet it must transmit. If FIFO (first-in first-out) queueing is required, then the node will attempt to transmit the first packet in its queue; but if SIRO (served in random order) queueing is used, then the node can transmit a packet in any queue position. After determining which packet to transmit and its final destination, the node will look up its next hop receiving node N_r in the routing table, which is calculated using a shortest-path algorithm with inverse capacity as link cost. For example, if a packet is to be transmitted from source N_1 to destination N_2 along shortest-path $N_1 \longrightarrow N_3 \longrightarrow N_2$, then while the packet is in N_1's queue, N_1 will attempt to transmit it to N_3. If the next hop receiving node is not in the sets $\boldsymbol{r}$, $\bar{\boldsymbol{r}}$ and $\boldsymbol{t}$, then both N_t and N_r are added to $\boldsymbol{t}$ and $\boldsymbol{r}$, respectively, which is an indication that N_t did not overhear a *CTS* while N_r did not overhear a *RTS*; and all nodes whose SNRs from N_t are above some SNR threshold γ_t are added to $\bar{\boldsymbol{r}}$, and all nodes whose SNRs at N_r are above γ_t are added to $\bar{\boldsymbol{t}}$ (i.e., nodes that only overhear *RTS* can still transmit but not receive, those that only overhear *CTS* can still receive but not transmit, and those that overhear both must remain silent). Otherwise, N_t is added to $\bar{\boldsymbol{t}}$ [1] because it has no available receiver to which to transmit. This process is

[1] In the case of SIRO queueing, all buffered packets are searched until one, if any, whose N_r not in the sets $\boldsymbol{r}$, $\bar{\boldsymbol{r}}$, and $\boldsymbol{t}$, is found

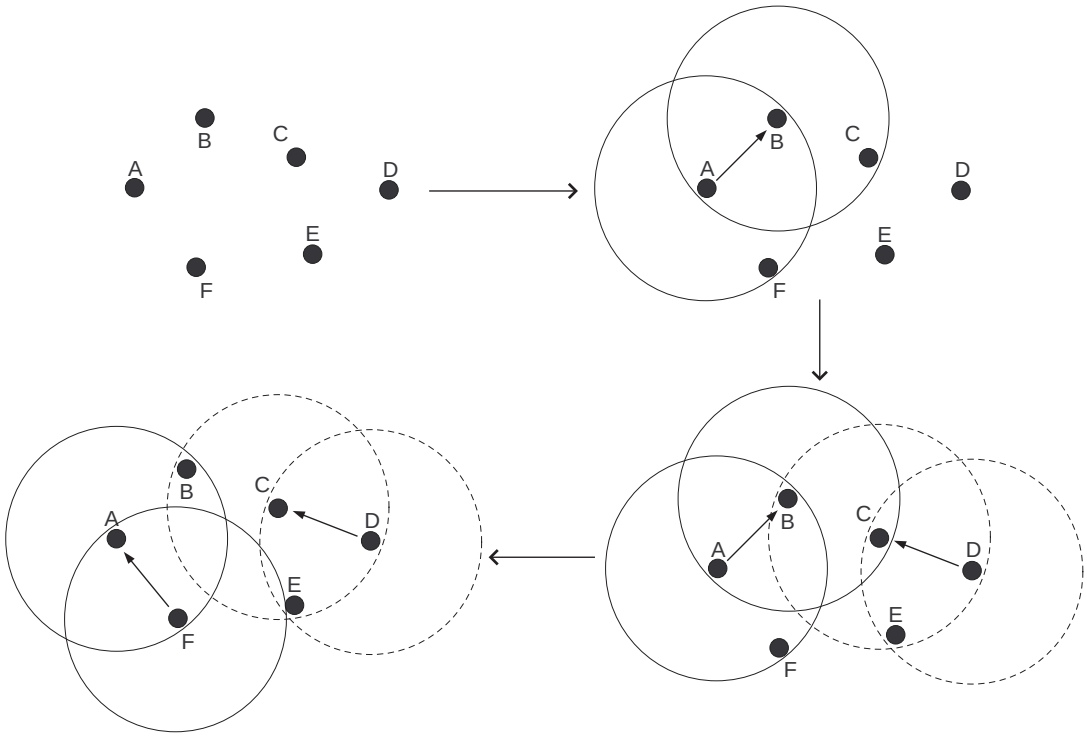

Figure 4.3 *CSMA/CA simulation process.*

repeated until all nodes are searched. At this point, we have a set of transmitters and its corresponding receivers that will communicate concurrently, and we can calculate the precise SINR between each transmitter and receiver. Because the capacities will differ among the transmitter-receiver pairs, one node will complete transmitting its packet before others. Once that node finishes transmitting its packet, the packet is removed from the transmitter's queue and inserted into the receiver's queue. The transmitter-receiver pair is also removed from $\boldsymbol{t}$ and $\boldsymbol{r}$, and the process begins anew at the search of possible additional transmitter-receiver pairs for $\boldsymbol{t}$ and $\boldsymbol{r}$, and the SINRs are updated accordingly.

An example of this process for a 6-node network is shown in Fig. 4.3. In this network, A and B have packets for each other, C and D have packets for each other, E and F have packets for each other, and F and A have packets for each other. Initially

(first step), the network is idle. Next (second step), A captures the channel first and begins transmitting to B. C overhears B's *CTS*, so it must refrain from transmitting, and F overhears A's *RTS*, so it cannot receive. The only options left are for D to transmit to C and for F to transmit to E. In the third step, D captures the channel as well and begins transmitting to C. E overhears the *RTS* and *CTS* between C and D, so it must remain silent until their transmissions are finished. F must also remain silent as well because all of its receivers are unavailable. In the fourth step, A completes its transmission to B, and F captures the channel and begins transmitting to A.

The SNR threshold γ_t is the threshold below which control packets can no longer be decoded correctly, causing neighboring nodes to fail to defer communication. It controls the communication range of a transmitter and indirectly affects the number of concurrent transmissions in a network of fixed size. A very small γ_t indicates a very large communication range, so more nodes will overhear an *RTS*-*CTS* exchange, resulting in very few concurrent transmissions. On the other hand, a very large γ_t indicates a very small communication range, resulting in many concurrent transmissions. It is ideal to operate at some γ_t such that spatial reuse is optimal [50, 51]. Fig. 4.4 shows effects of different values of γ_t.

4.4 Queueing Discipline

According to [52], *queuing discipline* represents the way the queue is organized; that is, it dictates how packets are inserted and removed from the queue. The commonly used queue disciplines are FIFO, LIFO (last-in first-out), SIRO, and priority queues, where packets are served in order of importance on the basis of their performance requirements. FIFO queueing are the most commonly considered discipline, but we will see next that SIRO queueing is more suitable for wireless ad hoc network, as it leads to better spatial reuse.

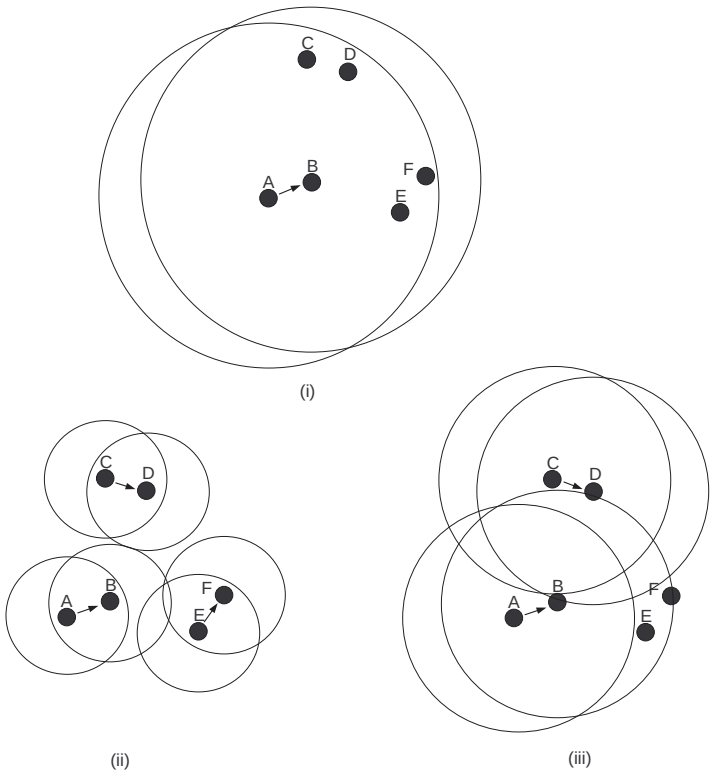

Figure 4.4 *The effect of various values of γ_t on spatial reuse. In all scenarios, A wishes to transmit to B, C wishes to transmit to D, and E wishes to transmit to F. (i) Very small γ_t translates to large communication range, resulting in transmission A $\longrightarrow$ B only. There are no interference at B but also no spatial reuse. (ii) Very large γ_t translates to small communication range, resulting in concurrent transmissions A $\longrightarrow$ B, C $\longrightarrow$ D, and E $\longrightarrow$ F. Each receiver experiences interference from two transmitters, but spatial reuse is maximized. (ii) A value of γ_t that optimizes spatial reuse, where only two concurrent transmissions are possible.*

Consider a four node network shown in Fig. 4.5, where nodes can only communicate with their closest neighbors. Suppose B wishes to transmit to A, and C wishes to transmit to B and D. In Fig. 4.5(i), B captures the channel first and begins transmission to A. C, which uses FIFO queueing, must remain silent until B finishes transmitting. On the other hand, in Fig. 4.5(ii), C, which uses SIRO queueing, can elect to transmit the packet destined to D first while B is busy transmitting to A. In this case, the spatial reuse is increased, and the channel is better utilized.

Figure 4.5 *Spatial reuse under FIFO and SIRO queueing. (i) B is transmitting to A, so C must wait until B is finished. (ii) B is transmitting to A, but C can elect to transmit to D at the same time.*

4.5 Simulation Results and Analysis

To find the average capacity of the network, packets are periodically inserted into the queues of random source nodes and forwarded towards random destination nodes. The number of packets inserted each period is increased until the queue in at least one node is slowly increasing unbounded. We equate the total throughput of the network as its capacity.

The throughput of a network depends on the SNR threshold γ_t, which controls the communication range. Smaller γ_t leads to larger communication range, which results in less interference and spatial reuse, and larger γ_t leads to smaller communication range, which results in more interference but also more spatial reuse. We expect the throughput to be a concave function of γ_t; therefore, given a network size, we look for the optimal throughput by slowly increasing γ_t from 0.

4.5.1 Full traffic pattern

Fig. 4.6 shows the performance of CSMA/CA under heavy traffic as a function of network size. The top plot considers a normalizing SNR $\gamma^{norm} = 20$dB, and it shows that even under favorable conditions (i.e., *RTS-CTS* handshake requires zero time, no *ACK*, perfect and continuous transmission rate adjustments), CSMA/CA falls a magnitude short of the FD and TD bounds. It should be noted that because we used a moderate SNR value, the benefits of spatial reuse over the detrimental effects of interference is only beginning to appear at 81 nodes. In terms of numbers, normalized by the FD upper bound, at $n = 9$, CSMA/CA with SIRO achieves 27%, and CSMA/CA with FIFO achieves 26%; and at $n = 81$, CSMA/CA with SIRO achieves 8%, and CSMA/CA with FIFO achieves 4%.

In the bottom plot, which considers γ^{norm} = 5dB, the benefits of spatial reuse becomes much more pronounced beyond 16 nodes. The difference between the CSMA/CA results and the FD upper bound is also much smaller at the lower SNR range. In terms of numbers, normalized by the FD upper bound, at $n = 9$, CSMA/CA with SIRO achieves 43%, and CSMA/CA with FIFO achieves 40%; and at $n = 81$, CSMA/CA with SIRO achieves 31%, and CSMA/CA with FIFO achieves 15%.

In both cases, when compared to CSMA/CA with FIFO queueing, CSMA/CA with SIRO queueing provides approximately two-fold increase in capacity at 81 nodes. The improvement will only increase with n, as the FIFO result is a decreasing function while the SIRO result is an increasing function due to better spatial reuse.

4.5.2 Skewed traffic pattern

In Fig. 4.7, we consider the performance of CSMA/CA under skewed traffic matrix. In Section 3.5.3, we found that the directionality of the traffic created bottlenecks near the base-station nodes and lead to a decrease of capacity with n. The same

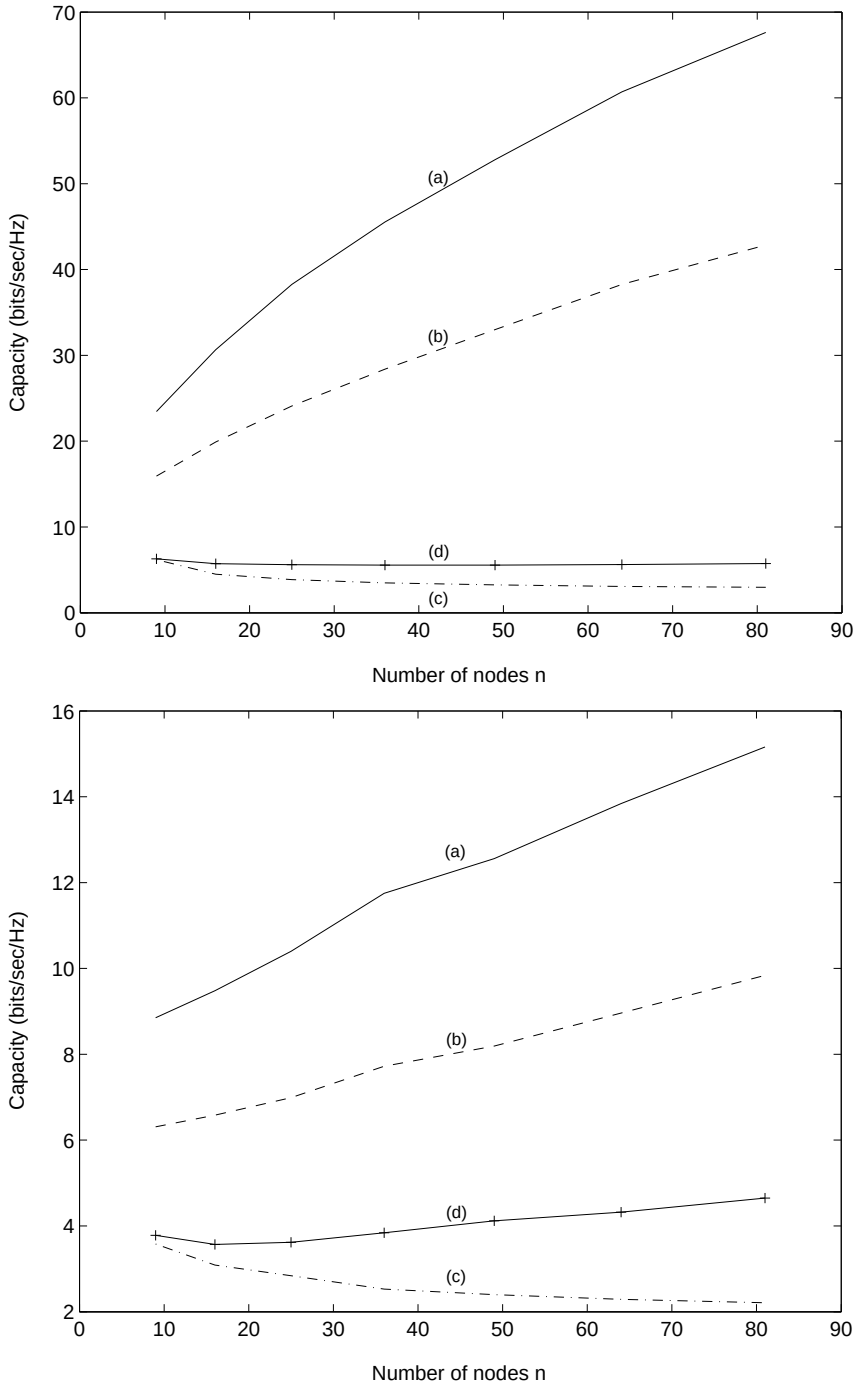

Figure 4.6 *Comparison of theoretical bounds and CSMA/CA throughput. Network considers full traffic matrix with channel model including path-loss and shadowing. Normalizing SNR $\gamma^{norm} = 20$dB for the top plot, 5dB for the bottom plot. (a) Average FD upper bound. (b) Average TD lower bound with no interference. (c) CSMA/CA with FIFO queueing. (d) CSMA/CA with SIRO queueing.*

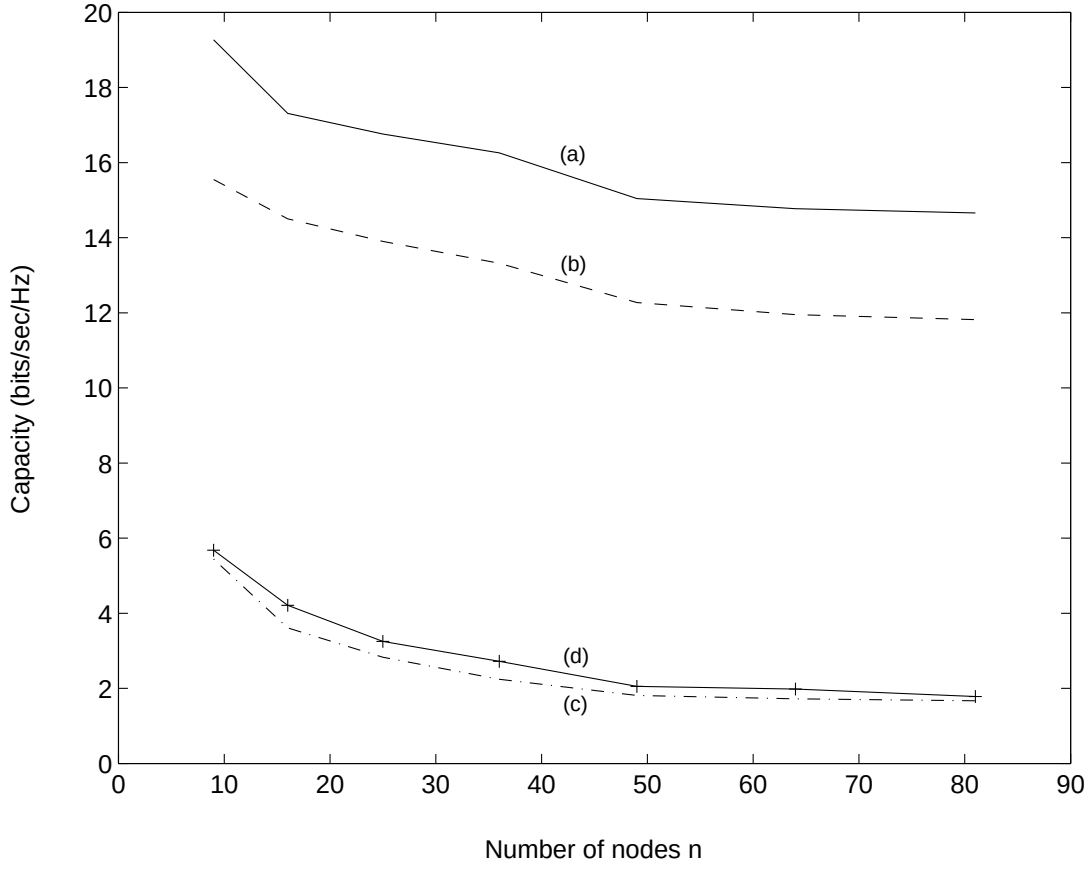

Figure 4.7 *Comparison of theoretical bounds and CSMA/CA throughput. Network considers skewed traffic matrix with channel model including path-loss and shadowing. Normalizing SNR $\gamma^{norm} = 20$dB. (a) Average FD upper bound. (b) Average TD lower bound with no interference. (c) CSMA/CA with FIFO queueing. (d) CSMA/CA with SIRO queueing.*

phenomenon is seen in the CSMA/CA results, where CSMA/CA with SIRO queueing appears to provide only a small improvement ($\sim 10\%$) in capacity when compared to CSMA/CA with FIFO queueing. This is because nodes far from the base-stations are competing to transmit to nodes closer to the base-stations, so transmitting random packets in the queues does not lead to large increase in channel utilization. In terms of numbers, normalized by the FD upper bound, at $n = 9$, CSMA/CA with SIRO achieves 29%, and CSMA/CA with FIFO achieves 28%; and at $n = 81$, CSMA/CA with SIRO achieves 12%, and CSMA/CA with FIFO achieves 11%.

4.6 Conclusions

We have described the CSMA/CA medium access control mechanism and presented its throughput analysis for multihop wireless networks. It is shown that while a very optimistic form of CSMA/CA was analyzed, the gap between the CSMA/CA result and the FD upper bound is still significant, especially if a large network operating at high SNR is considered. This leads to the conclusion that, although the gap between any realistic MAC protocol and the FD upper bound may not be closed completely, as the latter considers no interference, there is still room for improvement in MAC design that will decrease that gap. One of the ways is to use SIRO queueing instead of FIFO queueing, which increases spatial reuse and can provide significant capacity gain as we have shown.

Chapter 4, in part, is a reprint of the material to be submitted to *IEEE Transactions on Wireless Communications*. The dissertation author was the primary investigator and author of this paper, and the co-author listed in these publications directed and supervised the research which forms the basis of this dissertation.

CHAPTER 5

A New Time-Division Scheduling Scheme

5.1 Introduction

In Chapter 4, we compared the performance of a protocol based on CSMA/CA against the FD upper bound and concluded that there is a significant capacity gap that may be decreased by new MAC designs. In this chapter, we derive a time-division scheduling scheme from the FD flow that, even in the presence of interference, performs better than CSMA/CA.

5.2 Time-Division Schedules Derivation

Before we explain how to derive the new time-division schedules, we first refer back to the notations used in the proof of Claim 1 of Section 3.4. First, we introduce the normalized flow matrix $\boldsymbol{B}$ whose elements are defined by $B_{ij} = \hat{f}_{ij}/c_{ij}$, where $\hat{f}_{ij}$ is the element of the global minimum flow $\hat{\boldsymbol{f}}$ that corresponds to the flow on link (i, j) (i.e., $\hat{\boldsymbol{f}}$ is the final flow matrix produced by the FD algorithm). Since the flow elements are in bits and the capacities are in bits/second, then the elements in $\boldsymbol{B}$ have unit in seconds. Next, we rewrite

$$\boldsymbol{B} = \beta \tilde{\boldsymbol{B}} \tag{5.1}$$

where β is the largest row or column sum of $\boldsymbol{B}$. $\tilde{\boldsymbol{B}}$ has the property that there exists some $\hat{j}$ such that either the $\hat{j}$th column sum or $\hat{j}$th row sum or both equal to one. Now we

pad the elements of $\tilde{\boldsymbol{B}}$ so that all of its column and row sums equal to one and call this matrix $\tilde{\boldsymbol{B}}^{pad}$. $\tilde{\boldsymbol{B}}^{pad}$ is a doubly stochastic matrix and can be decomposed into a convex combination of permutation matrices [1].

Next, we seek to find a set of permutation matrices that can be convexly combined into $\tilde{\boldsymbol{B}}^{pad}$. This can be done as follows:

1. Let $m = 1$
2. Let E_m be a perfect matching of $\tilde{\boldsymbol{B}}^{pad}$
3. Define $\boldsymbol{P}_m$ such that $P_{ij}^m = \begin{cases} 1, & \text{if } E_m \text{ has an edge } (i,j) \\ 0, & \text{otherwise} \end{cases}$
4. Let α_m equal the smallest element in $\tilde{\boldsymbol{B}}^{pad}$ that corresponds to the non-zero elements of $\boldsymbol{P}_m$
5. Set $\tilde{\boldsymbol{B}}^{pad} = \tilde{\boldsymbol{B}}^{pad} - \alpha_m \boldsymbol{P}_m$
6. Set $m = m + 1$ and repeat from Step 2 until $\tilde{\boldsymbol{B}}^{pad} = \boldsymbol{0}$

In Step 2, a permutation matrix can be obtained by finding a perfect matching of $\tilde{\boldsymbol{B}}^{pad}$ using the Dulmage-Mendelsohn decomposition [53]. In Step 3 and 4, we weigh the permutation matrix by a factor α_m, which is the amount of time it will take to empty one element of $\tilde{\boldsymbol{B}}^{pad}$. Since we subtract the same value from all rows and columns of $\tilde{\boldsymbol{B}}^{pad}$ in Step 4, the new $\tilde{\boldsymbol{B}}^{pad}$ is another doubly stochastic matrix only weighted by a value less than 1. Therefore, it can further be decomposed into a convex combination of permutation matrices. We continue to empty elements from $\tilde{\boldsymbol{B}}^{pad}$ in Step 5 until all elements are zero, at which time we will have obtained a set of weighted permutation matrices whose weights sum to 1.

Each permutation matrix represents a set of concurrent transmissions, and its

[1] Birkhoff-von Neumann Theorem. See Section 3.4

weight the amount of time the concurrent transmissions are active. The resulting weighted permutation matrix dictates how much time is remove from $\tilde{\boldsymbol{B}}^{pad}$. If we define $\boldsymbol{f}^P = \alpha(\boldsymbol{C} .* \boldsymbol{P})$, then it denotes the corresponding flow removed by using the set of transmissions given by $\boldsymbol{P}$ for α seconds[2]. For example, if we have

$$\alpha = 0.5, \quad \boldsymbol{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \boldsymbol{C} = \begin{pmatrix} 0 & 1.7 \\ 2.3 & 0 \end{pmatrix},$$

then $\alpha\boldsymbol{P}$ implies that links $(1, 2)$ and $(2, 1)$ are both active for 0.5 seconds, and $\boldsymbol{f}^P$ implies that 0.85 bits and 1.15 bits are removed from links $(1, 2)$ and $(2, 1)$, respectively.

The set of weights $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots)$ and the corresponding permutation matrices obtained from the decomposition of $\tilde{\boldsymbol{B}}^{pad}$ indicate very optimistic performance that not only require a node to be able to establish multiple full-duplex connections simultaneously, but also require the channel to be interference-free. In Section 3.4, we analyzed the case where nodes are only allowed to establish single half-duplex connections and were able to decompose each permutation matrices into three or less maximum size rate matrices, each representing a set of concurrent transmissions, to obtained a TD lower bound with no interference. However, these rate matrices may not perform well when interference is considered. Therefore, we seek an alternate strategy to obtain a set of rate matrices from each permutation matrix that perform well under interference.

In decomposing a permutation matrix into a set of rate matrices that observes interference, there are two extremes. On one end, a permutation matrix can be decomposed into n rate matrices each representing a single transmission, which is, by definition, interference-free. On the other end, it can be decomposed into a set of maximum size rate matrices, in which each receiver experiences a large amount of interference. Analogous to CSMA/CA, obtaining a set of rate matrices of single transmission is equivalent to having a very small SNR threshold γ_t, and obtaining a set of rate matrices

[2].* is an element-by-element multiplication operator

of maximal concurrent transmissions is equivalent to having a very large γ_t. Therefore, we expect the optimal performance to occur when there are a moderate number of rate matrices each representing only a few concurrent transmissions. However, given a permutation matrix, in order to find the optimal performance under interference, we will have to consider all combinations of different size rate matrices (i.e., all combinations of singles, doubles, triples, etc.). Instead of doing an exhaustive search for each permutation matrix, we will seek to obtain a "good" suboptimal performance using a straightforward method.

First, we make the following observation. For an n-node network, each weighted permutation matrix $\alpha \boldsymbol{P}$ has n non-zero elements, corresponding to n transmitters transmitting to n receivers for α seconds. $\boldsymbol{f}^P$ denotes the amount of flows removed from the corresponding transmitter-receiver pairs, in the absence of interference. However, if we account for interference, then a node will take longer than α to completely transmit the same amount of data. However, it is still advantageous to allow concurrent transmissions if the overall throughput is greater than the weighted sum of the throughput of the individual transmissions. For example, if $c_{AB} = 1$bit/second and $c_{EF} = 1$bit/second, then if A transmit to B in the first second and E transmit to F in the second second, then the average throughput is 1bit/second. However, if both A and E are concurrently transmitting to their respective receivers and the capacities with interference are $c^I_{AB} = 0.75$bits/second and $c^I_{EF} = 0.75$bits/second, then the throughput is 1.5bits/second, which is higher than if they transmitted individually.

With the above observations, we formulated a straightforward strategy to derive rate matrices that perform well under interference from a weighted permutation matrix, a process of which is similar to that of the CSMA/CA scenario. Let $\boldsymbol{t}_m$ and $\boldsymbol{r}_m$ denote the sets of active transmitters and receivers, respectively, in the mth iteration. Let $\bar{\boldsymbol{t}}_m$ and $\bar{\boldsymbol{r}}_m$ denote the sets of nodes that cannot transmit and receive, respectively. Let c^I_m denote the sum capacity of the links of the pairs in $\boldsymbol{t}_m$ and $\boldsymbol{r}_m$, while taking into account

interference; and let $\tilde{c}_m^I$ denote the sum capacity of the links of the pairs in $\boldsymbol{t}_m$ and $\boldsymbol{r}_m$ and another pair not in the sets, while taking into account interference. In other words, c_m^I is the sum capacity of the current set of concurrent transmission, and $\tilde{c}_m^I$ is the hypothetical sum capacity if one other pair is added to the concurrent transmission.

Consider weighted permutation matrix $\alpha\boldsymbol{P}$. For the first iteration, we initialize by searching the permutation matrix left to right and then top to bottom. Find the first non-zero element and note its position $(i1, j1)$ in the matrix. Insert $i1$ into $\boldsymbol{t}_1$ and $\bar{\boldsymbol{r}}_1$, and $j1$ into $\boldsymbol{r}_1$ and $\bar{\boldsymbol{t}}_1$. Take note of its capacity $c_{i1,j1}$ and SNR $\gamma_{i1,j1}$. Search the rest of the permutation matrix for the next non-zero element and note its position $(i2, j2)$. If $i2$ is not in $\boldsymbol{t}_1$ and $\bar{\boldsymbol{t}}_1$, and $j2$ is not in $\boldsymbol{r}_1$ and $\bar{\boldsymbol{r}}_1$, then we call $(i2, j2)$ an *available* pair. We note its capacity $c_{i2,j2}$ and SNR $\gamma_{i2,j2}$. We then calculate the capacities with interference c_1^I and $\tilde{c}_1^I$, with the extra pair of transmitter-receiver being $(i2, j2)$, using Equations (2.1) and (2.2). If $\tilde{c}_1^I >= max(c_1^I, c_{i2,j2})$, that is, adding $(i2, j2)$ pair improves the throughput, then we call $(i2, j2)$ an *desirable*[3] pair with respect to the pairs already in $\boldsymbol{t}_1$ and $\boldsymbol{r}_1$, and insert $i2$ into $\boldsymbol{t}_1$ and $\bar{\boldsymbol{r}}_1$, and $j2$ into $\boldsymbol{r}_1$ and $\bar{\boldsymbol{t}}_1$. Otherwise, move to the next available non-zero element. If it is also desirable, then update $\boldsymbol{t}_1$, $\boldsymbol{r}_1$, $\bar{\boldsymbol{t}}_1$ and $\bar{\boldsymbol{r}}_1$ accordingly. Continue adding desirable pairs until one cannot be found after searching through n consecutive non-zero elements. At that point, stop because we have exhausted all the choices. The resulting set of concurrent transmissions given in $\boldsymbol{t}_1$ and $\boldsymbol{r}_1$ performs well under interference in the sense that its overall throughput is greater than each of the individual capacities. From $\boldsymbol{t}_1$ and $\boldsymbol{r}_1$, we can also calculate the new capacity matrix with interference $\boldsymbol{C}_1^I$.

Let $\boldsymbol{I}$ be the indicator matrix for $\boldsymbol{t}_m$ and $\boldsymbol{r}_m$, that is, it is an $n \times n$ zero-matrix, except at positions in the matrix that correspond to transmitter-receiver pairs in $\boldsymbol{t}_m$ and $\boldsymbol{r}_m$, where it is 1. For example, if $\boldsymbol{t}_m = (1, 3)$ and $\boldsymbol{r}_m = (2, 4)$, then $I_{1,2} = I_{3,4} = 1$. $\boldsymbol{I} .* \boldsymbol{f}^P$ is the matrix of flow that must be removed by the concurrent transmissions given

[3]Under this definition, a node pair must be available before it is desirable; therefore, desirable implies available.

in $\boldsymbol{t}_m$ and $\boldsymbol{r}_m$. We use the set of concurrent transmissions long enough (β_m seconds) to remove the smallest non-zero flow in $\boldsymbol{I}.*\boldsymbol{f}^P$. In other words, $\beta_m = min(\boldsymbol{I}.*\boldsymbol{f}^P./\boldsymbol{C}_m^I)$[4]. Once the smallest non-zero flow is removed, we retire the corresponding transmitter-receiver pair from $\boldsymbol{t}_m$ and $\boldsymbol{r}_m$ and restart search for additional desirable pairs. Each time a new set of concurrent transmission is found, m is incremented by one, $\boldsymbol{I}$, $\boldsymbol{C}_m^I$ and β_m are calculated and more flow are removed from $\boldsymbol{f}^P$. This continues until all flows in $\boldsymbol{f}^P$ are removed. At this point, the weighted permutation matrix $\alpha\boldsymbol{P}$ is retired, and we begin decomposing the next weighted permutation matrix. This continues until all weighted permutation matrices are decomposed. The series of sets of concurrent transmissions $\hat{\boldsymbol{t}} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \dots)$ and $\hat{\boldsymbol{r}} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \dots)$ is a time-division schedule under interference, and the series of weights $\hat{\boldsymbol{\beta}} = (\beta_1, \beta_2, \dots)$ becomes the amount of time each schedule is used. The total amount of time β_t it takes to remove the FD flow $\hat{\boldsymbol{f}}$ is equal to $\beta \sum_m \beta_m$, where β was derived in Equation (5.1). The capacity of the network while using the schedules $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{r}}$ is equal to the sum of the elements of the traffic matrix $\boldsymbol{T}$ divided β_t.

5.3 Scheduling Improvement

In the above section, the series of sets of concurrent transmissions $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{r}}$ and their weights $\hat{\boldsymbol{\beta}}$ were derived sequentially, i.e., $\boldsymbol{t}_1$, $\boldsymbol{r}_1$ and β_1 were obtained first, and so forth. Neither the sets of concurrent transmissions nor their weights are optimal. The sets of concurrent transmissions cannot be changed because they were the product of an suboptimal strategy we implemented in the previous section. However, the weights that corresponds to the sets may be optimized so that the same amount of flow may be removed while using less total time. This optimization problem is described below and

[4] ./ is the element-by-element division operator. Division by zero is ignored because we don't care about the values in those positions.

can be solved using linear programming [31].

$$\text{Given: } \hat{\boldsymbol{C}}^I$$

$$\text{Minimize: } \sum_m \beta_m$$

$$\text{Constraints: } \beta_m >= 0$$

$$\sum_m \beta_m \boldsymbol{C}_m^I >= \hat{\boldsymbol{f}}$$

where $\hat{\boldsymbol{C}}^I = (\boldsymbol{C}_1^I, \boldsymbol{C}_2^I, \dots)$ is the series of capacity matrices corresponding to the series of concurrent transmissions given in $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{r}}$. In words, the optimization problem seeks to minimize the total amount of time it takes to remove at least as much flow as the FD flow $\hat{\boldsymbol{f}}$, since there may not be an improvement if we were to require that exactly the same amount of flow is to be removed.

5.4 Simulation Results and Analysis

In the previous chapters, we have presented bounds with no interference and results of an protocol based on CSMA/CA. Now, we will show the performance of our time-division schedules relatives to those results.

5.4.1 Full traffic pattern

In Fig. 5.1, we show the performance of the time-division scheduling scheme presented in this chapter, in addition to the results presented in Fig. 4.6. We consider a full traffic matrix and normalizing SNR γ^{norm} of 20dB and 5dB. Channel model considers path-loss and shadowing. The new capacity curves are (c) and (d), which respectively represent the results for unoptimized TD schedules and the optimized TD schedules.

Let's first examine the case with $\gamma^{norm} = 20$dB. The unoptimized TD schedule

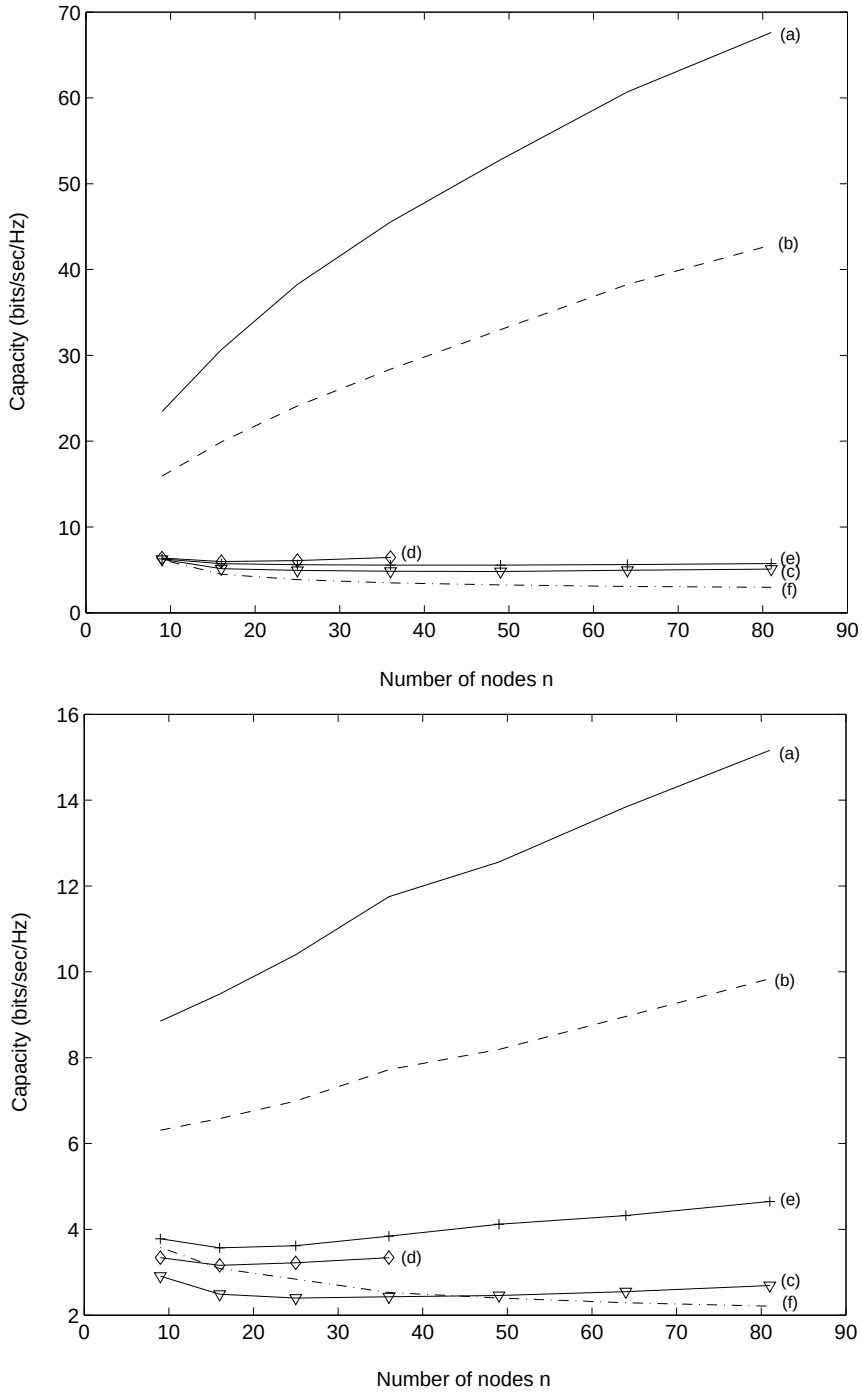

Figure 5.1 *Network capacity results obtained using different methods. Full traffic matrix and normalizing SNR γ^{norm} = 20dB (top) and 5dB (bottom). Channel considers path-loss and shadowing. (a) Average FD upper bound. (b) Average TD lower bound with no interference. (c) Average unoptimized TD results with interference. (d) Average optimized TD results with interference. (e) CSMA/CA-based scheme with SIRO queueing. (f) CSMA/CA-based scheme with FIFO queueing.*

results closely match that of the CSMA/CA scheme with SIRO queueing, which as we recall is very optimistic because unrealistic assumptions were made that idealized the performance of that protocol (i.e., control packets require negligible time to transmit, no control packet collisions, nodes can adjust their transmission rates instantaneously based on the number of concurrent transmissions at any point in time). No such unrealistic assumptions are necessary in our TD scheduling strategy because a central system computes the schedules and all transmission rates are pre-calculated because the sets of concurrent transmissions are known. This allows us to conclude that our unoptimized TD scheduling scheme can outperform a realistic implementation of CSMA/CA scheme.

The performance of the optimized TD schedules is shown in curve (d). Due to computational issues, which will be clarified in Section 5.5, it is only extended to networks of size 36. At $n = 36$, the optimized results are a 33% improvement over the unoptimized result. We expect, on average, the improvement to increase with n because the size of the sets of weights β_m increases with n, and larger sets mean there are more potential gain when optimizing them.

In terms of numbers, let's first look at the case of $n = 36$. Normalized by the FD upper bound, the TD lower bound with no interference is 62%, the unoptimized TD result with interference is 11%, and the optimized TD result is 14%. In the case of $n = 81$, and again normalized by the FD upper bound, the TD lower bound with no interference is 63%, the unoptimized TD result is 8%, and the optimized TD result is at least 10%, which is based on the assumption that the optimization provides at least a 33% gain over the unoptimized results.

Now, let's examine the case with $\gamma^{norm} = 5$dB. Here, we see that CSMA/CA scheme performs much better than our TD scheduling scheme. This is not surprising because the effects of interference is much less severe in the low SNR region. In the limit as the interference goes to zero, we expect the CSMA/CA results to approach the

optimal capacity, which is likely to be closer to the FD upper bound than the TD lower bound without interference, and we also expect the unoptimized TD results to approach the TD lower bound without interference because both are derived from the decomposed permutation matrices. However, we must keep in mind that a realistic implementation of CSMA/CA may still perform close to, if not below, the TD scheduling performance. Also, the performance improvement of the optimized TD schedules over unoptimized ones is 37%.

For $n = 36$, normalized by the FD upper bound, the TD lower bound with no interference is 66%, the unoptimized TD result is 21%, and the optimized TD result is 28%. For $n = 81$, the TD lower bound with no interference is 65%, the unoptimized TD result is 18%, and the optimized TD result is at least 24%.

5.4.2 Skewed traffic pattern

In Fig. 5.2, we show the results when considering a skewed traffic matrix. Compared to the performance under full traffic, the performance of TD scheduling under directional traffic is significantly better than CSMA/CA, with an improvement of almost 81%. However, under directional traffic, the gain from optimizing the weights of the TD schedules is only 23% at $n = 36$. Overall, for the same γ^{norm}, the ratios between the FD upper bound and the other results are significantly smaller. For $n = 36$, normalized by the FD upper bound, the TD lower bound without interference is 82%, the unoptimized TD result is 23%, and the optimized TD result is 29%. For $n = 81$, the TD lower bound without interference is 81%, the unoptimized TD result is 22%, and the optimized TD result is at least 27%.

5.4.3 Suboptimal Versus Optimal Scheduling

As a way to gauge the performance of our optimized TD schedules, we solved the optimization problem in Section 5.3 using the complete combinations of rate matri-

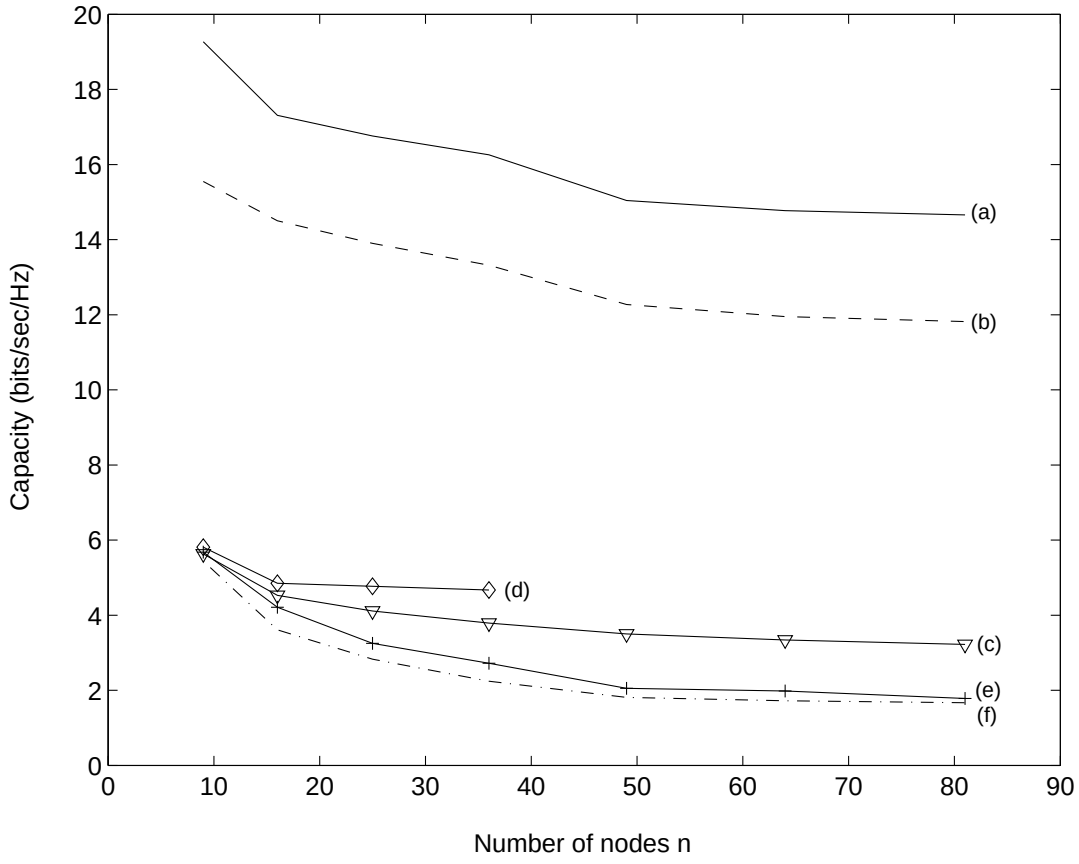

Figure 5.2 *Network capacity results obtained from different methods. Skewed traffic matrix and normalizing SNR γ^{norm} = 20dB. Channel considers path-loss and shadowing. (a) Average FD upper bound. (b) Average TD lower bound with no interference. (c) Average unoptimized TD results with interference. (d) Average optimized TD results with interference. (e) CSMA/CA-based scheme with SIRO queueing. (f) CSMA/CA-based scheme with FIFO queueing.*

Table 5.1 *Suboptimal versus Optimal TD scheduling for* $n = 9$

Scheduling Type:	Optimal	Suboptimal
Capacity (bits/sec/Hz):	6.71	6.38

ces for an 9-node network with full traffic matrix and $\gamma^{norm} = 20$dB. The results are shown in Table 5.1. As will be explained in Section 5.5, for a network with 9 nodes, the optimal scheduling scheme will require the calculation of 26,784 rate matrices, whereas our suboptimal strategy only require the calculation of a maximum of 5,832 rate matrices. However, even though the suboptimal strategy only considers a small fraction of the total available rate matrices, its performance is about 95% of the optimal performance. We expect the performance ratio to decrease with n, but the computational trade-off is much more significant as the optimal strategy becomes intractable as n increases even by a few nodes, e.g., there are over 4 billion rate matrices for a network of only 16 nodes.

5.4.4 Time-Varying Fading and Mobility

When the capacity performance of our TD schedules are obtained, we assumed the channel is static, that is, there are no mobility and time-varying fading. However, in any realistic setting, the channel gains change over time. Therefore, it is necessary to evaluate the degradation in the performance of our TD schedules when there are time-varying fading and/or mobility. We will isolate time-varying fading and mobility to two separate events and evaluate the degradation caused by each, where the channel gains change as a percentage of their means. The means of the channel gains are not necessarily defined in the statistical sense. It is defined as the instantaneous channel gain values when a set of TD schedules are calculated; i.e., TD schedules are calculated based on a snapshot of the channel gain matrix, and this snapshot is what we define as the means of the channel gains. TD schedules are updated either periodically or in response to some external command, and the duration between consecutive updates is dictated by the amount of change in channel gains that can be tolerated. We seek to

find the average performance degradation of TD schedules as a function of the speed of channel gain variation.

Performance degradation from time-varying channel is obtained as follows. Suppose that we have calculated a set of TD schedules, $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{r}}$, based on a snapshot of the channel at time t, which will be used for a duration of t_u, the period between TD schedule updates. Suppose further that the channel changes at time $t_c \in [t, t + t_u)$, causing most, if not all, of the link capacities to change. We assume that the channel changes on a time scale greater than the amount of time it takes to complete one cycle of the TD schedules, so the link capacities change but remain constant for one cycle of the TD schedule. After completing one cycle of the TD schedules, we will have delivered some amount of traffic equal to $\boldsymbol{f}'$. However, what we desired to have delivered was the amount of traffic given by the FD flow $\hat{\boldsymbol{f}}$. Comparing $\boldsymbol{f}'$ to $\hat{\boldsymbol{f}}$, we will see that some elements in $\boldsymbol{f}'$ will be greater than their counterparts in $\hat{\boldsymbol{f}}$, and some elements in $\boldsymbol{f}'$ will be less than their counterparts in $\hat{\boldsymbol{f}}$. If k is the minimum value such that $kf'_{ij} >= \hat{f}_{ij}$, for all i, j, then the performance degradation is $1/k$. In other words, using the TD schedules when the channel changes can only deliver an amount of traffic equal to $\boldsymbol{T}/k$.

Consider a set of TD schedules that are calculated based on a snapshot of the channel gains at some instance in time. In addition, assume that nodes move in two dimensions according to independent and identically distributed uniform random variables with range $[-d, +d]$, where d is normalized by the unit distance from which γ^{norm} is defined (e.g., if nodes are initially distributed 1km apart, and nodes move at a maximum of 10m in any direction, then $d = 0.01$). Fig. 5.3 shows the performance degradation in network capacity as a result of mobility. First, we notice that the larger networks suffer more degradation from the same amount of mobility. This is expected and can be explained by the fact that, on average, a larger network contains more links that are heavily degraded due to mobility, and each degraded link will increase the amount of time it takes to deliver the corresponding amount of flow in the FD flow matrix. Sec-

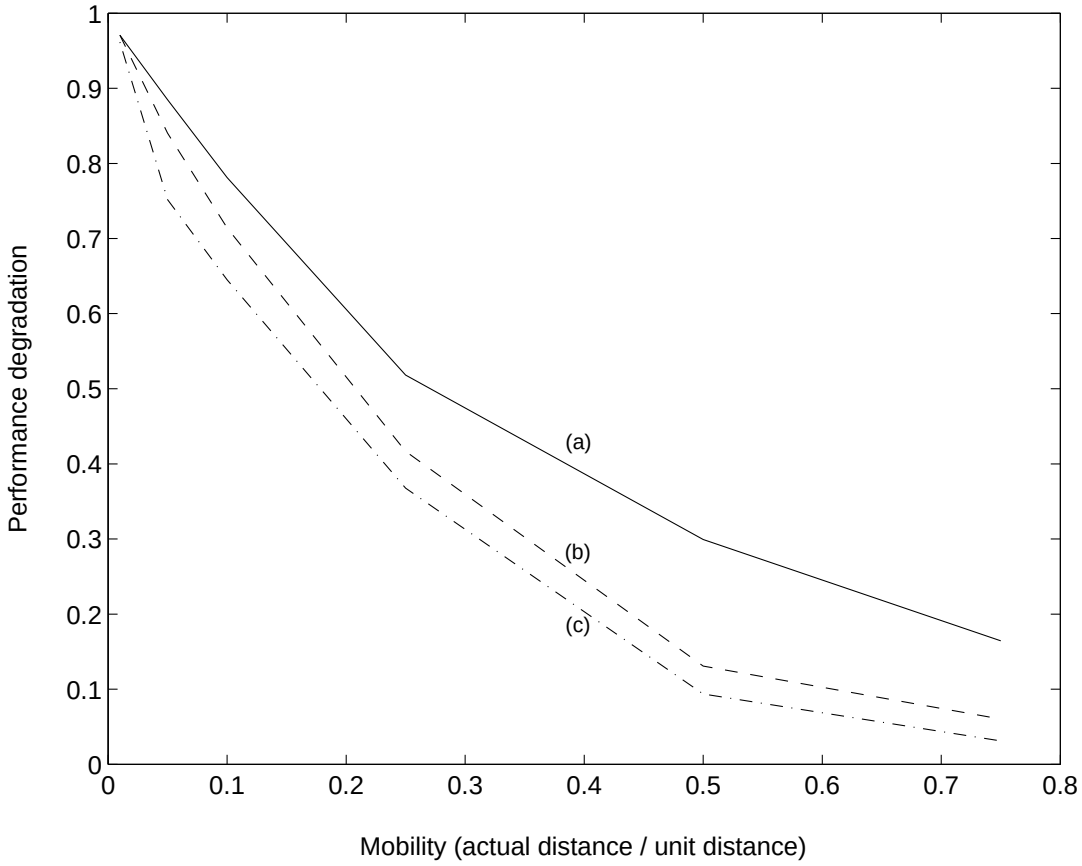

Figure 5.3 *Average performance degradation in network capacity as a result of node mobility. (a)* $n = 9$*. (b)* $n = 16$*. (c)* $n = 25$*.*

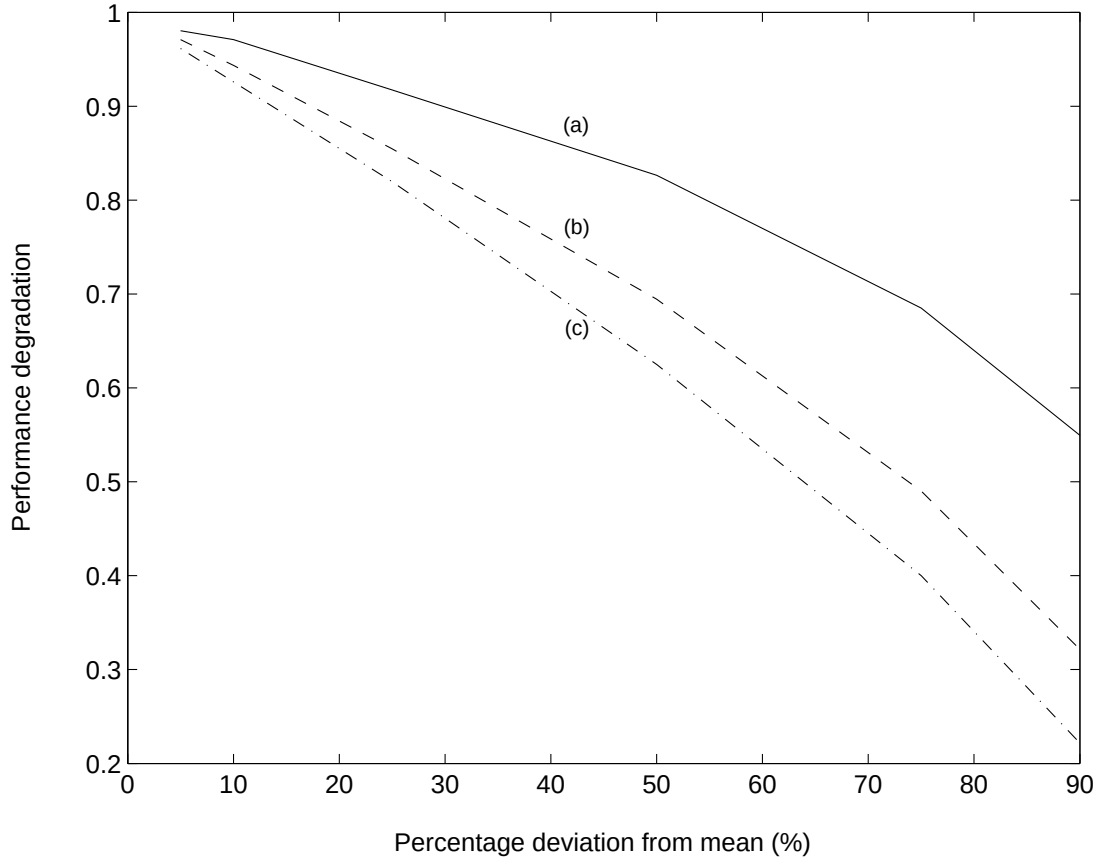

Figure 5.4 *Average performance degradation in network capacity as a result of time-varying fading. (a)* $n = 9$*. (b)* $n = 16$*. (c)* $n = 25$*.*

ondly, we see that even low mobility has an immediate impact on the network capacity. This is most likely due to the choice of attenuation factor used in the path-loss function, which is $\alpha = 4$. We expect the impact to lessen for smaller α and to grow for larger α.

Now let's consider the effects of time-varying fading on the network capacity. Assume that the channel gains are scaled by independent and identically distributed uniform random variables with interval between $[(1 - y/100), (1 + y/100)]$, where $0 \leq y \leq 100$ is the maximum percentage deviation. The channel is static when $y = 0$ and time-varying otherwise. Fig. 5.4 shows the performance degradation of network capacity as a function of y, caused by time-varying fading. Similar to the performance shown under mobility, we see that the larger networks suffers more degradation, and

this can be explained in the same way as previously. However, unlike mobility, small amount of time-varying fading does not cause significant degradation in network capacity because a change in fading is not exacerbated by a power factor of 4 as in the case with path-loss. Therefore, a network with immobile nodes can still perform well under our TD schedules as long as the time-varying fading does not vary by a large amount between schedule updates.

5.5 Computational Issues

In [25], the authors proposed a method to find the capacity region of an n-node wireless network by finding the convex hull of all the rate matrices. In a multihop network with spatial reuse, there are $\sum_{i=1}^{\lfloor n/2 \rfloor} \frac{n(n-1)\ldots(n-2i+1)}{i!} n^i + 1$ such rate matrices. Similarly, in the FD decomposition, to find the optimal decomposition of up to n^2 permutation matrices, we would need to search up to $\sum_{i=1}^{\lfloor n/2 \rfloor} \frac{n(n-1)\ldots(n-2i+1)}{i!}$ rate matrices for each permutation matrix[5]. However, the suboptimal strategy presented in Section 5.2 decomposes each permutation matrix into a maximum of $n(n-1)$ rate matrices, leading to a maximum of $n^3(n-1)$ rate matrices in the complete FD decomposition.

The optimization problem in Section 5.3 has n^2 simultaneous equations and $n^3(n-1)$ degrees of freedom, giving a maximum of $n^5(n-1)$ elements that must be stored in memory while they are being processed. This polynomial increase of memory requirement restricted the calculation of our optimal TD results to $n = 36$. Tables 5.2 and 5.3 shows the average time it takes to calculate a set of unoptimized and optimized TD schedules, respectively, for different network sizes.

[5]There are $n(n-1)\ldots(n-2i+1)$ distinct choices for the $2i$ nodes in i pairs of concurrent transmissions, but we must divide this by $i!$ to account for the fact that pair orderings are unimportant

Table 5.2 *Average time (sec) to obtain a set of unoptimized TD schedules*

$n =$	9	16	25	36	49	64	81
Avg. time =	0.1	1.1	7.0	31	110	320	840

Table 5.3 *Average time (sec) to optimize the TD schedules*

$n =$	9	16	25	36
Avg. time =	0.04	0.08	0.5	3.0

5.6 Conclusions

In this chapter, we proposed and described a new TD scheduling scheme derived from the FD results. We also described an improvement to the TD schedules that increased the total throughput significantly, even for small to medium size networks. Our TD schedules performed better than an idealized version of CSMA/CA, for medium to high SNR regions. Moreover, the TD schedules performed well against the FD upper bound when there is a directional traffic pattern, but not as well when the traffic pattern require all nodes to communicate with all other nodes. Finally, we examined the effects of time-varying fading and mobility on the TD schedules and found that mobility degrades the performance of the TD schedules much more than time-varying fading. This observation can be explained by the fact that mobility is exacerbated by the path-loss coefficient of $\alpha = 4$, while time-varying fading is not. Therefore, we conclude that our TD schedules performs well in an environment where the channel changes slowly relative to the TD schedule update rate. As such, the amount of time-varying fading and mobility we can tolerate affects how quickly we must update the TD schedules.

Chapter 5, in part, is a reprint of the material to be submitted to *IEEE Transactions on Wireless Communications*. The dissertation author was the primary investigator and author of this paper, and the co-author listed in these publications directed and supervised the research which forms the basis of this dissertation.

CHAPTER 6

Conclusions and Future Work

In this dissertation, we have made two major contributions in the consideration of the capacity of wireless ad hoc networks with arbitrary size, topology and traffic patterns. The first contribution is the calculation of an upper bound using the FD method. The upper bound is found by successively rerouting the traffic using relay nodes that are underutilized. This way, the relay burdens of the traffic demands are distributed across all the nodes, and the capacity of the network is maximized. However, the FD method operates on continuous flow elements, and this brings up two issues. One issue is we cannot enforce the requirement that a node can only establish half-duplex connection with one other node at any instance in time, i.e., it cannot transmit and receive simultaneously, or transmit to or receive from multiple nodes simultaneously. The other issue is that, with continuous flows, interference cannot be accounted for during the calculation of the upper bound because we do not know the set of concurrent transmissions at any given time.

The FD upper bound is not necessarily the tightest upper bound. In fact, if we enforce the requirement of half-duplex communications, then the FD upper bound is unachievable with high probability as the number of nodes in the network increases. However, we are able to prove that, under special cases, 2/3 of it is achievable, and under the worst case, 1/3 of it is achievable; that is, we can find a set of TD schedules that can achieve at least 2/3 or 1/3, depending on the situation, of the upper bound, in the absence of interference. We called the performance of this set of TD schedules the

TD lower bound without interference.

The FD method also allowed us to examine the behaviors of networks under different channel conditions and with different traffic patterns. We saw that if the channel conditions are known precisely, then shadow and multipath fading actually increase the network capacity because the network can take advantage of the links that are bolstered by the fading, while avoid those that are degraded. For networks with full or ring traffic patterns, the network capacity increases with n. Under the same traffic patterns, the capacity is an approximately linearly function with respect to SNR in dB, for small networks, and a polynomial function with respect to SNR in dB, for medium to large networks. For networks with skewed traffic pattern, the capacity actually decreases with n, and that the capacity functions with respect to SNR in dB for different network sizes only differ by some constant offsets.

The second contribution deals with both issues unresolved by the FD method. We produced a lower bound on the capacity by introducing a process to derive a set of TD schedules that performed well under interference. We are also able to optimize the TD schedules to increase its performance significantly. Both the unoptimized and optimized TD schedules performed better than an idealized protocol based on CSMA/CA, for medium to high SNRs. The performance of the TD schedules compared to the upper bound varied depending on the SNR and traffic pattern. For networks with medium to high SNR and full or ring traffic pattern, the upper bound is a magnitude larger than the performance of the TD schedules, but with low SNR, the upper bound is less than 4 times the performance of the TD schedules for networks of up to 81 nodes. For networks with skewed traffic pattern, even with moderate SNR, the upper bound is no more than 4 times the performance of the TD schedules. The TD schedules can also perform well under time-varying fading and mobility as long as the schedules are updated frequently enough so the fading and node positions do not change beyond some tolerance threshold.

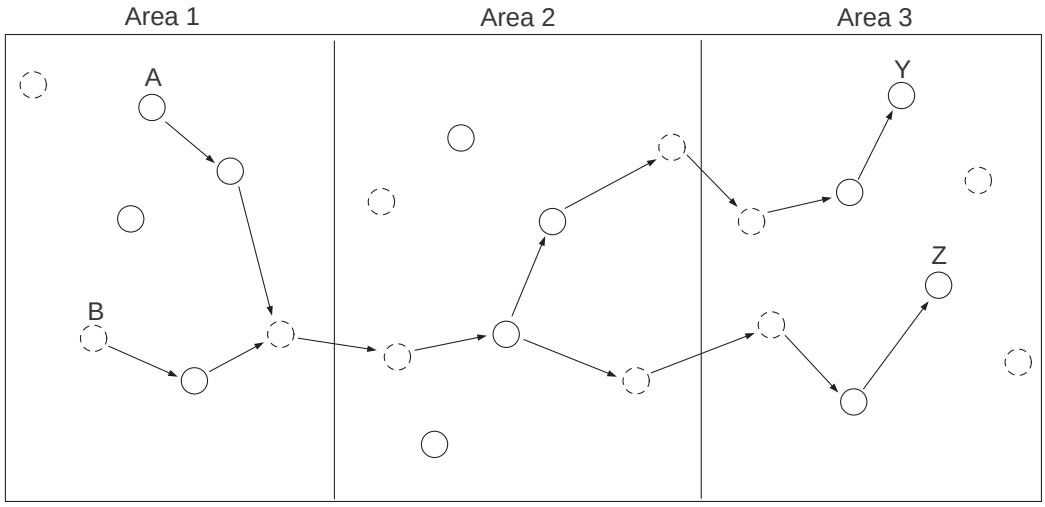

Figure 6.1 *An example of local and boundary traffic flows. Dotted circles denote boundary nodes. A has packet destined to Z, and B has packet destined to Y. A and B first use local TD schedules to transmit their packets to a boundary node in Area 1. The packets are forwarded to another boundary node in Area 2. They are routed locally in Area 2 from the boundary node close to Area 1 to boundary nodes close to Area 3. Finally, the packets cross into Area 3 using boundary TD schedules and are routed to their destination locally.*

The rate of update is dependent on the number of nodes and the computational power of the central station. If the update rate is too slow to accommodate the rate of change in channel conditions, then one suggestion is to break up the coverage area into multiple, smaller areas, each containing fewer nodes, and calculate local TD schedules for each area. To find a set of TD schedules, the original traffic matrix will have to be divided into two different types of traffic matrices: boundary traffic matrices denoting the traffic between boundary nodes in adjacent areas and local traffic matrices denoting the traffic between local nodes in the same area. If a node wishes to transmit to another local node, then a set of local TD schedules can be calculated from the channel and local traffic matrix. If a node wishes to transmit to another node in a different area, then a set of TD schedules can be calculated from the channel matrix and local and global traffic matrices (see Fig. 6.1). However, there must be some way to resolve time slot

contentions among the local and boundary TD schedules to minimize or avoid packet collisions. This is a subject of future work. Other possible subjects of future work include creating a more detailed performance function for the FD algorithm that takes into account realistic wireless network properties, such as interference, and finding a way to decompose the FD flow into a better set of TD schedules.

It must be noted that our work is ultimately non-information theoretic, meaning that our results are the byproduct of a set of specific transmission strategies that include multihop routing, spatial reuse, and variable transmission rates, and they may be proven suboptimal. However, it is a truism, in our field, that the ultimate usefulness of a theoretical work is its effects on the design and performance of practical systems. For a practical network that has a given size, topology and traffic pattern, our work provided a numeric upper bound against which current protocols may be compared, and a constructive lower bound that is achievable with current technologies.

REFERENCES

[1] R. Kahn, S. Gronemeyer, J. Burchfiel, and R. Kunzelman, "Advances in packet radio technology," *Proc. IEEE*, vol. 66, no. 11, pp. 1468–1496, November 1978.

[2] B. Leiner, D. Nielson, and F. Tobagi, *Proc. IEEE, Special Issue on Packet Radio Networks*, vol. 75, no. 1, January 1987.

[3] L. Kleinrock and J. Silvester, "Spatial reuse in multihop packet radio networks," *Proc. IEEE*, January 1987.

[4] C. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM*, October 1994.

[5] D. Johnson, "Routing in ad hoc networks of mobile hosts," in *Proc. ACM MobiComm*, December 1994.

[6] C. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, 1st ed. New Jersey: Prentice Hall, December 2001.

[7] M. Naghshineh, *IEEE Personal Commun. Mag., Special Issue on Advances in Mobile Ad Hoc Networks*, vol. 8, no. 1, February 2001.

[8] B. Leiner, R. Ruther, and A. Sastry, "Goals and challenges of the DARPA Glomo program (Global Mobile Information Systems)," *IEEE Personal Commun.*, pp. 34–43, December 1996.

[9] K. Negus, R. Stephens, and Lansford, "HomeRF: Wireless networking for the connected home," *IEEE Personal Commun.*, pp. 20–27, February 2000.

[10] B. Bing, R. Nee, and V. Hayes, *IEEE Commun. Mag., Special Issue on Wireless Local Area and Home Networks*, vol. 39, no. 11, November 2001.

[11] J. Haartsen and S. Mattisson, "Bluetooth - a new low-power radio interface providing short-range connectivity," *Proc. IEEE*, pp. 1651–1661, October 2000.

[12] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE ICASSP*, Salt Lake City, UT, May 2001, pp. 1–4.

[13] J. Proakis, E. Sozer, J. Rice, and M. Stojanovic, "Shallow water acoustic networks," *IEEE Commun. Mag.*, vol. 39, no. 11, pp. 1561–1571, November 2001.

[14] A. Michail and A. Ephremides, "Energy efficient routing for connection-oriented traffic in ad-hoc wireless," in *Proc. IEEE PIMRC*, vol. 2, London, UK, September 2000, pp. 762–766.

[15] A. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Commun. Mag.*, vol. 9, no. 4, pp. 8–27, August 2002.

[16] P. Gupta and P. R. Kumar, "Capacity of wireless networks," *IEEE Trans. Inf. Theory*, pp. 388–404, Mar. 2000.

[17] P. Viswanath and S. R. Kulkarni, "A deterministic approach to throughput scaling in wireless networks," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1041–1049, June 2004.

[18] S. Toumpis, "Capacity and cross-layer design of wireless ad hoc networks," Ph.D. dissertation, Stanford University, Stanford, CA, July 2003.

[19] M. Franceshetti, O. Dousse, D. N. Tse, and P. Thiran, "Closing the gap in capacity of random wireless networks via percolation theory," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 1009–1018, March 2007.

[20] L. L. Xie and P. R. Kumar, "A network information theory for wireless communication: Scaling laws and optimal operation," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 748–767, May 2004.

[21] O. Lévêque and E. Telatar, "Information theoretic upper bounds on the capacity of large extended ad hoc wireless networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 858–865, March 2005.

[22] M. Franceschetti, "A note on Lvque and Telatar's upper bound on the capacity of wireless ad-hoc networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 9, pp. 3207–3211, September 2007.

[23] M. Grossglauser and D. Tse, "Mobility increases capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[24] S. Diggavi, M. Grossglauser, and D. Tse, "Even one-dimensional mobility increases ad hoc wireless capacity," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, November 2005.

[25] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 736–748, July 2003.

[26] P. Bjorklund, P. Varbrand, and D. Yuan, "Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach," in *Proc. IEEE INFOCOM*, San Francisco, CA, March 2003.

[27] P. Stuedi and G. Alonso, "Computing throughput capacity for realistic wireless multihop networks," in *Proc. IEEE/ACM MSWiM*, Malaga, Spain, October 2006, pp. 191–198.

[28] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: an approach to store-and-forward communication network design," *Networks*, vol. 3, no. 2, pp. 97–133, 1973.

[29] E. Polak, *Optimization: Algorithms and Consistent Approximations*, ser. Applied Mathematical Sciences. New York: Springer, 1997, vol. 124.

[30] D. Bertsekas and R. Gallager, *Data Networks*, 1st ed. Englewood Cliffs, New Jersey: Prentice Hall, 1987.

[31] G. Dantzig and M. Thapa, *Linear Programming. 1, Introduction*, ser. Springer Series in Operations Research and Financial Engineering. New York: Springer, 1997.

[32] R. Bapat and T. Raghavan, *Nonnegative Matrices and Applications*, ser. Encyclopedia of Mathematics and its Applications. Cambridge: Cambridge University Press, 1997, no. 64.

[33] A. Goldsmith, *Wireless Communications*. New York: Cambridge University Press, 2005.

[34] H. Hashemi, "The indoor radio propagation channel," *Proc. IEEE*, vol. 81, pp. 943–967, July 1993.

[35] D. Parsons, *The Mobile Radio Propagation Channel*. New York: Wiley, 1992.

[36] P. Stuedi and G. Alonso, "Log-normal shadowing meets SINR: a numerical study of capacity in wireless networks," in *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, San Diego, CA, June 18-21 2007.

[37] "The IEEE 802.16 Working Group on Broadband Wireless Access Standards," Website, http://wirelessman.org.

[38] Y. Lin and Y. Hsu, "Multihop cellular: a new architecture for wireless communications," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, March 2000.

[39] R. Ananthapadmanabha, B. Manoj, and C. Murthy, "Multi-hop cellular networks: the architecture and routing protocols," in *Proc. IEEE PIMRC*, San Diego, CA, September 2001, pp. 78–82.

[40] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," in *Proc. ACM SIGCOMM*, London, UK, August 1994, pp. 212–225.

[41] C. Fullmer and J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," in *Proc. ACM SIGCOMM*, Cambridge, MA, August 1995, pp. 262–273.

[42] ——, "Solutions to hidden terminal problems in wireless networks," in *Proc. ACM SIGCOMM*, Cannes, France, September 1997, pp. 39–49.

[43] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999.

[44] T. Ozugur, M. Naghshineh, P. Kermani, and J. Copeland, "Fair media access for wireless LANs," in *Proc. IEEE GLOBECOM*, Rio De Janeiro, Brazil, December 1999, pp. 570–579.

[45] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proc. IEEE INFOCOM*, Miami, FL, March 2005.

[46] N. Gupta and P. Kumar, "A performance analysis of the IEEE 802.11 wireless LAN medium access control," *Communications in Information and Systems*, vol. 3, no. 4, pp. 279–304, 2003.

[47] G. Bianchi, "Performance analysis of the IEEE 802.11 Distributed Co-ordination Function," *IEEE Journal on Selected Area in Communications*, vol. 18, no. 3, 2000.

[48] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement," in *Proc. IEEE INFOCOM*, San Francisco, CA, March 1998.

[49] P. Ferré, A. Doufexi, A. Nix, and D. Bull, "Throughput analysis of IEEE 802.11 and IEEE 802.11e MAC," in *Proc. IEEE WCNC*, Atlanta, GA, March 2004.

[50] F. Ye, S. Yi, and B. Sikdar, "Improving spatial reuse of IEEE 802.11 based ad hoc networks," in *Proc. IEEE GlOBECOME*, San Francisco, CA, December 2003, pp. 1013–1017.

[51] F. Ye and B. Sikdar, "Distance-aware virtual carrier sensing for improved spatial reuse in wireless networks," in *Proc. IEEE GlOBECOME*, Dallas, TX, November 2004, pp. 3793–3797.

[52] L. Kleinrock, *Queueing Systems, Volume 1: Theory*, 1st ed. New York: Wiley, 1975.

[53] A. Dulmage and N. S. Mendelsohn, “Coverings of bipartite graphs,” *Canadian Journal of Mathematics*, vol. 10, pp. 517–534, 1958.