

# Low-Energy Mobile Packet Radio Networks: Routing, Scheduling, and Architecture

by

Hisham Ibrahim Kassab

S.B. Electrical Engineering, MIT (1993)

S.M. Electrical Engineering and Computer Science, MIT (1993)

S.M. Operations Research, MIT (1998)

Submitted to the Department of Electrical Engineering and Computer  
Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

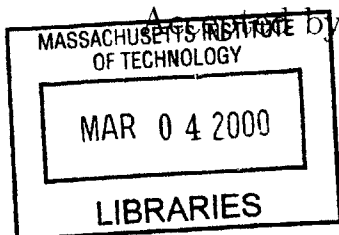
November 1999

February 2000

© Massachusetts Institute of Technology 1999. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
November 4, 1999

Certified by....  
Robert Gallager  
Professor of Electrical Engineering and Computer Science  
Thesis Supervisor



.....  
Arthur C. Smith  
Chairman, Department Committee on Graduate Students

# Low-Energy Mobile Packet Radio Networks: Routing, Scheduling, and Architecture

by

Hisham Ibrahim Kassab

Submitted to the Department of Electrical Engineering and Computer Science  
on November 4, 1999, in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy

## Abstract

Packet Radio Networks (PRNETs), which are also called ad-hoc networks, have the capability of fast (and ad-hoc) deployment and set-up, and therefore potentially have several useful civilian and military applications. Building low-energy PRNETs is an important design goal, because the communication devices are typically powered by batteries, and therefore are useless when the batteries are depleted. We choose to look at low-energy PRNETs by focusing on the problem of minimum-energy communication over a PRNET, resolving any related issues or design decisions in a manner consistent with the overall goal of low-energy PRNETs.

We conclude that the problem of minimum-energy communication over a PRNET is really a joint routing-scheduling-topological problem. We find the joint problem to be intractable, and therefore propose to solve it by decomposing it, solving each component separately. The resulting solution is not optimal but the degree of sub-optimality depends on how the problem is decomposed. Therefore we compare different decomposition methods, and select the one that is likely to yield the best solution to the joint problem.

After deciding how to decompose the joint problem, we study the separate components. For the topological problem we decide that nodes should communicate with a limited number of other nodes, referred to as neighbors. We also propose and analyze the performance of a procedure for managing the set of neighbors. For the scheduling problem, we propose a novel and practical class of scheduling algorithms. The routing problem is more complex than wireline routing because of interference and fading. When they are incorporated, routing becomes a non-convex problem; and we overcome this by a novel approach that is non-optimal, but is more robust than the optimal approach.

Thesis Supervisor: Robert Gallager

Title: Professor of Electrical Engineering and Computer Science

الحمد لله رب العالمين

## Acknowledgments

I wish to express my deepest gratitude and appreciation to my research advisor and mentor Professor Robert Gallager. His guidance and insightful ideas were invaluable to this research. Moreover, he spent many hours meticulously reading through early drafts of the thesis, which vastly improved the final version. Beyond that, however, he inspired, challenged, and motivated me to be the best researcher that I can be. I was in “good hands,” and it really was an honor and a privilege to be one of his students. I feel very fortunate to have had the opportunity.

I am also very thankful to the other members of our research group: Randy Berry, Dr. Steve Finn, and Dr. Jim Mills. Without their feedback, comments, and ideas, this thesis would not have reached its current form. Dr. Finn and Dr. Mills were also members of my thesis committee. Their academic and professional experience in the field of communications was very helpful, and I am extremely grateful for their careful readings of early drafts of the thesis. Moreover, I enjoyed and learned a lot from my conversations with them. My friend, colleague, and office-mate Randy was like a partner on this journey. I am very grateful for his generosity and support and wish him the best of luck with his career in academia. In addition, for his helpful comments and suggestions, I wish to thank Professor Hari Balakrishnan, who also was a member of my thesis committee.

Professor Alvin Drake left a lasting impression on me. I had one of my most memorable and educational experiences as a teaching assistant for his probability course. More importantly, I am extremely grateful for his support and encouragement at the early and critical stages of my graduate studies. I also wish to acknowledge Professor Donald Troxel, my academic advisor, for his advice and for keeping an eye on my progress.

For creating such a wonderful and intellectually stimulating environment to work in, I want to thank my colleagues at LIDS in general, and my office-mates in Room 35-303 in particular. I especially want to thank Thierry Klein for his friendship and for brainstorming with me on occasion.

I also wish to acknowledge the assistance of the administrative staff at the Department of Electrical Engineering and Computer Science, especially Marilyn Pierce, the unsung hero of EECS graduate students, for being infinitely supportive and accommodating over the years. Kathleen O’Sullivan and Monica Bell have also been very helpful.

On a more personal note, I want to thank all my friends at MIT, including those from the Arab Student Organization, the Lebanese Club, the Muslim Students Association, and the B-entry gang. They added a pleasant and much needed social dimension to my MIT experience. Issam Lakkis, Saad Mneimneh, Fadi Karamneh, and Rabih Zbib deserve special thanks for making possible one of my most enjoyable experiences at MIT, namely filming a short comedy. For their friendship and support, I also want to thank Christoforos Hadjicostis, Karim Hussein, and Assef Zobian.

I will forever be indebted to my best friend and the love of my life, my beautiful wife Nadine Alameh. Our partnership has truly enriched my life, and her boundless

love and constant support kept me going through some of the toughest times. I am also very grateful for her patience with my work schedule over the past year. I love her with all my heart, and I look forward to a long, happy, and prosperous life together.

Last but not least, I want to thank my mother Sanaa', my father Ibrahim, my brothers Samer and Kamal, and my sister Nada. A lot has changed in my life since I came to MIT as a freshman ten years ago. However, I could always count on their unconditional love and support, the encouragement of my siblings (whose growing up I sorely missed watching), the advice of my parents, and the prayers of my mother. I love them all dearly, and I hope one day we will be reunited in one place again.

As a young man, my father, Ibrahim, had to make a life-changing choice between the dream of pursuing his Ph.D. studies and supporting his parents and his siblings. He chose to fulfill his responsibility. I do not think I would be writing the acknowledgments for a Ph.D. thesis had it not been for the many hours he devoted to tutoring me during my childhood years, and his nurturing of my academic growth. I share this accomplishment with him.

The research in this thesis was supported through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAL01-96-2-0002.

*To my family  
Ibrahim, Kamal, Nada, Nadine, Samer and Sanaa'*

# Contents

<b>1</b>	<b>Introduction</b>	<b>14</b>
1.1	Thesis Motivation . . . . .	17
1.2	Initial Discussion of Problem . . . . .	19
1.3	Primary Development of The PRNET Model . . . . .	22
1.3.1	Topology . . . . .	23
1.3.2	Medium Access in a PRNET . . . . .	24
1.3.3	Interference in DS-CDMA . . . . .	26
1.3.4	Scheduling . . . . .	28
1.3.5	Summary of Partial PRNET Model . . . . .	30
1.4	Literature Review . . . . .	31
1.4.1	Topological Organization . . . . .	31
1.4.2	Scheduling . . . . .	32
1.4.3	Minimum-Delay Routing in Wireline Networks . . . . .	33
1.4.4	Routing in PRNETs . . . . .	36
1.5	Problem Statement . . . . .	40
1.6	Thesis Organization . . . . .	43
<b>2</b>	<b>The Scheduling Problem</b>	<b>44</b>
2.1	Minimum-Frame-Length Scheduling . . . . .	46
2.2	Neighbor-Based Scheduling . . . . .	49
2.2.1	Round-Robin Scheduling . . . . .	50
2.2.2	Binary Label Scheduling . . . . .	50
2.2.3	Bi-Orthogonal Binary Label Scheduling . . . . .	55

2.2.4	Simplex Binary Label Scheduling . . . . .	60
2.2.5	Summary of Neighbor-Based Scheduling . . . . .	62
2.3	Summary . . . . .	64
<b>3</b>	<b>Minimum-Energy Routing in Static PRNETs</b>	<b>65</b>
3.1	Completing The Model . . . . .	66
3.1.1	Summary of Interference . . . . .	70
3.2	Constructing A Convex Cost Function . . . . .	71
3.2.1	Adapting The Cost Function For Scheduling . . . . .	75
3.2.2	Comparable Levels of Energy Use . . . . .	78
3.2.3	Incorporating Delay . . . . .	79
3.3	Minimum-Combined-Energy-and-Delay Routing Formulation . . . . .	81
3.4	Decomposition of Routing and Power Control . . . . .	82
3.5	Distributed Implementation of MCED Routing . . . . .	83
3.6	Adjusting the Nominal Data Rates . . . . .	84
3.7	Summary . . . . .	87
<b>4</b>	<b>Minimum-Combined-Energy-and-Delay Routing in Dynamically Varying PRNETs</b>	<b>89</b>
4.1	The Effect of PRNET Dynamics on Routing . . . . .	89
4.2	Completing The Model . . . . .	93
4.2.1	Power Control in a Dynamic PRNET . . . . .	95
4.3	Development of A Cost Function for MCED Routing . . . . .	96
4.4	Summary . . . . .	98
<b>5</b>	<b>Analysis of a Neighbor Management Procedure</b>	<b>99</b>
5.1	The Data Channel and The Control Channel . . . . .	99
5.2	Hidden Node Discovery Procedure . . . . .	102
5.3	Splitting Algorithms for The HND Procedure . . . . .	106
5.3.1	MALOHA . . . . .	106
5.3.2	Uniform Splitting . . . . .	108



5.3.3	Minimum Splitting . . . . .	113
5.3.4	Single-Slot Splitting . . . . .	116
5.3.5	Comparing The Different Splitting Schemes . . . . .	118
5.4	Summary . . . . .	124
<b>6</b>	<b>Conclusion</b>	<b>125</b>
6.1	Thesis Contributions . . . . .	125
6.2	Directions for Future Research . . . . .	128
<b>A</b>	<b>Initial Approach: Mathematical Programming</b>	<b>131</b>
A.1	PRNET Model for Mathematical Programming . . . . .	132
A.1.1	Simple Example . . . . .	133
A.2	Formulation of The Mathematical Program . . . . .	135
A.2.1	Solving The Mathematical Formulation . . . . .	139
A.2.2	Convexity of Power in The Interference Channel . . . . .	140
A.2.3	Reducing The Number of Variables . . . . .	142
A.3	Summary and Conclusion . . . . .	142
<b>B</b>	<b>Medium Access Schemes for PRNETs with Emphasis on DS-CDMA</b>	<b>144</b>
B.1	Frequency Division Multiple Access . . . . .	144
B.2	Time Division Multiple Access . . . . .	145
B.3	Frequency Hopping . . . . .	145
B.4	Direct Sequence Code Division Multiple Access . . . . .	146
B.4.1	Power Control in DS-CDMA . . . . .	147
B.4.2	Similarities Between DS-CDMA and Frequency Hopping . . . . .	151
B.5	Random Access Protocols . . . . .	151
<b>C</b>	<b>Non-Convexity Results</b>	<b>153</b>
C.1	Non-Convexity of Aggregate Nominal Transmission Power . . . . .	153
C.2	Non-Convexity of Aggregate Average Transmission Power . . . . .	156

**D Transmission Matrices** 158

    D.1 Minimum-Frame-Length Scheduling . . . . . 163

**E Proof of Complexity of Minimum-Frame-Length Scheduling** 167

# List of Figures

1-1	A graphical representation of a PRNET. . . . .	15
1-2	The power advantage of multi-hop routing. . . . .	20
1-3	Transmission with two transmitters and one receiver. . . . .	28
1-4	Routing using backbone links. . . . .	40
1-5	Routing using backbone links with isolated subnetworks. . . . .	41
1-6	Interactions between the different algorithms. . . . .	42
2-1	Topology of a PRNET. . . . .	47
2-2	A feasible frame consisting of four slots. . . . .	47
2-3	A feasible frame consisting of minimum number of slots. . . . .	48
2-4	Scheduling using binary labeling. . . . .	50
2-5	A simple network with five colored nodes. . . . .	51
2-6	Labeling each node with a binary representation of its color. . . . .	52
2-7	Frame resulting from binary label scheduling. . . . .	52
2-8	Scheduling using ternary labeling. . . . .	53
2-9	Comparison between binary labeling and ternary labeling. . . . .	54
2-10	A variant bi-orthogonal BL schedule for $M = 4$ and $M = 5$ . . . . .	58
2-11	A variant simplex schedule for $M = 4, 5,$ and $6$ . . . . .	62
3-1	Transmission with one transmitter and one receiver. . . . .	67
3-2	Transmission with one transmitter and two receivers. . . . .	68
3-3	Transmission with two transmitters and one receiver. . . . .	70
3-4	A four-node simple network. . . . .	75
3-5	Modeling a link that is enabled in two slots. . . . .	80

4-1	Relative frequencies of events related to PRNET routing. . . . .	92
4-2	Time-scale for MCED routing relative to other time-scales. . . . .	94
5-1	Data channel and control channel interleaved. . . . .	101
5-2	Data channel and control channel in parallel. . . . .	101
5-3	A lost node attempts to find other nodes using increasing scope. . . .	104
5-4	Lost soul must filter out one of the nodes that respond. . . . .	106
5-5	MALOHA: Comparing $\Pr(S)$ for $x = 2$ and $x = 10$ . . . . .	108
5-6	Markov chain depicting binary and ternary uniform splitting. . . . .	109
5-7	The expected number of frames versus number of contending users. . .	112
5-8	The expected number of slots versus number of contending users. . .	113
5-9	Evolution of binary and ternary minimum splitting. . . . .	114
5-10	Markov chain depicting the evolution of single-slot splitting. . . . .	117
5-11	Comparing single-slot splitting with uniform splitting. . . . .	119
A-1	Illustration of frames and slots. . . . .	133
A-2	Paths of Session 1 and Session 2. . . . .	134
A-3	Frame for the simple network example. . . . .	135
B-1	A simple network. . . . .	149
C-1	Transmission with two transmitters and one receiver. . . . .	154
D-1	Links in a frame as dictated by topology. . . . .	164
D-2	A feasible frame consisting of four slots. . . . .	165
D-3	A feasible frame consisting of minimum number of slots. . . . .	166
E-1	Transforming an undirected edge to two directed edges. . . . .	168
E-2	Activity configurations for nodes. . . . .	169

# List of Tables

2.1	Comparison of scheduling algorithms for different values of $M$ . . . . .	63
5.1	Comparing splitting schemes using $E(S) + \sigma_S$ . . . . .	121
5.2	Comparing splitting schemes using $E(S) + 1.5\sigma_S$ . . . . .	122
5.3	Comparing splitting schemes using $E(S) + 2\sigma_S$ . . . . .	123
D.1	Growth of the number of admissible matrices with $N$ . . . . .	163

# Chapter 1

## Introduction

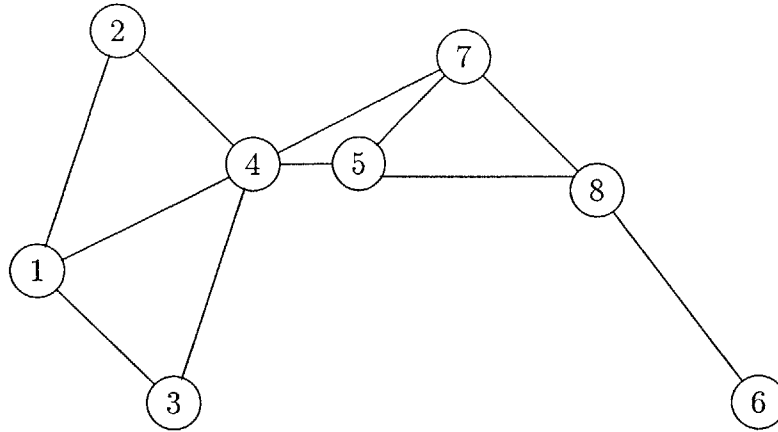
Packet Radio Network (PRNET) technology [Ka+78] was conceived as an extension to packet switching technology, which itself was introduced in the 1960's [BeG92] (pp. 1-5) for wireline data networks.<sup>1</sup> Packet switching technology helps achieve efficient utilization of network links, especially for bursty data traffic [BeG92] (pp. 14-16). In packet switching, each message is partitioned into packets. Each packet can then be stored and forwarded at each node from source to destination. This allows packets from several sessions to be multiplexed together on the same link. In packet radio networks (PRNETs), packet switching is extended to wireless broadcast radio networks.<sup>2</sup> The primary difference between PRNETs and traditional packet-switched networks is that in PRNETs there is one channel (i.e., the radio channel), that is shared and contended for by all the nodes in the network. In addition, the communication nodes in a PRNET are typically mobile.

Fig. 1-1 shows an example of a PRNET, represented by a graph. The nodes in the graph correspond to communication nodes, and the arcs in the graph connect pairs of communication nodes that have established the capability to communicate with each other. (We discuss what links signify in more detail in Section 1.2). The

---

<sup>1</sup>We use the term wireline network to include all networks with point-to-point links. A point-to-point link can be twisted wires, a microwave link over dedicated line-of-sight towers, an optical fiber etc.

<sup>2</sup>PRNETs are also known in the literature as ad-hoc networks [GuK97], and self-organizing wireless networks or SWAN's [ScB95].



**Figure 1-1:** A graphical representation of a PRNET.

nodes in a PRNET are equal peers in the sense that they have uniform technological capabilities. This is in contrast to cellular networks which today are the typical wireless communications networks for mobile users. In a cellular network, each cell consists of a centrally-located and stationary base-station, and the mobile nodes communicate only with the base-station (i.e., each mobile node is connected to the rest of the network through the base-station). The base-station is much more sophisticated than the cellular phone. In a PRNET, there is no base-station, and the nodes are capable of transmitting to and receiving from each other. Therefore it is possible to route data from source to destination over multiple hops. For example, referring to Fig. 1-1, if Node 1 wants to send a packet to Node 8, it can send the packet to Node 4 first. Node 4 in turn receives the packet and can then transmit it to Node 5, which in turn can forward it to Node 8. This is referred to as multi-hop communication.

The major advantage of a PRNET is that it has the potential for fast (and ad-hoc) deployment and set-up. Traditionally, PRNETs have been attractive for potential use in military applications, such as for soldiers in an infantry unit. They can also help achieve soft degradation under crisis conditions. For instance, if the normal mode of operation of the military network were cellular, and the base-station were hit, then

the nodes could shift to PRNET mode, and still be able to communicate. PRNETs have several other useful potential applications. For example, PRNETs could be used for communication between members of a rescue team, ships in a fleet, or indeed in any situation where prior communication infrastructure is absent.

Research and development in PRNETs began in the 1970's. The first landmark paper in the field is [Ka+78]. In that paper, the concept of the packet radio network is clearly defined, and several issues and questions associated with that technology are spelled out. Almost a decade later, another landmark paper, [LNT87], revised these issues in light of advancing technology.<sup>3</sup>

Despite the potential usefulness of PRNETs and nearly three decades of research in this field, the PRNET has not moved past the research stage to become a widely deployed technology like cellular technology. Thus far, there have been few major testbed realizations of PRNETs. The first is the DARPA<sup>4</sup> PRNET, which was initiated in 1972. The program aimed to study the feasibility of using PRNETs for military communications. In 1983, DARPA initiated the Survivable Adaptive Networks (SURAN) program, as the successor to the first PRNET program. The hope for SURAN was to build bigger and better packet radio networks, based on the lessons learned from the first DARPA PRNET. A third realization of PRNETs is what is referred to as Amateur PRNETs. As the name implies, Amateur PRNETs are set up and operated by amateur radio operators, and are not overseen by a single organization or standards committee. Consequently, they have not evolved in an organized fashion. Network management and routing techniques in Amateur PRNETs tend to be simple and ad-hoc. For more information about Amateur PRNETs we refer the reader to [KPD85], [BBS98], and [TAP99]. Finally, there is a current effort to incorporate PRNET technology into *home networking*. Home networking allows home electronic devices such as telephones, VCRs, refrigerators, home-security systems etc. to communicate with each other [Ibm99]. The HomeRF Working Group is one of the key players in the development of home networking [HRf99a]. The group has

---

<sup>3</sup>[LNT87] appeared with several other important papers in the PRNET literature in [SIP87].

<sup>4</sup>DARPA stands for Defense Advanced Research Project Agency.



designed a protocol, the HomeRF Shared Wireless Access Protocol (SWAP), for use in home networks. The SWAP system has the capability to operate in a PRNET mode [HRf99b].

## 1.1 Thesis Motivation

We use the term low-energy PRNETs to refer to PRNETs with low total/aggregate time-average power consumption. Building low-energy PRNETs is an important design goal for both civilian and military applications of PRNETs. The communication devices are typically powered by batteries, and are therefore useless when the batteries are depleted. This implies that energy utilization should be minimized so as to increase the time between required battery recharges.<sup>5</sup> For military applications, low-energy transmissions also lower the probability of an enemy detecting the existence of the PRNET. There are other desirable design considerations for a PRNET. It is beneficial that the PRNET be resistant to jamming, intercept, and listening-in. Jamming can occur if an adversary (i.e., the enemy in a military application, or a hacker in a civilian application) sends high-power noise on the communication frequencies, drowning the true signal(s). Intercept happens when the adversary detects the existence of the communication node from its radio transmissions. Listening-in refers to the adversary's ability to tune in on a communication link, and having access to the data sent on it. The need for these characteristics is self-evident within the context of military communications. It has also become important in commercial communication systems as customers become increasingly concerned about privacy.

There are many issues involved in the design of low-energy PRNETs, and we will not be able to resolve them all in this thesis. Instead, the central problem in this work is the problem of minimum-energy communication over a PRNET, and any related issues or design decisions that need to be addressed are resolved in a manner consistent with the overall goal of low-energy PRNETs. For now, we loosely de-

---

<sup>5</sup>Increasing time between battery recharges can also be achieved by building more efficient batteries, but that is outside the scope of our work.

fine minimum-energy communication as the problem of minimizing the time-average power radiated by the entire PRNET, while transporting data packets from source nodes to destination nodes.<sup>6</sup>

It is our aim to use mathematical analysis to gain insight into and understanding of the fundamental issues involved in minimum-energy communication over a PRNET. We hope that designers of practical algorithms that enable communication over a PRNET will use our results to develop “good” practical algorithms.

In order to use well-developed mathematical tools for the mathematical analysis, we do not preclude the use of models with simplifying assumptions. Such models may not capture all the complexities of the problem, but allow us to focus on the fundamental issues. For example, we make the simplifying assumption that the traffic entering the network for each origin-destination pair forms a stationary and ergodic arrival process.<sup>7</sup> This allows us to model data packets as flows. Now we can restate the problem of minimum-energy communication over a PRNET more formally as follows: we want to send data flows from sources to destinations, such that the total (radiated) energy used by the network, over some period of time of length  $T$ , is minimized while satisfying the end-to-end rate requirements. Over the period  $T$ , we assume that the data traffic in the PRNET is in steady-state. We also allow different nodes to transmit at different power levels, and the transmission power level of each node to vary over time. The period  $T$  is long relative to network delays, but not so long that the stationarity of end-to-end rates assumption is unreasonable. We observe that the objective of minimizing aggregate or total energy over the period  $T$  is equivalent to minimizing the aggregate time-average power over that period.

Given the collaborative nature of PRNET nodes in forwarding each other’s packets, it is reasonable to minimize aggregate energy used by the network. However, minimizing the aggregate energy does not guarantee that all nodes will use equal or even comparable levels of transmission energy, so some nodes’ batteries will be

---

<sup>6</sup>The problem is stated more formally below.

<sup>7</sup>Ergodicity is typically assumed when analyzing routing in data networks. The assumption allows us to base routing protocols on traffic patterns, because it allows us to assume that past traffic patterns give information about future traffic patterns.

depleted sooner than others'. One could extend the analysis for minimizing the aggregate energy to account for variability in energy use among different nodes. For example, one could impose a constraint on how much energy use by individual nodes can differ, or one could dynamically increase the system cost when batteries approach depletion.

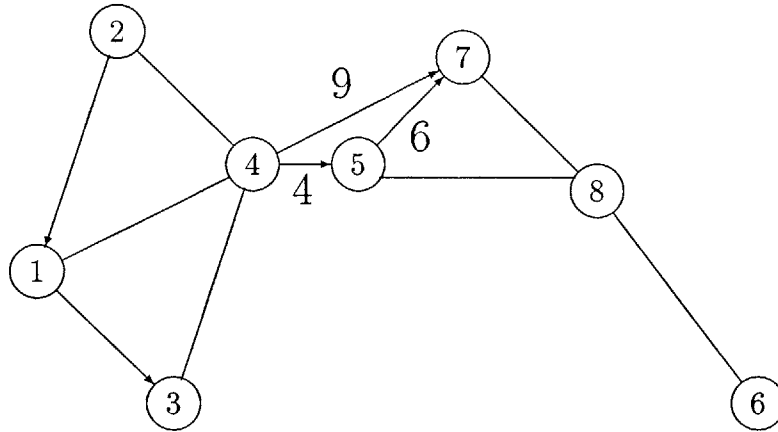
## 1.2 Initial Discussion of Problem

Initially, it may seem that minimum-energy communication over a PRNET is a non-problem. Since nodes have the ability to communicate directly with each other, source nodes can potentially transmit packets directly to the destination nodes by using a large energy per bit. However, there are at least two reasons why we need to consider multi-hop communication (i.e., sending packets through intermediate nodes). The first is rather trivial. Nodes can increase their transmission powers only up to a limit. If two nodes, say Node 1 and Node 2, are out of each other's transmission range, and there is another node between them, say Node 3, then the first two nodes can potentially still communicate through Node 3. For example, Node 1 can send a message (intended for Node 2) to Node 3, and the latter then forwards it to Node 2.

The second benefit of multi-hop communication is that it helps lower the network transmission energy, which is consistent with our general goal of low-energy PRNETs. It does so in a way analogous to children in a classroom forwarding a message by whispering from one person to another, rather than yelling out the message, lest it be heard by the teacher. The attenuation of radiated power is not linear in the distance; rather it is often proportional to a value between the distance squared and the distance to the sixth power [Lee82] (pp. 87-140). In other words, at distance  $\lambda$  from the source, the power of a signal is proportional to  $(\frac{1}{\lambda})^\theta$ , where  $2 \leq \theta \leq 6$ .<sup>8</sup> As a result, direct transmission may be more costly (energy-wise) to the network as a whole than multi-hop transmissions. We illustrate this point using the following example: consider the PRNET in Fig. 1-2, and suppose we want to send a message

---

<sup>8</sup>In fact sometimes transmission power can decay exponentially (e.g., in dense fog).



**Figure 1-2:** The power advantage of multi-hop routing.

from Node 4 to Node 7. Let the distances between these three nodes be as indicated in the figure, and assume the power to be proportional to the square distance, with a proportionality constant  $k$ . Then the power associated with the path of  $4 \rightarrow 5 \rightarrow 7$  is  $52k$ , and that associated with  $4 \rightarrow 7$  is  $81k$ . Therefore in this case multi-hop communication requires less energy. This effect is even more pronounced for  $\theta > 2$ .

We now explain why in Fig. 1-1 arcs are not drawn between every pair of nodes, even though all nodes can potentially communicate with each other directly. Nodes that communicate directly, need to have actually agreed to do so and, upon that agreement, established the capability to communicate with each other.<sup>9</sup> Two nodes that have done so are referred to as neighbors, and we say there is a *link* between the two neighbors. In Fig. 1-1, links are represented by arcs. By introducing the notion of a link, we are now able to talk about the *topology* of a PRNET, and say that it is characterized by the nodes and links of the PRNET.

Turning our attention to the problem of minimum-energy communication over a PRNET, we first observe that if the nodes in the PRNET exhibit high mobility,

---

<sup>9</sup>The establishment of the communication capability results from a handshaking protocol likely to happen over a designated control channel.

then the topology of the network will likely change rapidly. If the rate of change is sufficiently high then it will not be possible to construct communication algorithms that can react fast enough to the changes. In that case, the only viable solution is likely to be some form of a flooding algorithm [BeG92]. Therefore, we assume that the rate of change in the topology is slow enough that communication algorithms are allowed the time to intelligently and deliberately guide data packets from source nodes to destination nodes.

The minimum-energy communication problem consists of at least two problems. The first is the *topological problem* which is to determine the topological organization of the PRNET. The second problem is the *routing problem* which determines the paths from different sources to different destinations, and the flow rates on the different paths. We refer to the routing problem as the minimum-energy routing problem. There is a third problem that we have not yet motivated, but is discussed in detail in Subsection 1.3.4. That problem is the *scheduling problem*, and its purpose is to schedule the transmissions and receptions of each node such that no node transmits and receives simultaneously.

The three problems are interdependent. The selection of the links by the solution to the topological problem is based in part on the routing paths that need to be constructed from sources to destinations. The solution to the scheduling problem schedules transmissions among neighbors, which in turn are determined by the topological organization of the PRNET. Furthermore, as we explain in Section 1.3, nodes transmit DS-CDMA signals, and the transmission powers of simultaneously transmitted DS-CDMA signals depend on how much the signals interfere with each other. Therefore, in line with the goal of minimum-energy communication, we believe the solution to the scheduling problem tries to avoid allowing simultaneous transmissions that create too much interference for each other. Finally, the solution to the routing problem constructs the paths for different source-destination pairs using the links as determined by the topology. When minimizing energy, the routing algorithm needs to know the transmission schedule in order to assess the cost, in terms of transmission power (which is affected by interference), of sending packets on different links.

A natural first approach to the joint routing-scheduling-topological problem of minimum-energy communication over a PRNET was to formulate it as a constrained optimization problem, where the objective is to minimize a cost function, reflecting the aggregate average transmission power, subject to a constraint set. The constraint set includes constraints such as meeting end-to-end rate requirements, flow constraints, and scheduling constraints that ensure that no node transmits and receives simultaneously. This initial approach is described in detail in Appendix A. We found the approach to be not promising, primarily because the inclusion of the scheduling constraints makes the problem an intractable combinatorial problem.

The complexity of the joint problem suggests that it ought to be decomposed into its three separate component problems. Each problem can then be handled by a separate algorithm or set of algorithms. The separate algorithms communicate and coordinate among each other to form a solution to the original joint problem. Decomposing the original joint problem very likely yields a sub-optimal solution. The degree of sub-optimality, however, largely depends on how the original problem is decomposed.

### 1.3 Primary Development of The PRNET Model

Communication in PRNETs involves several issues that do not have counterparts in wireline packet-switched networks. These issues revolve around two factors (which we mentioned earlier but repeat for emphasis) associated with PRNETs:

F1: The nodes in the network share the same communication channel, since they all broadcast their signals over the same radio channel, (i.e., occupy the same bandwidth). Therefore a transmission by a node can potentially be heard by, and create interference at, nodes other than the intended receiving node(s).

F2: The topology of a PRNET is not static as it is for a wireline network. Nodes change their neighbors as they move around, and as the propagation characteristics of radio links change with time (including changes due to external

interference).

The design decisions we make in order to address these issues greatly influences how we approach the problem of minimum-energy communication over a PRNET. In this section we begin the construction of the PRNET model, going over any needed design decisions. By the end of this section we will have developed most of the PRNET model employed in this thesis. The incomplete model is summarized in Subsection 1.3.5. The piece missing from the incomplete model is whether the PRNET is modeled as a static network or a dynamic network; and in the case of a dynamic network, how the dynamic nature is modeled. We defer addressing these issues until Chapters 3 and 4, where we study minimum-energy routing in static PRNETs and dynamic PRNETs respectively.

### 1.3.1 Topology

The PRNET consists of  $N$  mobile nodes. We assume that each node has a unique ID. This facilitates the design and implementation of topological algorithms for managing the set of neighbors for a node. The PRNET is represented by a directed graph [ChL96] where the nodes in the graph are the communication nodes, and a pair of directed links (in opposite directions) connect each pair of neighbors. We use the shorthand notation  $(i, j)$  to denote the link from Node  $i$  to Node  $j$ . Note that  $(i, j)$  and  $(j, i)$  are two distinct links, emphasizing the difference in transmission directions.

In our PRNET model, we put an upper bound on the number of neighbors a node has. There are several reasons for doing so. First, in [TaK84] it is argued that if a node has too many neighbors, then typically a significant number of them will be distant. The power required to reach the distant neighbors will be high and will create undesirable interference to other nodes, which goes against our general goal of low-energy PRNETs. On the other hand, if nodes have too few neighbors the network will become somewhat disconnected. It is argued in [TaK84] that 8 is roughly the optimal average number of neighbors. This optimal average number of neighbors is found by a heuristic argument based on quite a few assumptions and approximations about the

medium access scheme, the interference, the network topology, and the geographic distribution of the nodes. We will expect the optimal average number of neighbors to be lower if we consider minimizing power as well as interference. Nonetheless, the essence of the argument for the existence of an optimal average number of neighbors seems valid. Under different assumptions for the network, the exact optimal average number of neighbors may be different, but it is likely to be on the order of 8.

We take the modified view that there is an optimal upper limit,  $M$ , on the number of neighbors each node should have. Imposing an upper limit maintains the reduction of a node's transmission energy and interference to other nodes, which is consistent with our main goal of minimum-energy communication, and also simplifies scheduling, as we see in Chapter 2. Furthermore, limiting the number of neighbors reduces the complexity of the routing algorithm. The fewer neighbors a node has, the fewer the paths that need to be considered for routing.

### 1.3.2 Medium Access in a PRNET

The purpose of a medium access method is to allocate the common channel resources, namely bandwidth, time, and space, among different contending transmitting nodes. The issue of medium access methods has received much attention in the PRNET literature [LNT87], [Pur87].<sup>10</sup> There are three fundamental techniques which are the basis for all the commonly used medium access schemes. They are time division multiple access (TDMA), frequency division multiple access (FDMA), and direct sequence code division multiple access (DS-CDMA). We describe the three techniques in some detail in Appendix B. The techniques fall under the classification of multiple access techniques, which strictly speaking apply to the scenario of one receiver and multiple transmitters. Nonetheless, they also provide mechanisms for sharing the channel resources among contending transmitters in the case of multiple receivers. There are many other multiple access schemes which are various combinations and extensions of FDMA, TDMA, and DS-CDMA. The most promising combination technique is

---

<sup>10</sup>Indeed, it has received a lot of attention in the wireless communication literature in general.



frequency hopping, which we also describe in some detail in Appendix B.<sup>11</sup>

Our choice for the medium access method is DS-CDMA. In DS-CDMA, each transmitted signal occupies the entire channel bandwidth simultaneously with all other transmitted signals. The channel bandwidth is typically much larger than the minimum required bandwidth needed to transmit the signal, assuming BPSK or QPSK modulation with a given bit-rate. Each DS-CDMA signal is spread over the bandwidth via a code/signature sequence unique to the signal. A receiver can de-spread and separate the signals by cross-correlating the total received signal with each signature. Spread signals are designed such that they partially interfere with each other without completely destroying each other. This allows a receiver to receive simultaneously from several transmitters (over the entire bandwidth). In addition, a transmitter can transmit simultaneously to several receivers. DS-CDMA has anti-jamming, anti-listening-in, and anti-intercept capabilities, which automatically address the design considerations mentioned in Section 1.1.<sup>12</sup> Moreover, DS-CDMA has an anti-multipath-fading capability, which is particularly useful in wireless communications.

Frequency hopping also has anti-jamming, anti-listening-in, and anti-intercept capabilities, as well as desirable properties for wireless transmissions. The question of which of the two, frequency hopping or DS-CDMA, is more suitable for PRNETs is an open one [Be+98]. We choose to focus on DS-CDMA because we can mathematically model interference in DS-CDMA as we see in the next subsection. Modeling interference is a major focus in this work, because interference directly affects the transmission power levels of the nodes. However, we argue that, since at some level frequency hopping and DS-CDMA are very similar, the results we develop assuming DS-CDMA carry over qualitatively to frequency hopping.

---

<sup>11</sup>In Appendix B we also give explanations and references for statements made about DS-CDMA and frequency hopping in this subsection.

<sup>12</sup>We note that listening-in can be further combatted through encryption. However, we view encryption as a separable issue that is not addressed in our research.

### 1.3.3 Interference in DS-CDMA

It is typically assumed that PRNET nodes broadcast their signals in an omnidirectional manner. Omnidirectional broadcasting has two major implications. The first is the potential of using it for passive acknowledgments [LNT87]. The second, and the more important to us, is that omnidirectional broadcasting causes interference among signals. In FDMA and TDMA, interference between signals in the same sub-band tends to be binary: they either completely destroy each other or do not interfere. In DS-CDMA interference between signals tends to occur more gradually, depending on the relative powers of the signals. Furthermore, as we clarify later in this subsection, the transmission powers of DS-CDMA signals are typically controlled such that signals overcome the interference from each other.

Interference complicates minimum-energy routing. When one introduces traffic on one link, the transmitted signal creates interference for other signals, forcing them to increase their powers to overcome the interference. A feedback effect ensues, as the transmitter of the first signal must then increase its power to overcome the increased interference from the other transmitters. The mutual increase of power continues until the transmitters reach an equilibrium point.<sup>13</sup> As we explain later in Subsection 1.4.3, in minimum-delay routing for wireline networks, when traffic is introduced on a link, the cost per packet of using *that* link increases. In contrast, in minimum-energy PRNET routing, when traffic is introduced on a link, the cost per bit of using *other* links increases, due to interference. Because interference changes the costs of other links, it is seen as the major reason why minimum-energy PRNET routing is considered more difficult than minimum-delay routing in wireline networks.

#### Modeling Power and Interference

In order to be able to formulate a cost function reflecting radiated energy in the PRNET, we need to be able to mathematically model how the transmitting nodes set

---

<sup>13</sup>The equilibrium will not always be guaranteed to exist if it is not feasible for the channel bandwidth to absorb all the transmissions. In that case some transmissions may need to be stopped.

their transmission power levels in DS-CDMA, particularly in the face of interfering signals.

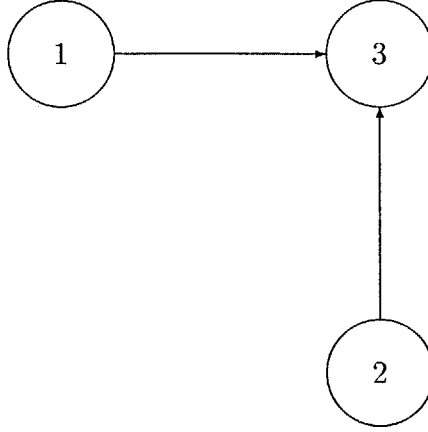
Let  $P_{ij}$  be the power used by Node  $i$  when transmitting to Node  $j$ . The signal experiences attenuation, where the power of the signal received by Node  $j$  is  $\alpha_{ij}P_{ij}$ . The parameter  $\alpha_{ij}$  ( $0 < \alpha_{ij} < 1$ ) is referred to as the path propagation loss. In our model, we assume that a transmitting node can adjust its radiated power level, such that the intended receiving node receives the transmitted signal at the required signal-to-interference ratio (SIR). The required SIR is the ratio that guarantees a specified symbol error rate. The SIR requirement translates to an  $E_b/N_0$  requirement, where

- $E_b$  is the energy-per-bit of the received signal.
- $N_0$  is the spectral density of the noise, which is typically assumed to be additive white Gaussian noise (AWGN). In the case where interfering signals are present, the interference of each signal is modeled as AWGN, with spectral density equal to the signal's power divided by the channel bandwidth. In that case,  $I_0$  is used instead of  $N_0$  to emphasize the presence of interference.

The higher  $E_b/N_0$ , the lower the bit-error-rate in the communication. We assume there is a required  $E_b/N_0$  threshold,  $\gamma_{ij}$ , for the communication from Node  $i$  to Node  $j$ , that guarantees a certain bit-error rate for that communication. Each signal is decoded separately, assuming other signals to be interferers.

In Chapter 3, we use examples to systematically illustrate how transmission powers can be derived from the  $E_b/I_0$  threshold requirement. However, we give a quick illustration here for the network example shown in Fig. 3-3. Let the transmission rate from Node  $i$  to Node  $j$  be  $c_{ij}$ . Then for the transmission from Node 1 to Node 3

$$E_b = \frac{\alpha_{13}P_{13}}{c_{13}} \quad \text{and} \quad I_0 = N_0 + \frac{\alpha_{23}P_{23}}{W}$$



**Figure 1-3:** Transmission with two transmitters and one receiver.

where  $W$  is the channel bandwidth. Therefore, the  $E_b/I_0$  requirement translates to<sup>14</sup>

$$\frac{\alpha_{13}P_{13}/c_{13}}{N_0 + \frac{\alpha_{23}P_{23}}{W}} = \gamma_{13}$$

A similar equation can be derived for the transmission from Node 2 to Node 3, and upon solving for the total transmission power in the network we get

$$\begin{aligned} P_T &= P_{13} + P_{23} \\ &= \frac{c_{13}\gamma_{13}WN_0(W + \gamma_{23}c_{23})}{\alpha_{13}(W^2 - \gamma_{13}\gamma_{23}c_{13}c_{23})} + \frac{c_{23}\gamma_{23}WN_0(W + \gamma_{13}c_{13})}{\alpha_{23}(W^2 - \gamma_{13}\gamma_{23}c_{13}c_{23})} \end{aligned} \quad (1.1)$$

In Appendix C we show that  $P_T$  as a function of  $c_{13}$  and  $c_{23}$  is not convex.

### 1.3.4 Scheduling

As a consequence of all the nodes sharing the same radio channel, we assume that the nodes must operate in half-duplex mode (i.e., they cannot transmit and receive simultaneously in the same frequency band). This constraint arises because the transmitted

---

<sup>14</sup>Strictly speaking the equality in the equation should be an inequality. This point is discussed in Chapter 3.

signal is much more powerful than, and consequently completely floods, the received signal. We note that this constraint will also arise if we use TDMA or FDMA.<sup>15</sup> In cellular DS-CDMA this constraint is handled by having the base-station receive and transmit in separate bands that are several megahertz apart. The separation is such that signals in one band do not spill into the other. The two-band solution however does not generally work in PRNETs. Suppose we have three nodes that need to transmit to each other. If Node 1 transmits in Band A and receives in Band B, then Nodes 2 and 3 must transmit in Band B and receive in Band A. But then Node 2 and Node 3 will not be able to communicate with each other. The deadlock can be resolved if more frequency bands are used. However, the preferred approach is to have all the nodes transmit over the same bandwidth, and use scheduling over time to ensure that no node receives and transmits simultaneously.

One possible, albeit trivial, scheduling scheme is to divide time into non-overlapping slots, and have nodes transmit in a round-robin fashion. This scheduling scheme results in a frame of length  $N$ , where  $N$  is the total number of nodes in the PRNET. A round-robin schedule is wasteful of the channel bandwidth, as in many cases, especially with the employment of DS-CDMA, it is possible for several transmitters to transmit simultaneously. This can lead to a higher overall network throughput, as the same total number of bits (on all the links) can be transmitted over a shorter period of time.

Scheduling schemes with shorter frame lengths are more desirable and generally possible. In addition to improving overall network throughput, scheduling schemes with shorter frames have smaller frame delays, which dominate the end-to-end delay when the network is lightly loaded. Shorter frames also imply smaller inter-packet delays. Indeed the goal of finding schedules that minimize the frame length is the aim of several papers that study scheduling in PRNETs, as we see in Section 1.4. We refer to the problem as minimum-frame-length (MFL) scheduling.

---

<sup>15</sup>The constraint is definitely true in TDMA, because the signals occupy the same frequency band. It is also true, but less serious, in FDMA, because signals in one sub-band spill into adjacent sub-bands. It can be eliminated in FDMA by placing the transmission sub-band far from the reception sub-band.

## Implication of Scheduling for Our Model

The need to schedule transmissions has a significant implication for our PRNET model, namely that time is divided into slots, in order to facilitate scheduling. We assume all the slots have equal lengths. The resulting medium access method is then *slotted DS-CDMA*, as opposed to pure DS-CDMA.

Furthermore, we assume that all the nodes have synchronized clocks, so that all the nodes agree on when each time slot starts and ends. Although clock synchronization has been made easier by improving clock technology, it remains an important and complicated problem especially for very large networks. Despite its importance, we ignore the issue of clock synchronization.

### 1.3.5 Summary of Partial PRNET Model

In this section, we investigated several design decisions for the PRNET, and constructed a partial model for the PRNET based on those decisions. The remainder of the model is discussed in Chapters 3 and 4. Below we summarize the partial model.

The PRNET consists of  $N$  mobile nodes, each of which has a unique user ID (UID). Each node has up to  $M$  neighbors, where  $M$  is a number to be determined. The PRNET is represented by a directed graph where the nodes in the graph are the communication nodes, and a pair of directed links (in opposite directions) connect each pair of neighbors. We use the shorthand notation  $(i, j)$  to denote the link from Node  $i$  to Node  $j$ . Note that  $(i, j)$  and  $(j, i)$  are two distinct links, emphasizing the difference in transmission directions. Each link is characterized by the parameter  $\alpha_{ij}$  which represents the path propagation loss along that link.

The medium access scheme employed by the PRNET is slotted DS-CDMA. We assume that the clocks in the PRNET are synchronized. Dividing time into slots facilitates scheduling, which is necessary since no node can transmit and receive simultaneously. The DS-CDMA component allows a node to transmit to multiple receivers simultaneously, and receive from multiple transmitters simultaneously. We assume there is a required received  $E_b/N_0$  threshold for all signals. We further assume

the use of conventional detection strategies under which each interfering user appears as white noise over the DS-CDMA band to the reception of other users.

## 1.4 Literature Review

In Section 1.2 we decided to approach the problem of minimum-energy communication over a PRNET by decomposing it into a minimum-energy PRNET routing problem, a scheduling problem, and a topological organization problem. Below, we review the literature for each of the three problems.<sup>16</sup> We also review minimum-delay routing in wireline packet switched networks, because we believe there are useful lessons that can help in our study of the minimum-energy PRNET routing problem. A more general overview of routing in wireline packet switched networks is given in [BeG92].

### 1.4.1 Topological Organization

One of the main problems in topological organization is clustering. A clustering algorithm refers to an algorithm by which the PRNET groups its nodes into clusters. There have been several papers in the literature on clustering algorithms for PRNETs. [Par85] tries to develop a clustering algorithm with the objective of minimizing the number of clusters in the network, such that each cluster has a cluster leader node (a clusterhead) that is linked directly to (i.e., a neighbor of) all the other nodes in the cluster. [Par85] found the minimization problem to be NP-Complete. If a problem is NP-Complete, then it is not known whether there exists a polynomial-time algorithm (i.e., an algorithm that requires a number of steps that is at most polynomial in the number of inputs to the problem) for solving it, and all known algorithms for solving it are exponential-time algorithms [GaJ79]. Typically, to solve an NP-Complete problem exactly, one must perform an exhaustive search over the entire solution space. Many combinatorial optimization problems tend to fall into the class of NP-Complete problems. One suspects that if a clustering algorithm tries to

---

<sup>16</sup>This section is a relatively long one. The reader may skip it, except for Subsection 1.4.3, without loss of continuity.

group nodes into clusters according to a constrained minimization or maximization objective, thus forming a combinatorial optimization problem, then the problem will be NP-Complete. In general, heuristic clustering algorithms, which do not guarantee an optimal solution, are proposed, such as the ones in [Par85], [BaE81], and [EWB87].

In terms of neighbors, as mentioned in Subsection 1.3.1, [TaK84] addresses the question of what the optimal average number of neighbors is. The work in [TaK84] is based on earlier work in [KlS78].

## 1.4.2 Scheduling

Minimum-frame-length (MFL) scheduling is the focus of several papers that study scheduling in PRNETs, such as [Ari84], [HaS88], [RaL93], and [SeH97]. Unfortunately, the scheduling constraints are not the same in all the papers, and more importantly, differ from our own scheduling constraint of no simultaneous transmit-receive. The difference in the scheduling constraints is due to having different PRNET models, especially regarding the medium access method. Despite not agreeing on the scheduling constraints, the above papers generally conclude that the problem of MFL scheduling is an NP-Complete problem. The only exception is [HaS88] which finds a polynomial-time algorithm for solving the problem. However, by the authors' own admission, the polynomial-time algorithm in [HaS88] would be difficult to implement in practice.

Due to the difficulty of MFL scheduling, some papers suggest different approaches to scheduling. For example, [WBE94] suggests using neural networks to solve the scheduling problem. Another example is [She96], where slots are not grouped into frames. Instead [She96] proposes that each node label a time slot pseudo-randomly with a 0 or a 1. If Node 1 wishes to transmit to Node 2, it will do so in a slot which is labeled 1 by Node 1 and 0 by Node 2. It is assumed that nodes, that wish to communicate with each other, know each other's pseudo-random sequences used for labels, so they know a priori when such slots occur. With this scheme, and assuming



nodes have synchronized clocks,<sup>17</sup> a node is able to transmit to any other node in one-quarter of the slots. This scheduling algorithm is novel, but it suffers from the drawback of not grouping slots into frames, making the delay of end-to-end sessions less controllable.<sup>18</sup>

The existence of an interdependence between scheduling and PRNET routing is recognized in [HaS88] and [WBE94]. Both papers investigate the joint routing-scheduling problem, although the emphasis is more on MFL scheduling than on finding good routing paths. These studies confirm that joint routing-scheduling is a hard problem.

### 1.4.3 Minimum-Delay Routing in Wireline Networks

A survey of minimum-delay routing in wireline networks is given in [BeG92] (pp. 363-478). An appealing approach to minimum-delay routing is to assign each link in the network a length that estimates the delay on that link; and subsequently apply a shortest-path algorithm to find the path, between the source and destination nodes, with the least total estimated delay. The shortest-path problem is well studied, and can be solved using well-known algorithms such as the Bellman-Ford algorithm and Dijkstra's algorithm. We refer to the general approach of assigning lengths to links and selecting the route with the smallest total length as *shortest-path routing*. Min-hop routing (i.e., routing over the smallest number of hops) is a special case of shortest-path routing, with each link assigned a length of one unit. We use the term *shortest-delay-path routing* to refer to the special case of assigning delay as the link length.

Shortest-delay-path routing suffers from two drawbacks that are explained in [BeG92] (pp. 370-372). One of the drawbacks is related to the fact that the delay on a link is a function of the amount of data flow on it. This implies that as flows are added or removed from different paths, the lengths of those paths change.

---

<sup>17</sup>[She96] analyzes the performance of this scheduling scheme in the case where clocks are not synchronized.

<sup>18</sup>As we see in Chapters 3 and 4 having a frame structure also simplifies the formulation of the minimum-energy routing problem.

This feedback effect can and does in certain situations affect the stability of the shortest-delay-path routing algorithm.

Despite its drawbacks, routing is typically implemented using shortest-delay-path routing [e.g., in NSFNET [BeG92] (pp. 374-375)]. In NSFNET, which is a datagram network, additional mechanisms are employed to improve the stability of shortest-delay-path routing.

Analytically, the drawbacks of shortest-delay-path routing are overcome in an alternative approach, which is referred to in the literature as *optimal routing* [BeG92], [CaG74], [TsB86]. It is unfortunate that the approach is referred to as optimal routing, because it is optimal only with respect to certain criteria. Whereas in shortest-delay-path routing the objective is to minimize the delay for a packet without regard to delays of other packets, in optimal routing the objective is to minimize the average end-to-end delay experienced by all packets in the network.

The optimal routing problem is formulated as a constrained optimization problem with the global cost function

$$\sum_{(i,j)} \frac{f_{ij}}{c_{ij} - f_{ij}} \tag{1.2}$$

where  $f_{ij}$  is the flow rate on  $(i, j)$ , the link from Node  $i$  to Node  $j$ , and  $c_{ij}$  is the nominal data rate of that link. The term  $\frac{f_{ij}}{c_{ij} - f_{ij}}$  in the summation approximates the average number of packets waiting to be transmitted on  $(i, j)$ . The expression is really the average number of packets for an M/M/1 queue with packets arriving according to a Poisson process with rate  $f_{ij}$ , and having exponentially distributed lengths. In practice the approximation is not likely to be accurate. In the Internet, for example, a significant number of packets are short control packets, which likely gives rise to a bimodal distribution of packet lengths. Nevertheless, the M/M/1 approximation captures the important effect that as the flow rate  $f_{ij}$  approaches the nominal data rate  $c_{ij}$ , congestion starts to set in.

The cost function in Eq. 1.2 reflects the average number of packets in the network, which by Little's law is proportional to the average end-to-end delay experienced by

a packet entering the network. The global cost function is minimized subject to the constraints that

1. Link flow rates are non-negative.
2. End-to-end (stationary) rate requirements are met.
3. Flow is conserved (i.e., flow from a particular end-to-end session entering a node equals flow from that end-to-end session exiting it).

The exact details of optimal routing are explained in [BeG92]; but to summarize, in optimal routing, incremental flow is added on the path that results in the least incremental cost to the whole network (where the cost is in terms Eq. 1.2). One strong reason for using the above cost function as an approximation is that it is convex over the space of link flow rates (i.e., the  $f_{ij}$ 's). It is well-known that convexity of the cost function greatly simplifies global minimization. Indeed several results and algorithms pertaining to optimal routing rely on the convexity of the cost function [Gal77], [CaG74]. Therefore it is important that the global cost function developed for minimum-energy routing be convex. We refer to the approach of formulating the routing problem as a minimization of a convex cost function subject to linear flow constraints as the *optimal routing approach*. We use the term *optimal-delay routing* to denote the specific application of the optimal routing approach to minimizing delay, as explained above. We again emphasize that optimal routing is only optimal with respect to certain criteria.

The two approaches of shortest-delay-path routing and optimal-delay routing are not totally distinct from each other. As mentioned in [TsB86] and shown in [GaB87], under certain conditions, most significant of which is having a network with a large number of small users utilizing virtual circuits, shortest-delay-path routing approaches optimal-delay routing in an asymptotic sense. Moreover, the optimal-delay routing problem can be solved by solving a series of shortest-path problems, in which the length assigned to each link is the first derivative of delay with respect to the link flow rate. Indeed, converting the optimal-delay routing problem into a series of shortest-

path problems is the best-known way of implementing optimal-delay routing in a distributed way [Gal77].

We briefly discuss the issue of centralized routing versus distributed routing. Typically it is preferable to implement a routing algorithm in a distributed manner, as opposed to a centralized one. A distributed implementation of routing has several benefits [Gal77], [BeG92]. For example, by not relying on a central node to compute the routes, the routing algorithm is made more robust versus link and node failures. At the early stages of designing a distributed algorithm, it is not uncommon to design, study, and analyze the algorithm as a centralized algorithm, in order not to get bogged down in issues related to distributed systems. As long as issues linked to a distributed implementation do not alter the problem fundamentally, the centralized algorithm can eventually be converted to a distributed one (e.g., the Bellman-Ford shortest path algorithm [BeG92] and the optimal-delay routing algorithm [Gal77]).

#### 1.4.4 Routing in PRNETs

The type of PRNET routing algorithms considered in the literature are mostly of the shortest-path type. One notable exception is [Yat96] in which a preliminary investigation of applying the optimal routing approach to minimum-energy routing in PRNETs is presented. The fundamental difference between the shortest-path algorithms proposed in the literature is the distance metric used to measure the length of a link. For example, in [Be+89]<sup>19</sup> the length of a link is a measure reflecting the probability of successful transmission and of interference. In [Ste88],<sup>20</sup> the length of a link is the number of nodes that can overhear (presumably above a certain threshold) the transmission from the source of the link to the destination of the link. In [PuR93] the routing algorithm is referred to as least-resistance routing (LRR), and the distance metric used is a function of the interference experienced by the receiving node on the link. [GuK97] uses the more familiar distance metric of mean delay. The novelty of

---

<sup>19</sup>The description here is based on the summary in [Lau95], due to unsuccessful attempts to obtain the original paper.

<sup>20</sup>Again, the description here is based on the summary in [Lau95], due to unsuccessful attempts to obtain the original paper.

the algorithm proposed in [GuK97] is that clocks at different nodes need not be synchronized. However, the paper confuses the difference between shortest-delay-path and optimal-delay routing. In [She96] minimum-energy routing is proposed, with the distance metric being the reciprocal of the link's channel path propagation loss,<sup>21</sup> which is proportional to the energy required to transmit a packet over that link. This measure is a first order approximation, and ignores the effect of the interference that different transmitters create for each other, as described in Subsection 1.3.3. Finally, in the DARPA PRNET, min-hop routing is used [JuT87].

The variety of choices for link lengths, and the choices themselves indicate that the PRNET routing algorithms in the literature are ad-hoc, and that the aim is to propose “reasonable” distance metrics, without rigorously justifying why such metrics are indeed reasonable.

The routing algorithms mentioned above are implementable in a distributed fashion, and ought to work correctly for any network size. However as the number of nodes in the PRNET gets larger, the frequency of changes that the routing algorithm needs to respond to increases. At some point the time between changes becomes smaller than the settling time of the distributed routing algorithm (i.e., the duration of the transient reaction of the distributed algorithm). At that point, methods for reducing the settling time of the distributed algorithm will become almost necessary, if the distributed routing algorithm is to be able to track changes in the network. One such method is the consolidation of routing information for groups of nodes. For instance the routing information propagated in the Internet pertains to domains of nodes, rather than individual nodes. So when determining a path to a node, typically a path to the domain is determined, and once the data reaches the domain, it is routed locally. The consolidation of routing information about groups of nodes implies that a local routing algorithm running on a node need not react to every single small change in the network. This in turn helps reduce the settling time of the distributed routing algorithm. Consolidation of routing information also helps reduce the amount of routing information carried by the network, as well as the complex-

---

<sup>21</sup>The channel path propagation loss is the fraction of transmitted power seen at the receiver.

ity of the local routing algorithms running at the nodes. We will say a PRNET is large-scale if the number of nodes in it is large enough that without consolidation of information it becomes almost impossible to implement a routing algorithm.<sup>22</sup>

Several approaches (based on consolidation of routing information) are proposed for routing in large-scale PRNETs. The first is hierarchic cluster routing, and it relies on having nodes organize themselves into clusters. In [EWB87] and [BaE81], it is proposed that each cluster be identified by a unique clusterhead with all the other nodes in the cluster restricted to be neighbors of the clusterhead. Within each cluster, some of the nodes, which do not include the clusterhead, are designated as gateway nodes. A gateway node is a node that provides a link from one cluster to an adjacent cluster (i.e., it communicates directly with another gateway node in the adjacent cluster). Accordingly, the clusterheads and the gateway nodes form the skeleton of the network. Any pair of non-neighbor nodes may communicate over the skeleton network. There are several aspects that make this large-scale PRNET routing algorithm unappealing. First, since each cluster consists of nodes that are one hop away from the clusterhead, and some of those nodes are designated as gateways, it seems that the resulting skeleton network typically contains a significant fraction of the nodes in it. In other words, it is not clear whether this large-scale PRNET routing algorithm is scalable. Second, in the skeleton network clusterheads are not permitted to communicate directly. Their communication has to go through two gateway nodes, even if it might make sense for them to communicate directly.

Another version of hierarchic cluster routing is discussed in [Lau95] and [Kat96], where clusters are not restricted to have a clusterhead that is a neighbor to all the nodes in the cluster. However, an inherent burden in hierarchic cluster routing is configuring and reconfiguring clusters as the mobile nodes move around. As mentioned in Subsection 1.4.1, heuristic clustering algorithms are proposed, because optimal clustering problems are generally NP-Complete.

A second approach for large-scale PRNET routing is routing with backbone links.

---

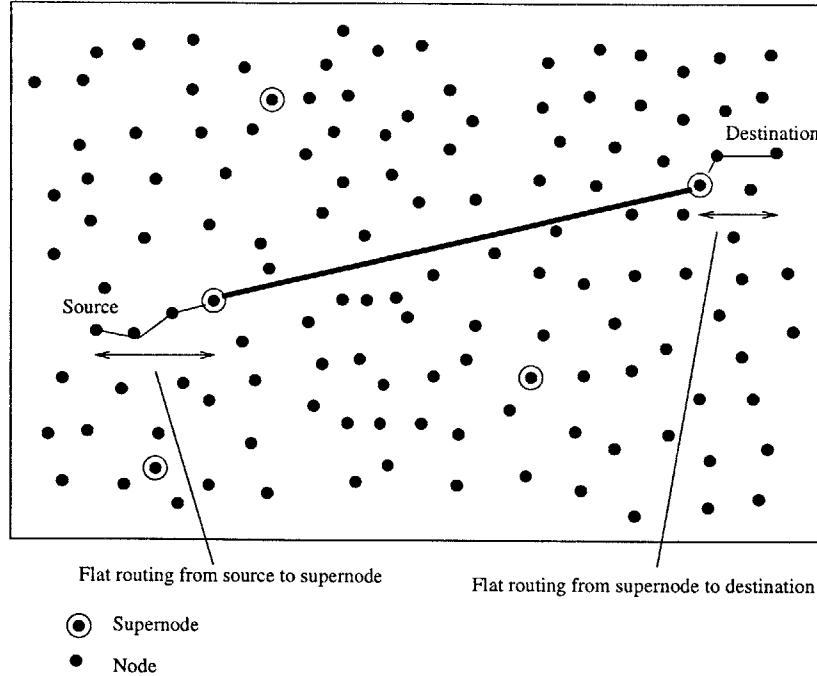
<sup>22</sup>Assuming there is no other alternative for reducing the settling time of the distributed routing algorithm.

In this approach, the restriction that all the nodes in the PRNET be equal is relaxed. In particular, the notion of a supernode is introduced. In a military context a supernode can be an Unmanned Air Vehicle (UAV), a satellite, or a jeep. A supernode has more technical capabilities and more power available than a regular node. Most importantly, the supernodes can send and receive at much higher rates than regular nodes. For example, line-of-sight microwave communication between UAVs or jeeps can provide higher channel capacities. The supernodes form a backbone network that overlays on top of the PRNET. The supernodes might communicate among each other over a different frequency band than used by ordinary PRNET nodes to communicate with each other and with supernodes. The notion of a backbone network consisting of supernodes is proposed in [Ka+78].<sup>23</sup>

When the source and the destination nodes are far away from each other, the source node need only worry about routing the data to the nearest supernode, which can be done using an appropriate flat routing algorithm. By flat routing, we mean routing among a group of equal nodes, where the topology of the group is fully known to each node. The shortest-path PRNET routing algorithms discussed earlier are examples of flat routing algorithms. The supernode routes the data to the supernode closest to the destination node using the high capacity backbone network. The second supernode then routes the data to the destination node. This is illustrated in Fig. 1-4. Routing with backbone links will also be appropriate if the large scale network consists of several isolated small-scale PRNETs as shown in Fig. 1-5. The situation depicted in Fig. 1-5 closely resembles a cellular network consisting of a group of cells. The only difference is that within a cell or a cluster, a mobile's packets are not necessarily transmitted directly to the base-station/supernode. We believe that routing with backbone links is the most suitable approach for large-scale PRNET routing because, as evidenced by the method's prevalence in large-scale wireline networks, it seems to be the more convenient method for handling large-scale routing. We emphasize that flat routing, which is used in small-scale PRNETs, is also critical in large-scale PRNET routing. For example, it is a building block for routing using backbone links.

---

<sup>23</sup>Supernodes are referred to as stations in [Ka+78].



**Figure 1-4:** Routing using backbone links.

There are other proposed approaches in the literature for large-scale PRNET routing, such as hierarchic landmark routing [Tsu88], hierarchic route routing [Lau95],<sup>24</sup> and threshold distance vector routing [Lau95].<sup>25</sup> These have not been developed enough to understand their strengths and weaknesses.

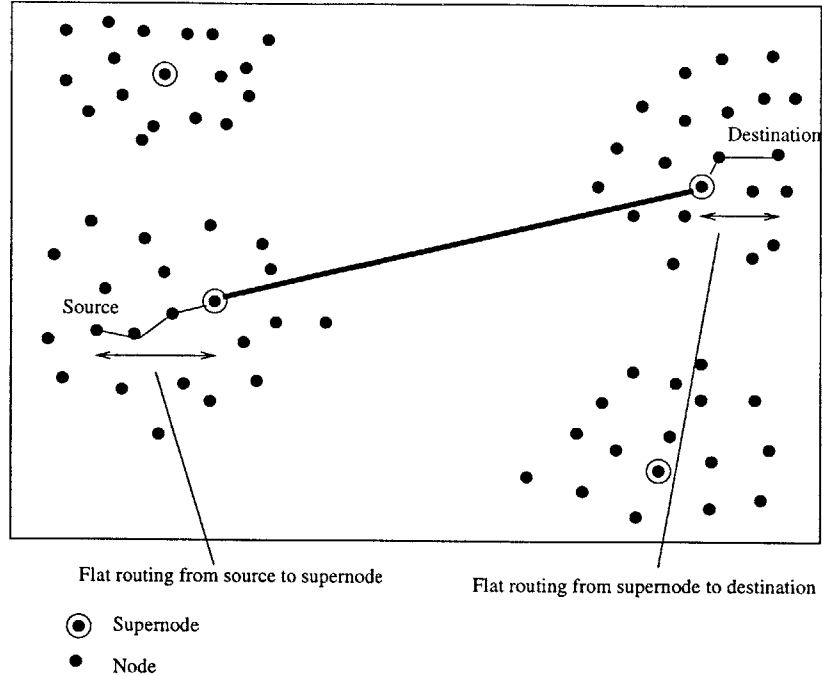
## 1.5 Problem Statement

Building low-energy PRNETs is an important design goal for both civilian and military applications of PRNETs. There are many issues involved in the design of low-energy PRNETs, so we address the problem through the lens of the minimum-energy communication problem. As argued in Section 1.2, the problem of minimum-energy communication over a PRNET is really a joint routing-scheduling-topological problem. Given the complexity of the joint problem, we propose that it be decomposed

<sup>24</sup>[Lau95] summarizes the description from [Be+89].

<sup>25</sup>[Lau95] summarizes the description from [Ste85].



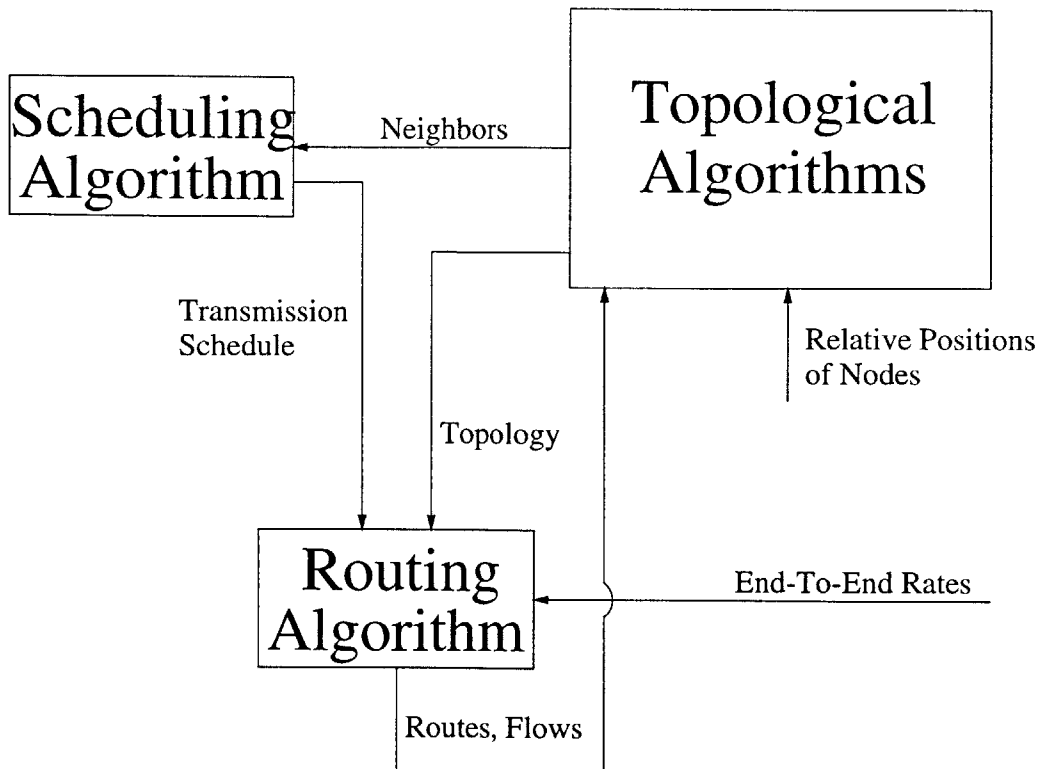


**Figure 1-5:** Routing using backbone links with isolated subnetworks.

into its three separate component problems. Each problem can then be handled by a separate algorithm or set of algorithms. The separate algorithms communicate and coordinate among each other to form a solution to the original joint problem. Fig. 1-6 summarizes the proposed decomposition. Decomposing the original joint problem very likely yields a sub-optimal solution. The degree of sub-optimality largely depends on how the original problem is decomposed.

Topological organization is an enormous problem, that is likely to involve considerations outside the realm of minimum-energy communication over a PRNET. Some of those considerations may be heuristic, including that the network be strongly connected (i.e., every node has a path to every node), that there be a path from every node to a supernode, or that certain nodes necessarily be neighbors of each other. We have already decided to have an upper bound on the number of neighbors a node has. We also want to focus on how nodes select neighbors, by studying a neighbor management procedure, related to the task of identifying potential new neighbors.

As for the scheduling problem, we want to investigate several approaches for designing the scheduling algorithm, and propose metrics for comparing the different



**Figure 1-6:** Interactions between the different algorithms.

approaches. The design of the scheduling algorithm depends on how the joint problem is decomposed. Therefore, we want to carefully investigate the different options for decomposing the problem, and select the most suitable one.

In terms of the minimum-energy PRNET routing problem, we want to extend the elegant optimal routing approach developed for minimum-delay routing in wireline networks to minimum-energy routing in PRNETs. We focus on developing an appropriate global cost function. By replacing the cost function in Eq. 1.2 with one that reflects the aggregate radiated energy in the PRNET, we can use any algorithms and insight developed for optimal-delay routing for minimum-energy routing. However, we face an immediate obstacle in that the aggregate transmission power is, in general, not a convex function of the transmission rates (see Eq. 1.1). Recall that an important component in the optimal routing approach is having a convex cost function.

We want to concentrate on studying flat routing in small-scale PRNETs because,

as pointed out in Subsection 1.4.4, this is a fundamental problem in PRNET routing. As a result, a simplifying yet reasonable assumption we make, is that we have a complete and current picture of the network topology. Finally, we want to follow the strategy of developing the routing algorithm from a centralized point of view, with the goal of developing an understanding of the critical issues and trade-off's involved in minimum-energy PRNET routing.

## 1.6 Thesis Organization

In Chapter 2 we analyze how to best separate the routing, the scheduling, and the topological organization components. We also propose several practical scheduling algorithms. In Chapter 3 we begin our investigation of minimum-energy routing using the optimal routing approach. In Chapter 3 we derive results for routing in static PRNETs, and in Chapter 4 we extend the results to routing in dynamically varying PRNETs. In Chapter 5 we propose several approaches to a neighbor management procedure and analyze their performance. Finally in Chapter 6 we summarize the main contributions of this work and outline problems for future work. There are several appendices in this thesis. They are reserved for deriving detailed results that would have been too cumbersome to derive in the main body of the thesis.

## Chapter 2

# The Scheduling Problem

The main focus of this chapter is the scheduling problem. We briefly remind the reader what the scheduling problem is. The medium access scheme for our PRNET model is slotted DS-CDMA. There is a constraint that no node can transmit and receive simultaneously, although a node may receive from several transmitters simultaneously and transmit to several receivers simultaneously (see Section 1.3). The scheduling problem is that of constructing a frame with the appropriate number of slots and scheduling which transmissions occur in which slots, such that no node transmits and receives in the same slot.

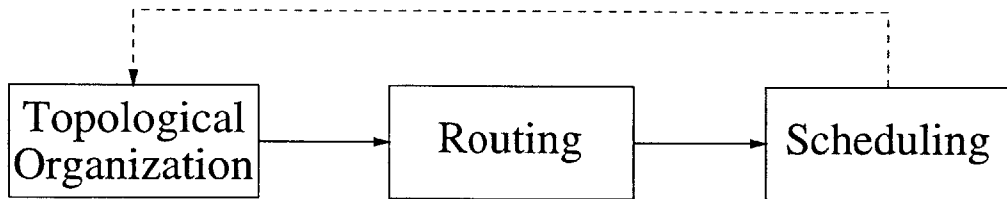
We want the scheduling algorithm to be as simple and as practical as possible, in order to increase its usefulness for the design of practical PRNETs. (Some of the scheduling algorithms proposed in the literature are difficult to implement in practice.)

Before we approach the scheduling problem, we discuss how the joint routing-scheduling-topological problem ought to be decomposed. In Section 1.2 we already argue for the assumption that topological changes in the PRNET occur at a slow enough rate, that the routing and scheduling algorithms can effectively treat the topology as being static.<sup>1</sup> We assume that the routing and scheduling algorithms solve their respective problems assuming a given static topology, such as the one

---

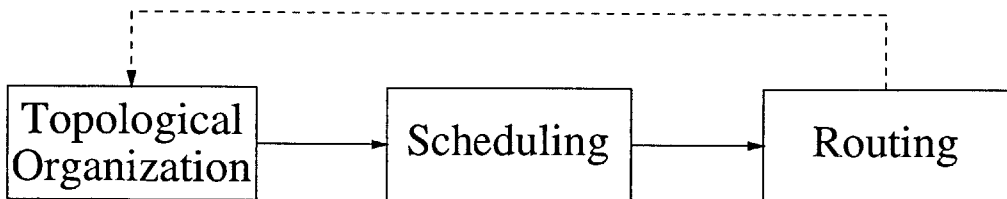
<sup>1</sup>The concept of modeling the topology as being static is treated more formally in Chapter 4.

shown in Fig. 2-1. There are three possible strategies for decomposition. The first is to have the minimum-energy routing algorithm construct the paths, along the given links, and determine the flow rates on the different paths. The scheduling algorithm then inherits the topology and the routes, and schedules transmissions such as no node transmits and receives in the same slot. This strategy is illustrated below:



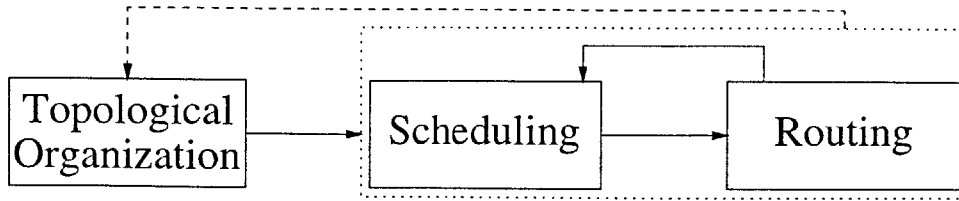
The dashed arrow indicates the potential for a slow iteration of the three problems. In other words, we envision the topological algorithms to potentially adjust the topology based in part on the prevailing solutions to the routing and scheduling problems. We find this strategy unappealing. A serious shortcoming of this strategy is that unless the minimum-energy routing algorithm knows which links are scheduled to have data sent on them simultaneously, it cannot account for the interference between signals on such links, which is important in the calculation of the transmission powers. Accurate knowledge of transmission powers will be important if the routing algorithm is to minimize aggregate energy.

The second strategy, which is the one we favor, is to reverse the order of the routing and scheduling algorithms, as illustrated below:



With this strategy, the minimum-energy routing algorithm can account for interference between transmissions. Therefore, the minimum-energy routing algorithm can try to avoid sending data on links that highly interfere with each other.

The third decomposition strategy is depicted below:



This strategy can be viewed as a hierarchical decomposition strategy, whereby the joint routing-scheduling-topological problem is decomposed into a topological organization problem and a routing-scheduling problem. The latter is then solved by yet another decomposition. The solution is approximated by a continuous iteration between a scheduling algorithm and a minimum-energy routing algorithm. However, this strategy is not preferable to us. It suggests that the scheduling algorithm makes combinatorial decisions based on adjustments of flow rates, which are continuous quantities, along different paths. This likely means that the scheduling algorithm is a complicated one, and that it may not be practical.

Therefore to summarize, we prefer to use the second of the three decomposition strategies outlined above. This implies that the goal of the scheduling algorithms we develop is to construct a frame such that the PRNET links, as given by the topology, are all *enabled* within the span of the frame. When Link  $(i, j)$  is enabled, Node  $i$  may transmit to Node  $j$ , although it does not necessarily have to. We explore two approaches for the scheduling algorithm. The first is minimum-frame-length scheduling, and the second is several types of scheduling algorithms that are simple to implement and that aim at a desirable trade-off between the length of the frame and the fraction of slots in which each link is enabled.

## 2.1 Minimum-Frame-Length Scheduling

Consider the network in Fig. 2-1. As previously mentioned, the set of links in the PRNET are already selected by topological algorithms. One feasible frame for this PRNET consists of the four slots shown in Fig. 2-2. A shorter frame with three slots, shown in Fig. 2-3, is also possible. Indeed, three is the minimum number of slots needed.

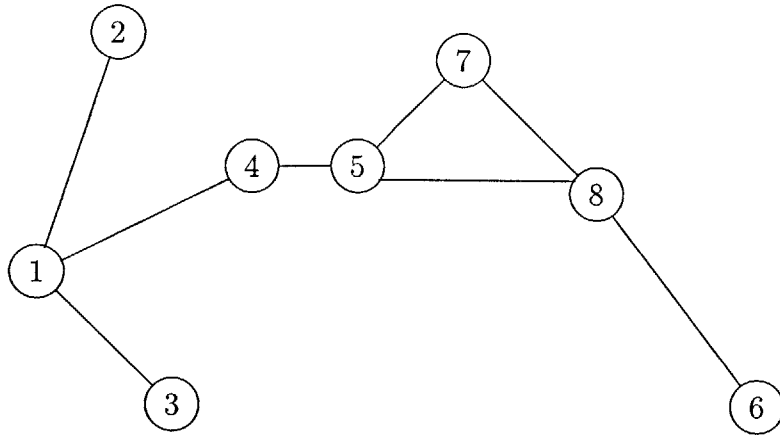


Figure 2-1: Topology of a PRNET.

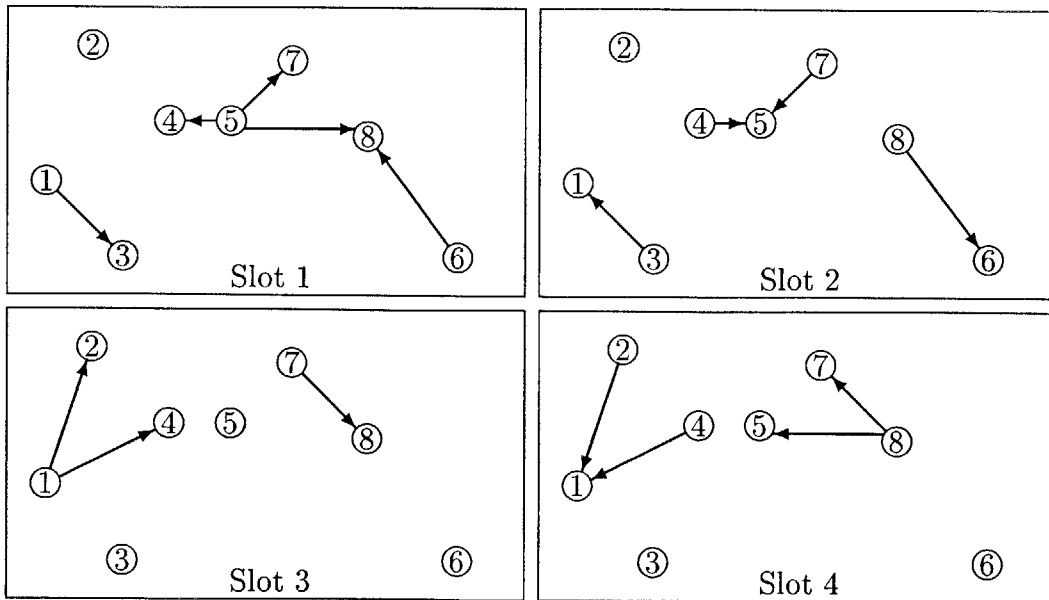
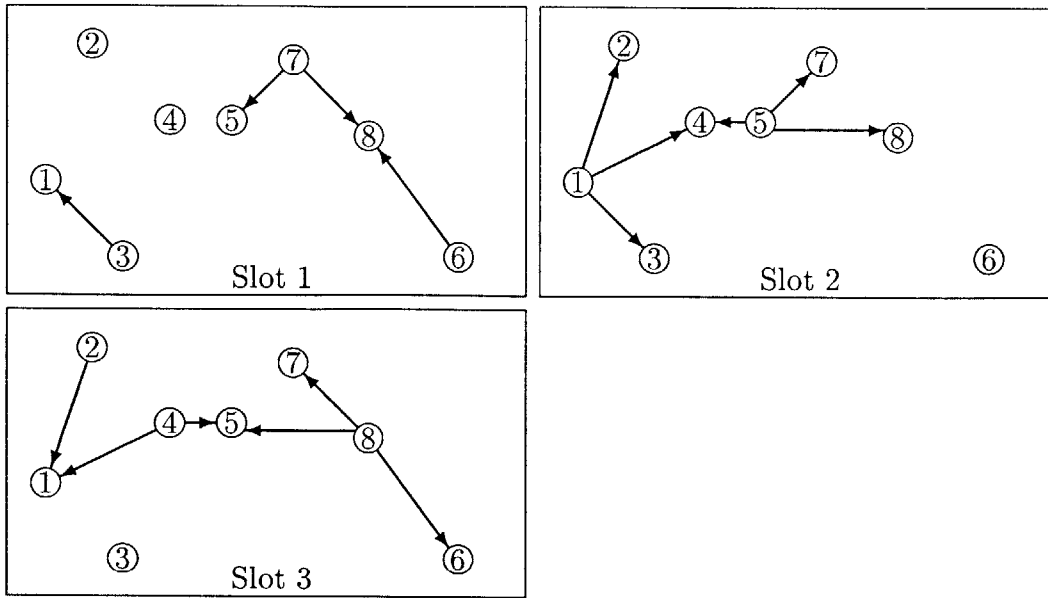


Figure 2-2: A feasible frame consisting of four slots.



**Figure 2-3:** A feasible frame consisting of minimum number of slots.

As discussed in Chapter 1 (Subsection 1.3.4), trying find to the minimum-frame-length schedule is a prevalent approach in the literature [HaS88], [RaL93], [SeH97]. Shorter frames are desirable because they improve overall network throughput and result in smaller end-to-end and inter-packet delays. The complexity of the problem of minimum-frame-length scheduling is addressed in the following theorem:

**Theorem 2.1** *The problem of Minimum-Frame-Length Scheduling is NP-Complete.*

The proof is given in Appendix E.

The NP-Completeness of MFL scheduling is of concern as it implies computational intractability for sufficiently large PRNETs. Even for small networks, MFL scheduling can be unappealing if we start to consider the difficulties associated with a distributed implementation of it. Another issue with MFL scheduling is that the length of the frame is likely to change with topological changes. Possibly non-trivial algorithms are needed to ensure that all the nodes in the network switch to the new frame at the same time.



In the next section we propose a class of scheduling algorithms that follow the general strategy of scheduling before routing, and are simpler to implement in practice.

## 2.2 Neighbor-Based Scheduling

The algorithms in this section use the upper bound on the number of neighbors of each node. Recall from Chapter 1 (Subsection 1.3.1) that if the transmission radii of nodes are too large, then transmitting nodes will create high interference in the PRNET and will themselves use high levels of transmission powers. The high interference and high power levels go against our general goal of low-energy PRNETs. We choose to enforce small transmission radii by imposing an upper limit on the number of neighbors a node can have. As we see below, knowing there is an upper limit on the number of neighbors can help us design simple scheduling algorithms.

The scheduling problem is now viewed from the perspective of allowing each node to transmit to each of its neighbors and receive from each of its neighbors, within the span of a frame. Our neighbor-based scheduling algorithms rely on associating a color with each node. In order to allow a node to communicate with its neighbors, it must have a different color than each of its neighbors (the exact connection between scheduling and colors is explained in the subsections below). In general, coloring a graph is a complex problem.<sup>2</sup> However by restricting the number of neighbors to be at most  $M$ , we know that  $M + 1$  colors are sufficient. When two nodes with the same color decide to become neighbors, without loss of generality, the node with the smaller UID (user ID) can change its color. Since that node has at most  $M$  neighbors, there are at most  $M$  colors it cannot assume. This leaves at least one color out of the  $M + 1$  which the node can assume without conflicting with the colors of its neighbors. If the two nodes that are about to become neighbors have different colors, then there will be no need for either one to change its color.

---

<sup>2</sup>In particular finding the minimum number of colors for a graph is known to be an NP-Complete problem.

Slot #	Transmissions
1	0 * → 1 *
2	1 * → 0 *
3	* 0 → * 1
4	* 1 → * 0

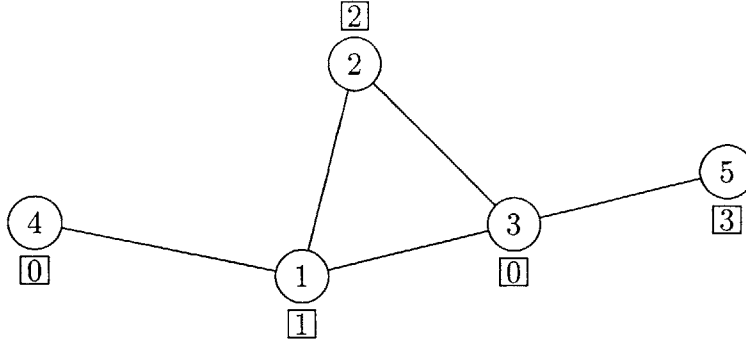
**Figure 2-4:** Scheduling using binary labeling. The \* denotes 0 or 1.

### 2.2.1 Round-Robin Scheduling

Given that each node has a different color from each of its neighbors, one possible schedule is the round-robin scheduling scheme among neighbors. The frame consists of  $M+1$  slots, and each slot is associated with a different color. In each slot, nodes of the associated color are allowed to transmit to their neighbors. Round-robin scheduling has two drawbacks. First, the length of the frame increases linearly with the number of neighbors. Second, it does not take full advantage of the DS-CDMA-enabled capability to receive from multiple neighbors. Using that capability can lead to a more efficient use of the channel bandwidth. Below we propose another neighbor-based scheduling algorithm that performs better in those two regards.

### 2.2.2 Binary Label Scheduling

In the binary label (BL) scheduling algorithm we associate with each node a binary label that is the binary representation of its color. Suppose the number of colors,  $M+1$ , is 4. Then the binary label consists of 2 digits. In the first slot of the frame, we allow the nodes whose label's first (leftmost) digit is 0 to transmit to the nodes whose first digit is 1. In the second, all the nodes whose first digit is 1 can transmit to the nodes whose first digit is 0. The process is repeated for the second digit. The entire schedule is summarized in Fig. 2-4. Note that the frame contains  $2\lceil\log_2(M+1)\rceil$  slots.



**Figure 2-5:** A simple network with five colored nodes.

We clarify the BL scheduling algorithm with an example. Consider the simple network in Fig. 2-5. In that example, the upper limit on the number of neighbors is 3. A feasible coloring of the nodes such that no two neighbors have the same color is also shown in Fig. 2-5, where the colors are 0, 1, 2, and 3. In Fig. 2-6, we show the same network, with the colors replaced by their binary representations. Note also that certain pairs of nodes now have two links between them. The number of links represents the Hamming distance between the nodes' labels.<sup>3</sup> The significance of the Hamming distance is discussed later. The BL schedule for our example is summarized in Fig. 2-4, and the resulting frame is shown in Fig. 2-7.

The binary label scheme can be generalized by labeling nodes using base  $b$  instead of base 2 (as above). For example in the case of  $b = 3$ , which we refer to as ternary label scheduling (TL scheduling), the frame contains  $3\lceil\log_3(M + 1)\rceil$  slots. Fig. 2-8 summarizes the TL schedule for  $M = 8$ . For an arbitrary base  $b$ , the frame length,  $K$ , is given by

$$K = b\lceil\log_b(M + 1)\rceil \tag{2.1}$$

---

<sup>3</sup>The Hamming distance between binary labels is the number of digits in which they differ.

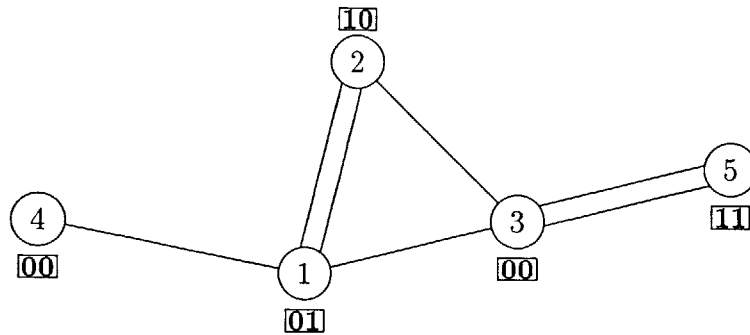


Figure 2-6: Labeling each node with a binary representation of its color.

1 → 2	2 → 1	4 → 1	1 → 2
3 → 5	2 → 3	2 → 1	1 → 4
3 → 2	5 → 3	3 → 5	1 → 3
		3 → 1	5 → 3
0* → 1*	1* → 0*	*0 → *1	*1 → *0

Figure 2-7: Frame resulting from binary label scheduling.

Slot #	Transmissions
1	0 * $\rightarrow$ {1, 2} *
2	1 * $\rightarrow$ {0, 2} *
3	2 * $\rightarrow$ {0, 1} *
4	* 0 $\rightarrow$ * {1, 2}
5	* 1 $\rightarrow$ * {0, 2}
6	* 2 $\rightarrow$ * {0, 1}

**Figure 2-8:** Scheduling using ternary labeling. The \* denotes 0, 1 or 2.  $\{a, b\}$  denotes  $a$  and  $b$ .

We now investigate whether base-2 or binary labeling (as opposed to using a different base) is optimal in terms of the resulting frame length. To simplify the mathematics, we temporarily ignore the integer constraint and say

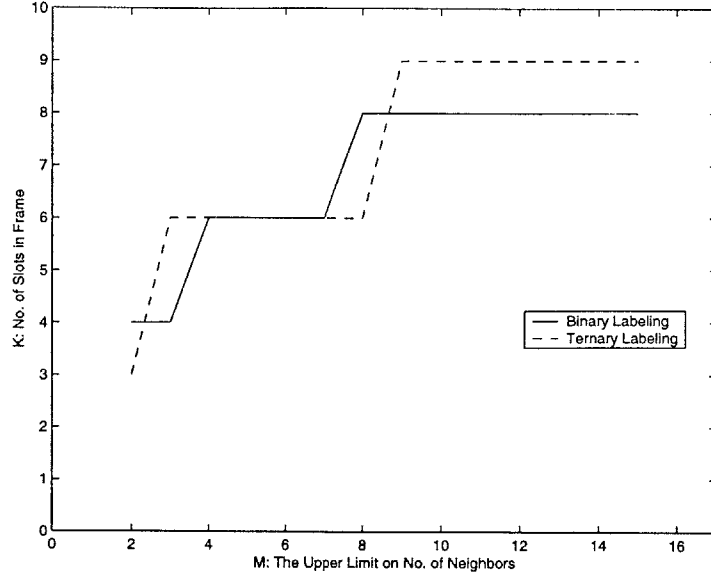
$$K = b \log_b(M + 1)$$

$K$  is minimized for  $b^* = e \approx 2.718$ . Since  $b$  needs to be an integer, this suggests that the optimal value for  $b$  is either 2 or 3. In Fig. 2-9 we plot  $K$  (as given in Eq. 2.1), for  $b = 2$  and  $b = 3$ , versus  $M$ . We restrict the range of consideration for  $M$  to  $2 \leq M \leq 15$ , as we do not anticipate the upper limit on the number of neighbors to be outside that range. As Fig. 2-9 indicates neither  $b = 2$  nor  $b = 3$  dominates over that range.<sup>4</sup> Therefore, the choice of whether to use binary or ternary labeling may depend on the value of  $M$ . Note that since every node in the PRNET adopts the same  $M$  for all time, the choice of binary labeling versus ternary labeling is made off-line.

We turn our attention back to BL scheduling, and use the example of  $M = 3$  to illustrate our next point. As seen in Fig. 2-7, a node with the label 11 transmits to any neighbor with the label 00 in two of the four slots. Let  $\rho_{ij}$  be the fraction

---

<sup>4</sup>It is only for  $M \geq 2.65 \times 10^8$  that one of the two, namely  $b = 3$ , dominates throughout.



**Figure 2-9:** Comparison between binary labeling and ternary labeling for  $2 \leq M \leq 15$ .

of slots in which  $(i, j)$  is enabled. Given that links may be enabled in multiple slots within the frame, hence potentially increasing the throughput of those links, it seems reasonable to have the  $\rho_{ij}$ 's form an additional figure of merit in conjunction with  $K$ , the number of slots in the frame. However in BL and TL scheduling, the  $\rho_{ij}$ 's are not uniform for all pairs of labels. For instance in the BL scheduling, 11 transmits to 01 in only one of the four slots.

It is of interest to have all the  $\rho_{ij}$ 's be uniform, because we do not want certain links to have potentially higher throughput than other links by virtue of having higher  $\rho_{ij}$ 's. Although some links are more heavily utilized than others, according to our adopted strategy routing is yet to be performed. Therefore, at the scheduling stage, it is not known which links are more heavily utilized than others. In addition, the value of  $\rho_{ij}$  depends on the relative coloring of Node  $i$  and Node  $j$ . The coloring scheme described above is simple in part because the colors are assigned to nodes in a somewhat arbitrary fashion. If we want to make certain links have higher  $\rho_{ij}$ 's than others, then their end nodes will have to be colored in a deliberate manner. This will complicate the coloring of the nodes, which in turn will complicate the scheduling algorithm.

We can “artificially” make all the  $\rho_{ij}$ 's equal. For example, we can have it such

that the pair 00 and 11 “pretend” to be the pair 00 and 01 for transmissions between them, and maintain their real labels for other transmissions. If we make the  $\rho_{ij}$ ’s uniform by this method, then the minimum of the  $\rho_{ij}$ ’s will dictate what the uniform  $\rho$  is. In other words,

$$\rho = \min_{(i,j)} \rho_{ij} = \rho_{\min}$$

An alternative method to making  $\rho_{ij}$ ’s uniform is to add redundant bits to the labels. The hope then is that the redundancy allows nodes to pick labels that inherently have uniform  $\rho_{ij}$ ’s. Consider again the case  $M = 3$ . If we use 3-digit labels, instead of 2-digit labels, then we will have eight possible labels. For  $M = 3$  we only need four. Suppose we select the following four 3-digit labels:

$$\begin{array}{cc} 000 & 011 \\ 110 & 101 \end{array}$$

Constructing the frame in the same manner as before (i.e.,  $0** \rightarrow 1**, 1** \rightarrow 0**, \text{etc.}$ ), we end up with a frame of length 6, with  $\rho_{ij}$  being uniformly equal to  $\frac{1}{3}$ . Note that in BL scheduling,  $\rho_{\min} = \frac{1}{4}$ . ( $\rho_{\max} = \frac{1}{2}$ .)

There are two points that we need to take from the above discussion. First, in conjunction with  $K$ , the number of slots in the frame, we perhaps ought to consider  $\rho_{\min}$  as another figure of merit, although one might be considered more important than the other. Second, the concept of adding redundant bits is reminiscent of error-correcting coding theory. Error-correcting codes add extra bits to the information bits prior to transmission so that the receiver may potentially find and correct any bit errors.<sup>5</sup> The field of coding theory is rich with many families of codes [Bla99]. Below we investigate two families of codes that have promising uniformity properties, namely the family of *bi-orthogonal codes* and the family of *simplex codes*.

### 2.2.3 Bi-Orthogonal Binary Label Scheduling

Bi-orthogonal codes are also known as Walsh codes and first-order Reed-Muller codes. They are  $(2^{m-1}, m)$  codes. This means that bi-orthogonal coding maps  $m$ -bit binary

---

<sup>5</sup>When bits are transmitted over a communication channel, some can get lost or corrupted.

strings to codewords of length  $2^{m-1}$  bits. In bi-orthogonal BL scheduling, we essentially map the  $m$ -bit labels of the BL scheduling algorithm to their corresponding bi-orthogonal codewords. We refer to the corresponding bi-orthogonal codewords as bi-orthogonal labels.

We mentioned previously that we do not expect the upper limit on the number of neighbors to fall outside the range  $2 \leq M \leq 15$ . Accordingly, we are interested in mapping 2-bit, 3-bit, and 4-bit binary labels (i.e., binary representation of the colors) into bi-orthogonal codewords. The bi-orthogonal code is a linear block code that can be summarized by a generator matrix. For instance, a generator matrix for the  $(4, 3)$  bi-orthogonal code is

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

A 3-bit binary string is mapped into its corresponding bi-orthogonal codeword, by treating the former as a row vector, and left-multiplying it with the generator matrix (using modulo-2 addition and multiplication). For example, 101 maps to 1010 in the following manner:

$$\begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$$

The eight codewords in the  $(4, 3)$  bi-orthogonal code are

0000 1111  
0101 1010  
0011 1100  
0110 1001

We listed the codewords above in such a way to emphasize that each codeword has a complement. Furthermore, we note that the Hamming distance between any codeword and another non-complement codeword is uniformly equal to 2. These two



observations generalize for all bi-orthogonal codes: given a  $(2^{m-1}, m)$  bi-orthogonal code, each codeword has a complement with which the Hamming distance is  $2^{m-1}$ . The Hamming distance between a codeword and non-complement codewords is  $2^{m-2}$ . Furthermore, with the exception of the all-zero and the all-one codewords, each bi-orthogonal codeword is half ones and half zeros.

The bi-orthogonal BL scheduling scheme uses a frame of length  $2 \times 2^{m-1} = 2^m$  where  $m = \lceil \log_2(M + 1) \rceil$ . ( $M$  is the upper limit on the number of neighbors, and  $m$  is the number of digits in the binary representation of  $M + 1$  different colors.) As in the BL scheduling algorithm, nodes with bi-orthogonal labels starting with 0 in the first slot are allowed to transmit to the nodes with bi-orthogonal labels starting with 1. In the second slot, nodes with labels starting with 1 can transmit to the nodes with labels starting with 0. The process is repeated for the remaining  $2^{m-1} - 1$  digits. Such a scheduling scheme yields

$$\rho_{\min} = \frac{2^{m-2}}{2^m} = \frac{1}{4}$$

and as mentioned earlier

$$K = 2^m$$

This is not the entire story for bi-orthogonal BL scheduling. If  $M = 4$ , then  $m = \lceil \log_2 5 \rceil = 3$ . This implies that for  $M = 4$  we need to use five 4-bit bi-orthogonal codewords, out of the possible eight. Suppose we select the following five:

0101  
1010  
0110  
1001  
1100

Then we can use a variant of the bi-orthogonal BL schedule, in which the frame consists of four slots (instead of eight previously). In the first slot, nodes with labels starting with 1 are allowed to transmit to nodes with labels starting with 0. In the

Slot #	Transmissions								
1	1	*	*	*	→	0	*	*	*
2	*	1	*	*	→	*	0	*	*
3	*	*	1	*	→	*	*	0	*
4	*	*	*	1	→	*	*	*	0

**Figure 2-10:** A variant bi-orthogonal BL schedule for  $M = 4$  and  $M = 5$ .

second slot, nodes whose labels' second digit is 1 are allowed to transmit to nodes with a second digit of 0. The process is repeated for the remaining digits. Fig. 2-10 summarizes the variant bi-orthogonal schedule in the case  $M = 4$  (provided neither the all-zero or all-one bi-orthogonal codeword is employed). The following lemma helps us with the analysis of  $\rho_{\min}$  for the variant scheduling algorithm:

**Lemma 2.1** *Consider a  $(2^{m-1}, m)$  bi-orthogonal code. Let  $c_1$  and  $c_2$  be any two codewords in that code, such that they are not complements of each other, and also such that neither is the all-zero or the all-one codeword. The number of 1's in  $c_1$  that overlap with 0's in  $c_2$  is  $2^{m-3}$ .*

**Proof:** Let  $x$  be the number of 1's in each codeword.  $x = 2^{m-2}$ . Let  $y_1$  be the number of 1's in  $c_1$  that overlap with 0's in  $c_2$ . Similarly, let  $y_2$  be the number of 0's in  $c_1$  that overlap with 1's in  $c_2$ . Without any loss, we rearrange the bit positions in the two codewords into four blocks, representing the four different combinations of overlaps, as shown below:

	$y_1$	$y_2$	$x - y_1$	$x - y_2$
$c_1$	1 1 ... 1 1	0 0 ... 0 0	1 1 ... 1 1	0 0 ... 0 0
$c_2$	0 0 ... 0 0	1 1 ... 1 1	1 1 ... 1 1	0 0 ... 0 0
$c_1 \oplus c_2$	1 1 ... 1 1	1 1 ... 1 1	0 0 ... 0 0	0 0 ... 0 0

We also show the bit-wise modulo-2 addition of the two codewords, namely  $c_1 \oplus c_2$ .

The bi-orthogonal block code is a linear code, implying that the sum,  $c_1 \oplus c_2$ , is also a bi-orthogonal codeword. By assumption,  $c_1$  and  $c_2$  are not complements, and neither is all-zero or all-one. Therefore, the number of 1's in  $c_1 \oplus c_2$  is also  $x$  implying,

$$x = y_1 + y_2 \tag{2.2}$$

Moreover, since the number of 1's in  $c_2$  is also  $x$ , we have

$$x = y_2 + x - y_1 \tag{2.3}$$

Solving Eqs. 2.2 and 2.3 yields,

$$y_1 = y_2 = \frac{x}{2} = \frac{2^{m-2}}{2} = 2^{m-3}$$

**Q.E.D.**

Therefore, in the variant bi-orthogonal BL scheduling algorithm

$$\rho_{\min} = \frac{2^{m-3}}{2^{m-1}} = \frac{1}{4}$$

and

$$K = 2^{m-1}$$

The  $\rho_{\min}$  is the same as in the proper bi-orthogonal BL scheduling algorithm. The advantage of the variant bi-orthogonal BL scheduling algorithm is that it results in a shorter frame. That advantage can also be realized in the case  $M = 5$ . However, for  $M = 6$  and  $M = 7$  we are forced to use at least one of the bi-orthogonal codewords 0000 and 1111, which prevents us from using the variant bi-orthogonal BL scheduling algorithm (0000 would be always receiving and 1111 would be always transmitting). Similarly, the variant bi-orthogonal BL scheduling algorithm can be applied to  $8 \leq M \leq 13$ , but not to  $M = 14$  or  $M = 15$ .

One final note about bi-orthogonal codes is that for  $m = 2$ , the 2-bit strings are essentially mapped to themselves. In a way, this implies that there is no bi-orthogonal

code for  $m = 2$ .

### 2.2.4 Simplex Binary Label Scheduling

The simplex code is a derivative of the bi-orthogonal code. One way to view the derivation, is to take a  $(2^{m-1}, m)$  bi-orthogonal code, remove all the codewords whose first (leftmost) digit is 1, and strip the first digit (which is a 0) from the remaining codewords. For example the  $(3, 2)$  simplex code can be derived from the  $(4, 3)$  bi-orthogonal code in the following manner:

0000	→	0000	→	000
1111	→	×		
0101	→	0101	→	101
1010	→	×		
0011	→	0011	→	011
1100	→	×		
0110	→	0110	→	110
1001	→	×		

Note that in the simplex code there is no all-one codeword.

To generalize then, the simplex code is a  $(2^m - 1, m)$  code. The Hamming distance between *any* two codewords is *uniformly*  $2^{m-1}$ . From the above description of deriving simplex codes, we note that in a non-zero simplex codeword, the number of 1's is  $2^{m-1}$ , and the number of zeros is  $2^{m-1} - 1$ .

As in the bi-orthogonal BL scheduling algorithm there is a proper simplex BL scheduling algorithm and a variant of it. In the proper simplex BL scheduling algorithm, the frame consists of  $2(2^m - 1)$  slots, where  $m = \lceil \log_2(M + 1) \rceil$ . As before, for each digit there are two slots. In the first, the 0's are allowed to transmit to the 1's. In the second, the 1's are allowed to transmit to the 0's. To summarize, in the

proper simplex BL scheduling algorithm,

$$\rho_{\min} = \frac{2^{m-1}}{2(2^m - 1)} = \frac{2^{m-2}}{2^m - 1} > \frac{1}{4}$$

and

$$K = 2(2^m - 1) \leq 2M$$

We emphasize that because the Hamming distance is uniform between all the codewords,  $\rho_{ij} = \rho_{\min}$  for all  $(i, j)$ . *None* of the other scheduling algorithms (except round-robin) has that property.

The variant simplex BL scheduling algorithm is based on the following lemma:

**Lemma 2.2** *Consider a  $(2^m - 1, m)$  simplex code. Let  $c_1$  and  $c_2$  be any two codewords in that code, such that neither is the all-zero codeword. The number of 1's in  $c_1$  that overlap with 0's in  $c_2$  is  $2^{m-2}$ .*

The proof is almost identical to that for Lemma 2.1, with the difference being that  $x$ , the number of 1's in  $c_1$  and  $c_2$  is  $2^{m-1}$ . We leave it to the reader to prove Lemma 2.2.

If  $M + 1$  is not a power of two, then we will have flexibility in choosing  $M + 1$  simplex codewords out of  $2^m$ , where  $m = \lceil \log_2(M + 1) \rceil$ . As long as we do not use the all-zero codeword, we can implement a schedule with a frame of length  $2^m - 1$  (i.e., one slot for each digit). In the first slot of the frame, nodes with a first digit of 1 are allowed to transmit to nodes with a first digit of 0. In the second, nodes with a second digit of 1 are allowed to transmit to nodes with a second digit of 0. The process is repeated for the remaining digits. Fig. 2-11 summarizes a variant simplex BL schedule that can be employed for  $M = 4, 5$ , and 6.

Lemma 2.2 implies that  $\rho_{ij}$  is uniformly equal to  $\frac{2^{m-2}}{2^m - 1}$ , which is the same as in the proper simplex BL scheduling algorithm. Therefore in the variant simplex BL scheduling algorithm

$$\rho_{\min} = \frac{2^{m-2}}{2^m - 1}$$

Slot #	Transmissions															
1	1	*	*	*	*	*	*	*	→	0	*	*	*	*	*	*
2	*	1	*	*	*	*	*	*	→	*	0	*	*	*	*	*
3	*	*	1	*	*	*	*	*	→	*	*	0	*	*	*	*
4	*	*	*	1	*	*	*	*	→	*	*	*	0	*	*	*
5	*	*	*	*	1	*	*	*	→	*	*	*	*	0	*	*
6	*	*	*	*	*	1	*	*	→	*	*	*	*	*	0	*
7	*	*	*	*	*	*	1	*	→	*	*	*	*	*	*	0

Figure 2-11: A variant simplex schedule for  $M = 4, 5,$  and  $6$ .

and

$$K = 2^m - 1$$

### 2.2.5 Summary of Neighbor-Based Scheduling

In this section we proposed several neighbor-based scheduling algorithms. We suggested two figures of merit to compare among them, namely  $K$ , the number of slots in the frame, and  $\rho_{\min}$ , the minimum fraction of slots in which a node transmits to another node. The preference is for a smaller value of  $K$ , and a larger value of  $\rho_{\min}$ . Table 2.1 compares the different neighbor-based scheduling algorithms for different values of  $M$ . Unfortunately as one might expect, there is a trade-off between  $K$  and  $\rho_{\min}$ , such that for no value of  $M$  is one scheduling algorithm clearly the best. The choice of the scheduling algorithm depends on the relative importance of small values of  $K$  versus large values of  $\rho_{\min}$ , and may need to be determined through experimentation.

If  $K$  is the only parameter of interest, then one might simply find MFL schedules for  $M + 1$  nodes by exhaustive search. We tried to find MFL schedules for some values of  $M$  through exhaustive search, but found that at least one of the proposed systematic scheduling schemes listed in Table 2.1 matched the best result of the

M	Scheduling Algorithm	K	$\rho_{\min}$	
2	Round-Robin	3	1/3	•
	Binary Labeling	4	1/4	
	Ternary Labeling	3	1/3	•
	Bi-Orthogonal BL	4	1/4	
3	Simplex BL $\vee$	3	1/3	•
	Round-Robin	4	1/4	•
	Binary Labeling	4	1/4	•
	Ternary Labeling	6	1/6	
4	Bi-Orthogonal BL	4	1/4	•
	Simplex BL	6	1/3	•
	Round-Robin	5	1/5	
	Binary Labeling	6	1/6	
5	Ternary Labeling	6	1/6	
	Bi-Orthogonal BL $\vee$	4	1/4	•
	Simplex BL	7	2/7	•
	Round-Robin	6	1/6	
6	Binary Labeling	6	1/6	•
	Ternary Labeling	6	1/6	•
	Bi-Orthogonal BL	8	1/4	
	Simplex BL $\vee$	7	2/7	•
7	Round-Robin	8	1/8	
	Binary Labeling	6	1/6	•
	Ternary Labeling	6	1/6	•
	Bi-Orthogonal BL	8	1/4	•
8	Simplex BL	14	2/7	•
	Round-Robin	9	1/9	
	Binary Labeling	8	1/8	•
	Ternary Labeling	6	1/6	•
9	Bi-Orthogonal BL $\vee$	8	1/4	•
	Simplex BL $\vee$	15	4/15	•
	Round-Robin	10	1/10	
	Binary Labeling	8	1/8	
10	Ternary Labeling	9	1/9	
	Bi-Orthogonal BL $\vee$	8	1/4	•
	Simplex BL $\vee$	15	4/15	•
	Round-Robin	11	1/11	
11	Binary Labeling	8	1/8	
	Ternary Labeling	9	1/9	
	Bi-Orthogonal BL $\vee$	8	1/4	•
	Simplex BL $\vee$	15	4/15	•
12	Round-Robin	12	1/12	
	Binary Labeling	8	1/8	
	Ternary Labeling	9	1/9	
	Bi-Orthogonal BL $\vee$	8	1/4	•
13	Simplex BL $\vee$	15	4/15	•
	Round-Robin	13	1/13	
	Binary Labeling	8	1/8	
	Ternary Labeling	9	1/9	
14	Bi-Orthogonal BL $\vee$	8	1/4	•
	Simplex BL $\vee$	15	4/15	•
	Round-Robin	14	1/14	
	Binary Labeling	8	1/8	•
15	Ternary Labeling	9	1/9	
	Bi-Orthogonal BL	16	1/4	•
	Simplex BL $\vee$	15	4/15	•
	Round-Robin	16	1/16	
15	Binary Labeling	8	1/8	•
	Ternary Labeling	9	1/9	
	Bi-Orthogonal BL	16	1/4	•
	Simplex BL	30	4/15	•

**Table 2.1:** Comparison of different neighbor-based scheduling algorithms for different values of  $M$ .  $\vee$  indicates that the variant version of the algorithm is used.  $\bullet$ 's are used to identify the better scheduling algorithms in terms of the performance measures  $K$  and  $\rho_{\min}$ .

exhaustive search.

One scheduling algorithm which is comparable to the BL scheduling algorithms is the scheduling algorithm proposed in [She96] and summarized in Section 1.3. In that algorithm  $\rho_{ij} = \frac{1}{4}$ , for all  $(i, j)$ . However in that algorithm, the slots are not organized into frames, implying less controllable and higher variance end-to-end delay than if there is a frame structure.

## 2.3 Summary

We considered three strategies for decomposing the joint routing-scheduling-topological problem. The most appealing strategy is to first solve the topological organization problem, then have the scheduling algorithm schedule the transmissions on all the links, and then have the routing algorithm determine the paths and flow rates based on the topology and the schedule. This strategy admits the possibility of the topological algorithms adjusting the topology based on the prevailing solutions to the scheduling and routing problems. Based on this strategy, we proposed several scheduling algorithms that use the upper limit on the number of neighbors. We introduced two metrics to compare them. Each of the proposed scheduling algorithms is simple and places little computational burden on the network.



# Chapter 3

## Minimum-Energy Routing in Static PRNETs

In Chapter 2 we propose that a good decomposition strategy for the joint routing-scheduling-topological problem is one in which the minimum-energy routing algorithm solves the routing problem based on topology and schedule, as inherited from the topological organization algorithms and the scheduling algorithm. In this chapter, we focus our attention on the minimum-energy routing component, and in particular investigate how to extend the optimal routing approach described in Section 1.4 to minimum-energy routing in PRNETs. We note that the schedule that the minimum-energy routing algorithm inherits need not be produced by any of the scheduling algorithms proposed in the previous chapter.<sup>1</sup> In this chapter we study routing for static PRNETs in order to develop as much insight as possible, without getting bogged down in issues related to the mobility of the nodes and the changing characteristics of the radio channel. We defer routing for dynamically varying PRNETs to Chapter 4.

In Section 1.5 we point out that one of the obstacles in trying to apply the optimal routing approach is that we need to have a convex cost function. The aggregate transmission power, as a function of the transmission rates, is not convex in general.

---

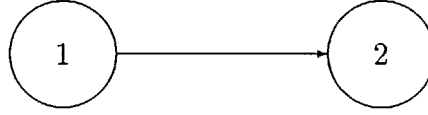
<sup>1</sup>As long as the schedule consists of identical recurring frames.

In this chapter we find a way to apply the optimal routing approach to minimum-energy PRNET routing. However, this application has several facets that need to be developed gradually. First we start by developing a meaningful convex cost function (for power) for a simple network where scheduling is not required. We then incorporate the need for scheduling and consider how to construct the cost function in the presence of slots. Finally we address how to incorporate delay into minimum-energy routing, resulting in what we refer to as minimum-combined-energy-and-delay routing (MCED routing).

### 3.1 Completing The Model

In this section we complete the model presented in Section 1.3, to provide us with the full PRNET model to be used in this chapter. First, the nodes are assumed to be immobile. Second,  $\alpha_{ij}$ , the path propagation loss from a Node  $i$  to a Node  $j$ , is assumed to be constant (although not generally equal for all pairs of nodes). The two assumptions together imply that power control is needed only up to the point where the transmission powers are adjusted to the levels that meet the  $E_b/I_0$  requirements with equality. In Appendix B (Section B.4.1) we show, using results from [Yat95], that starting with an arbitrary feasible set of transmission powers, the nodes can indeed converge to the set of power levels that meet each  $E_b/I_0$  threshold requirement with equality. Given that the end-to-end rates are constant, the two assumptions above also imply that if each node settles on a set of neighbors, there will be no need to change that set. Therefore, the links in the PRNET are assumed to be fixed, implying that the overall topology of the PRNET is fixed.

In Chapter 1, we briefly describe how transmission powers for different signals can be determined. Here we elaborate on that description. We give three simple examples that show how to construct equations, based on meeting the  $E_b/I_0$  threshold requirements, that can be solved to yield the required transmission power levels for the different signals.



**Figure 3-1:** Transmission with one transmitter and one receiver.

**Example 3.1** We look at the trivial network in Fig. 3-1. Node 1 is transmitting a DS-CDMA signal to Node 2. Let the transmission rate for this transmission be  $c_{12}$ . The energy-per-bit for that signal is

$$E_b = \alpha_{12}P_{12}/c_{12}$$

Let the required  $E_b/N_0$  threshold for this communication be  $\gamma_{12}$ . Then the power of the signal transmitted by Node 1 must satisfy the following condition:

$$\frac{\alpha_{12}P_{12}/c_{12}}{N_0} \geq \gamma_{12}$$

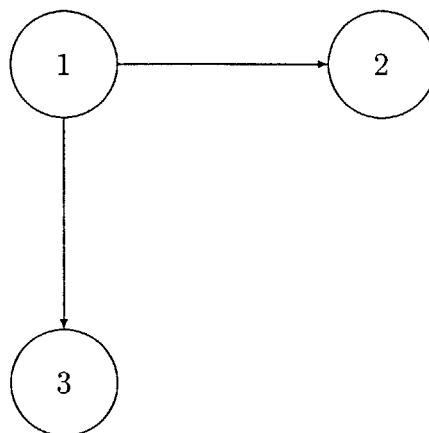
which implies the transmission power  $P_{12}$  needs to satisfy

$$P_{12} \geq \frac{\gamma_{12}N_0}{\alpha_{12}}c_{12}$$

Since it is of interest to conserve transmission energy, Node 1 wants to minimize the power of the signal it transmits, setting it to

$$P_{12} = \frac{\gamma_{12}N_0}{\alpha_{12}}c_{12}$$

**Example 3.2** In Fig. 3-2 we have one transmitting node, Node 1, and two receiving nodes, 2 and 3, with Node 1 transmitting simultaneously to both receivers. As



**Figure 3-2:** Transmission with one transmitter and two receivers.

discussed in Appendix B, we assume that signals from the same source can be made orthogonal everywhere. Therefore we can ignore their interference at the receiver. Accordingly, in this example, to meet the  $E_b/N_0$  threshold requirements, the following conditions must be satisfied,

$$P_{12} \geq \frac{\gamma_{12}N_0}{\alpha_{12}}c_{12} \quad \text{and} \quad P_{13} \geq \frac{\gamma_{13}N_0}{\alpha_{13}}c_{13}$$

Again, in the interest of conserving transmission energy, Node 1 wants to set the transmission powers to be as small as possible, namely

$$P_{12} = \frac{\gamma_{12}N_0}{\alpha_{12}}c_{12} \quad \text{and} \quad P_{13} = \frac{\gamma_{13}N_0}{\alpha_{13}}c_{13}$$

Let  $P_T$  be the aggregate transmission power, then

$$P_T = \frac{\gamma_{12}N_0}{\alpha_{12}}c_{12} + \frac{\gamma_{13}N_0}{\alpha_{13}}c_{13}$$

**Example 3.3** Next we look at the case of two transmitters and one receiver, as shown in Fig. 3-3. Since the two signals are from different transmitters, they interfere with

each other at the receiver. We assume that the spreading code sequences chosen by the two transmitters are independent, and the carrier phase for each transmitter is independent. Thus it is reasonable to assume that the signal from Node 2 to Node 3 appears as AWGN to the signal from Node 1 to Node 3, and vice versa (see Appendix B). To meet the required  $E_b/I_0$  threshold,<sup>2</sup> the power transmitted by Node 1 must then obey the following condition

$$\frac{\alpha_{13}P_{13}/c_{13}}{N_0 + \frac{\alpha_{23}P_{23}}{W}} \geq \gamma_{13} \quad (\text{C-3.1})$$

where  $W$  is the bandwidth over which the signals are spread. By symmetry, the following condition must also be satisfied

$$\frac{\alpha_{23}P_{23}/c_{23}}{N_0 + \frac{\alpha_{13}P_{13}}{W}} \geq \gamma_{23} \quad (\text{C-3.2})$$

As in Examples 3.1 and 3.2, Conditions C-3.1 and C-3.2 will need to be met with equality if the aggregate transmission power is to be minimized while satisfying the transmission rates  $c_{13}$  and  $c_{23}$ . It is not as obvious, as it is in Examples 3.1 and 3.2, why satisfying the conditions with equality results in minimizing the aggregate transmission power. The reason is discussed in Appendix B (Section B.4.1). There we also note that for a general PRNET, to minimize the aggregate transmission power, the  $E_b/I_0$  threshold requirements need to be met with equality.

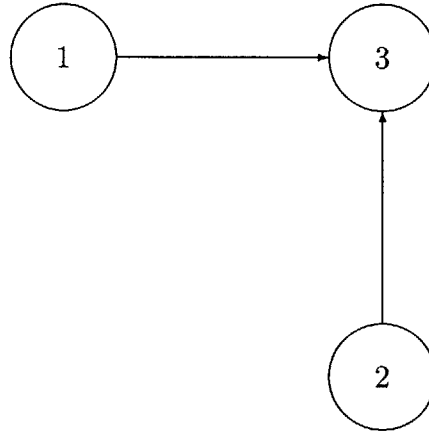
Solving for the powers in this example, we get

$$P_{13} = \frac{c_{13}\gamma_{13}WN_0(W + \gamma_{23}c_{23})}{\alpha_{13}(W^2 - \gamma_{13}\gamma_{23}c_{13}c_{23})}$$

$$P_{23} = \frac{c_{23}\gamma_{23}WN_0(W + \gamma_{13}c_{13})}{\alpha_{23}(W^2 - \gamma_{13}\gamma_{23}c_{13}c_{23})}$$

---

<sup>2</sup>Note that we are using  $I_0$  instead of  $N_0$  to emphasize that there is interference as well as thermal noise.



**Figure 3-3:** Transmission with two transmitters and one receiver.

yielding an aggregate transmission power of

$$P_T = \frac{c_{13}\gamma_{13}WN_0(W + \gamma_{23}c_{23})}{\alpha_{13}(W^2 - \gamma_{13}\gamma_{23}c_{13}c_{23})} + \frac{c_{23}\gamma_{23}WN_0(W + \gamma_{13}c_{13})}{\alpha_{23}(W^2 - \gamma_{13}\gamma_{23}c_{13}c_{23})} \quad (3.1)$$

In Appendix C we show that  $P_T$  is not a convex function of  $c_{13}$  and  $c_{23}$ .

### 3.1.1 Summary of Interference

Interference is seen as the major complication in analyzing minimum-energy PRNET routing. In order to handle interference we need to be able to model it. The three simple examples above provide the building blocks for modeling the interference in any PRNET, through the construction of governing equations (i.e., equations dictated by meeting the required  $E_b/I_0$  thresholds with equality). Solving these equations to get the aggregate transmission power may not be easy.

A few details are not incorporated in the examples above. First, strictly speaking the threshold  $\gamma_{ij}$  is a function of  $c_{ij}$ , (for a given error rate). Second, as discussed in Appendix B, signals emanating from the same transmitter are not completely orthogonal at a receiver, and therefore can create some interference for each other.

Finally, the  $\alpha_{ij}$ 's are generally not fixed and the power levels need to be continually adjusted as they change. We ignore the first two points for now. Later in this chapter and in Chapter 4 we argue that ignoring them does not decrease insight into the minimum-energy routing problem. Finally, we address the issue of non-static  $\alpha_{ij}$ 's in Chapter 4.

In Example 3.3, we show that, in general, the aggregate transmission power is not a convex function of the transmission rates. This is a major obstacle to contend with in extending optimal routing results to minimum-energy routing, which is what we want to do. Having a convex global cost function is an important part of the optimal routing approach.

## 3.2 Constructing A Convex Cost Function

We begin our investigation by revisiting the simple network shown in Fig. 3-3. Assuming  $\gamma_{13} = \gamma_{23} = \gamma$ , from Eq. 3.1,

$$\begin{aligned} P_T &= P_{13} + P_{23} \\ &= \frac{c_{13}\gamma N_0 W(W + c_{23}\gamma)}{\alpha_{13}(W^2 - \gamma^2 c_{13}c_{23})} + \frac{c_{23}\gamma N_0 W(W + c_{13}\gamma)}{\alpha_{23}(W^2 - \gamma^2 c_{13}c_{23})} \end{aligned}$$

For emphasis we remind the reader that  $P_T$  is not convex as a function of  $c_{13}$  and  $c_{23}$ .

The key observation that allows us to construct a convex cost function for minimum-energy routing is that  $c_{13}$  and  $c_{23}$  are the nominal data rates. The nominal data rate is the rate at which the transmitter sends bits *while* transmitting. This is in contrast to the average bit-rate, which is the total number of bits transmitted over a period of time, divided by that period of time. This distinction between nominal data rate and average bit-rate is important in the formulation of the optimal-delay routing problem for wireline packet-switched networks (see Section 1.4). There, the average bit-rate on  $(i, j)$  (the link from Node  $i$  to Node  $j$ ) is referred to as the flow rate,  $f_{ij}$ , and the

cost associated with the Link  $(i, j)$  is approximated as

$$D_{ij} = \frac{f_{ij}}{c_{ij} - f_{ij}}$$

According to the above expression, if the average bit-rate, which we henceforth refer to as flow rate,  $f_{ij}$ , is equal to the nominal data rate,  $c_{ij}$ , then the average number of packets waiting to be transmitted on  $(i, j)$  will go to infinity, causing a buffer overflow. Since the PRNET is also a packet-switched network, we want to ensure that we make the distinction between the flow rate and the nominal data rate, and that the flow rate is indeed less than the nominal data rate.

Initially, the difference between the nominal data rate and the flow rate in PRNETs was lost on us. In retrospect, this difference appears simple and obvious, especially since it is rather clear in minimum-delay routing. Part of the reason for missing it is that in wireline networks that distinction is more clear-cut. The links and their nominal data rates are thought of as hard-wired in point-to-point networks. In contrast, in PRNETs the nominal data rate of a link can be adjusted by the node. Therefore, when thinking about satisfying the end-to-end rate requirements, it is easy to confuse adjusting the nominal data rates as opposed to the flow rates. In Section 3.4 we discuss how making the distinction between flow rates and nominal data rates amounts to another form of decomposition, this time between routing and power control.

In any case, we still need to explain why focusing on flow rates instead of nominal data rates allows us to construct a desirable convex cost function. Assume the flow rate on  $(i, j)$  is  $f_{ij}$ , while the nominal data rate is  $c_{ij}$ . Then  $\phi_{ij}$ , the fraction of time that Node  $i$  transmits to Node  $j$ , is such that

$$\begin{aligned} f_{ij} &= \phi_{ij} c_{ij} \\ \Rightarrow \phi_{ij} &= \frac{f_{ij}}{c_{ij}} \end{aligned}$$

The important point here is that a node transmits only a fraction of the time,  $\phi_{ij}$ ,



which depends on the relative values of  $f_{ij}$  and  $c_{ij}$ . If that is the case however, we will need to investigate any implication this point has on the transmission powers. Recall that the transmission powers are set such that signals overcome each other's interference. Returning to our simple example in Fig. 3-3, if Node 2 is not transmitting to Node 3, then Node 1 can set its transmission power to

$$P_{13} = \frac{\gamma c_{13} N_0}{\alpha_{13}}$$

instead of

$$P_{13} = \frac{c_{13} \gamma N_0 W (W + c_{23} \gamma)}{\alpha_{13} (W^2 - \gamma^2 c_{13} c_{23})} > \frac{\gamma c_{13} N_0}{\alpha_{13}} \quad (3.2)$$

and still meet the required  $E_b/I_0$  threshold. The only problem is determining whether Node 2 is transmitting or not. One possibility is to use carrier sensing. However, apart from the problems with carrier sensing mentioned in Appendix B, by our assumption of no simultaneous transmitting and receiving, Node 1 cannot listen for Node 2's signal while Node 1 is transmitting. Even if we somehow overcome that obstacle, Node 2's signal is by design spread using a spreading code that is unknown to Node 1. Node 1 will not be able to distinguish Node 2's signal from thermal noise. Even if Node 1 knows Node 2's spreading code, it will have to de-spread Node 2's signal to detect its presence. If Node 1 is to do the same for other potential transmitters, there will be a large cost to be paid in system complexity. These reasons imply that Node 1 needs to transmit at the conservative and non-optimal power level given in Eq. 3.2 regardless of whether Node 2 is transmitting or not, in order to *guarantee* meeting the required  $E_b/I_0$  threshold. Therefore, we adopt the model that when a node is transmitting, it assumes the worst-case interference, which occurs when all the potential transmitters transmit.

The final component in setting up the convex cost function is the realization that what we really want is to minimize the energy and not the power. Minimizing the energy is equivalent to minimizing the average power. *While* Node 1 is transmitting to Node 3, the transmission power is  $P_{13}$ , as given above. However, Node 1 transmits

to Node 3 in only a fraction  $\phi_{13}$  of the time. Therefore, the average transmission power (over time) that Node 1 radiates, assuming fixed  $c_{13}$ , is

$$\begin{aligned}\tilde{P}_{13} &= \phi_{13}P_{13} \\ &= \frac{f_{13}}{c_{13}}P_{13} \\ &= \frac{f_{13}}{c_{13}} \frac{c_{13}\gamma N_0 W(W + c_{23}\gamma)}{\alpha_{13}(W^2 - \gamma^2 c_{13}c_{23})} \\ &= f_{13} \frac{\gamma N_0 W(W + c_{23}\gamma)}{\alpha_{13}(W^2 - \gamma^2 c_{13}c_{23})}\end{aligned}$$

Similarly,

$$\tilde{P}_{23} = f_{23} \frac{\gamma N_0 W(W + c_{13}\gamma)}{\alpha_{23}(W^2 - \gamma^2 c_{13}c_{23})}$$

To make the distinction between  $P_{ij}$  and  $\tilde{P}_{ij}$  clear, we refer to the transmission power for Link  $(i, j)$  while Node  $i$  is transmitting (i.e.,  $P_{ij}$ ) as the *nominal transmission power*.  $\tilde{P}_{ij}$  is the *average transmission power*.

Returning to our example, the aggregate average transmission power is

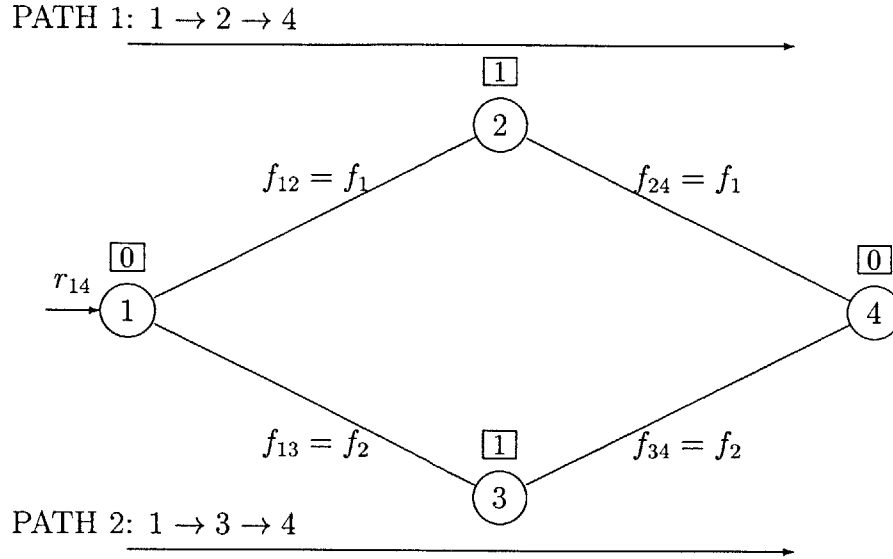
$$\tilde{P}_T = f_{13} \frac{\gamma N_0 W(W + c_{23}\gamma)}{\alpha_{13}(W^2 - \gamma^2 c_{13}c_{23})} + f_{23} \frac{\gamma N_0 W(W + c_{13}\gamma)}{\alpha_{23}(W^2 - \gamma^2 c_{13}c_{23})} \quad (3.3)$$

We note that  $\tilde{P}_T$  is a linear, and thus convex, function of  $f_{13}$  and  $f_{23}$  as desired. The linearity generalizes to an arbitrary number of receivers and transmitters. We simply solve for the different  $P_{ij}$ 's, assuming that all the transmitters are indeed transmitting, and the resulting aggregate average transmission power is

$$\tilde{P}_T = \sum_{(i,j)} f_{ij} \frac{P_{ij}}{c_{ij}} \quad (3.4)$$

which is still linear as a function of the  $f_{ij}$ 's.

In Subsection 3.1.1 we mention that our interference model does not account for the fact that the  $E_b/I_0$  threshold is a function of the nominal data rate, and that signals from the same transmitter can create some interference for each other. Since the  $P_{ij}$ 's (the nominal transmission powers) are now independent of the flow rates,



**Figure 3-4:** A four-node simple network, with the two paths for the end-to-end session from Node 1 to Node 4 shown.

we can be more precise when solving for their values. If we do incorporate the above details into our model, then the result will be that the nominal transmission powers will converge to different (and more accurate) levels for the same nominal data rates. However, those different power levels simply change the constants multiplying the  $f_{ij}[k]$ 's in the cost function, which is not likely to alter any of the insight we develop.

### 3.2.1 Adapting The Cost Function For Scheduling

We now adapt the cost function to the case where scheduling is needed. Consider the simple four-node network in Fig. 3-4. Associated with each node is a single digit label. The frame consists of two slots. In Slot 1, the nodes with the label 0 (i.e., 1 and 4) are allowed to transmit to those with the label 1. In Slot 2 the direction of transmission is reversed. Suppose there is only one end-to-end session in the network from Node 1 to Node 4, with a required end-to-end bit rate of  $r_{14}$ . As is apparent from Fig. 3-4, there are two paths for the data. The first path is  $1 \rightarrow 2 \rightarrow 4$  and the second is  $1 \rightarrow 3 \rightarrow 4$ . In order to satisfy the required  $E_b/I_0$ , the following equations

need to be satisfied

$$\begin{aligned}
1 \rightarrow 2 & : \frac{\alpha_{12}P_{12}/c_{12}}{N_0} = \gamma \\
1 \rightarrow 3 & : \frac{\alpha_{13}P_{13}/c_{13}}{N_0} = \gamma \\
2 \rightarrow 4 & : \frac{\alpha_{24}P_{24}/c_{24}}{N_0 + \alpha_{34}P_{34}/W} = \gamma \\
3 \rightarrow 4 & : \frac{\alpha_{34}P_{34}/c_{34}}{N_0 + \alpha_{24}P_{24}/W} = \gamma
\end{aligned}$$

yielding

$$P_{12} = \frac{\gamma c_{12} N_0}{\alpha_{12}} \quad (3.5)$$

$$P_{13} = \frac{\gamma c_{13} N_0}{\alpha_{13}} \quad (3.6)$$

$$P_{24} = \frac{c_{24} \gamma N_0 W (W + c_{34} \gamma)}{\alpha_{24} (W^2 - \gamma^2 c_{24} c_{34})} \quad (3.7)$$

$$P_{34} = \frac{c_{34} \gamma N_0 W (W + c_{24} \gamma)}{\alpha_{34} (W^2 - \gamma^2 c_{24} c_{34})} \quad (3.8)$$

From conservation of *flow*,

$$r_{14} = f_{12} + f_{13}, \quad f_{12} = f_{24}, \quad f_{13} = f_{34}.$$

Recall that  $f_{ij}$  is the flow rate, in bits per second, on  $(i, j)$ . We observe that  $f_{12}$  must be less than  $c_{12}/2$ , because  $(1, 2)$  is enabled in only one of the two slots in the frame. Recall that when  $(i, j)$  is enabled, Node  $i$  may transmit to Node  $j$ , although it does not necessarily have to.

The aggregate average transmission power is the average power due to signals transmitted in Slot 1 plus that due to signals transmitted in Slot 2. Let  $\tilde{P}_T[1]$  and  $\tilde{P}_T[2]$  be the average transmission powers due to signals transmitted in Slots 1 and 2 respectively. Calculating those average transmission powers we get,

$$\tilde{P}_T[1] = f_{12} \frac{P_{12}}{c_{12}} + f_{13} \frac{P_{13}}{c_{13}} \quad (3.9)$$

and

$$\tilde{P}_T[2] = f_{24} \frac{P_{24}}{c_{24}} + f_{34} \frac{P_{34}}{c_{34}} \quad (3.10)$$

In a more complex network which uses neighbor-based scheduling, there may be multiple slots in which  $(i, j)$  is enabled. Suppose there are two slots, Slot  $k$  and Slot  $k'$ , in which  $(i, j)$  is enabled. There is no reason for the nominal data rate,  $c_{ij}$ , to be the same in both slots. To make the distinction, we use the notation  $c_{ij}[k]$  and  $c_{ij}[k']$  to denote the nominal data rates on Link  $(i, j)$  in Slot  $k$  and Slot  $k'$  respectively. If a link is not enabled in a slot, then its nominal data rate in that slot will be zero.

Continuing with the supposition that  $(i, j)$  is enabled in two slots, the nominal transmission power,  $P_{ij}$ , is also generally different in each of the two slots.  $P_{ij}$  in a slot is a function of the nominal data rates of other links that are enabled in that slot. Therefore, for the nominal transmission powers we also have  $P_{ij}[k]$  and  $P_{ij}[k']$ .

If the flow rate on the link is  $f_{ij}$ , then part of that flow will occur in Slot  $k$  and the other part in  $k'$ . Accordingly,  $f_{ij} = f_{ij}[k] + f_{ij}[k']$ , where  $f_{ij}[k]$  is the flow rate, in bits per second, of the traffic on  $(i, j)$  in Slot  $k$ .

Therefore, Eqs. 3.9 and 3.10 should be rewritten as

$$\tilde{P}_T[1] = f_{12}[1] \frac{P_{12}(c_{12}[1], c_{13}[1], c_{24}[1], c_{34}[1])[1]}{c_{12}[1]} + f_{13}[1] \frac{P_{13}(c_{12}[1], c_{13}[1], c_{24}[1], c_{34}[1])[1]}{c_{13}[1]}$$

and

$$\tilde{P}_T[2] = f_{24}[2] \frac{P_{24}(c_{12}[2], c_{13}[2], c_{24}[2], c_{34}[2])[2]}{c_{24}[2]} + f_{34}[2] \frac{P_{34}(c_{12}[2], c_{13}[2], c_{24}[2], c_{34}[2])[2]}{c_{34}[2]}$$

The overall aggregate average transmission power is then

$$\tilde{P}_T = \tilde{P}_T[1] + \tilde{P}_T[2]$$

We can generalize the above example to any arbitrary PRNET. Let

- $K$  be the total number of slots in the frame.

- $\mathbf{F}[k] = [f_{ij}[k]]$ , where  $f_{ij}[k]$  is the flow rate on  $(i, j)$  that occurs in Slot  $k$  (i.e.,  $f_{ij} = \sum_k f_{ij}[k]$ ).
- $\mathbf{C}[k] = [c_{ij}[k]]$ , where  $c_{ij}[k]$  is the nominal data rate on  $(i, j)$  in Slot  $k$ .
- $\mathbf{P}(\mathbf{C}[k])[k] = [P_{ij}(\mathbf{C}[k])[k]]$ , where  $P_{ij}[k]$  is the nominal transmission power radiated by Node  $i$  due to the data transmitted on  $(i, j)$  (at rate  $c_{ij}[k]$ ) in Slot  $k$ .

The convex cost function we want to minimize is

$$\tilde{P}_T = \sum_{k=1}^K \sum_{(i,j)} f_{ij}[k] \frac{P_{ij}(\mathbf{C}[k])[k]}{c_{ij}[k]} \quad (3.11)$$

which, we again emphasize, is linear and therefore convex as a function of the link flow rates.

### 3.2.2 Comparable Levels of Energy Use

In Chapter 1 we argue that by minimizing the aggregate transmission energy, we do not necessarily ensure that the different nodes use comparable levels of energy. One way to help achieve comparable levels of power use is to minimize the sum of the squares of the average transmission powers instead of the linear sum. In other words we can use the following cost function

$$\sum_{k=1}^K \sum_{(i,j)} \left( f_{ij}[k] \frac{P_{ij}(\mathbf{C}[k])[k]}{c_{ij}[k]} \right)^2$$

Minimizing the sum of the squares penalizes higher average transmission powers more than in the case of minimizing the linear sum. Accordingly, the solution is discouraged from having certain average transmission powers be significantly higher than the others. If we use higher powers (than 2), then we will tend even more towards a solution with comparable levels of average transmission powers. Finally we note that the above cost function is still convex as a function of the flow rates (even if we use higher powers).

### 3.2.3 Incorporating Delay

One of the useful properties of convex functions is that the sum of convex functions is also convex. This property allows us to incorporate delay into our objective function. There is a very strong reason for wanting to include delay. The maximum flow rate that can be achieved on  $(i, j)$  in Slot  $k$  is  $\frac{c_{ij}[k]}{K}$ . The transmission on  $(i, j)$  occurs at most a fraction  $1/K$  of the time, and when it occurs the nominal data rate is  $c_{ij}[k]$ . Therefore, in the mathematical formulation we need to have the constraint

$$f_{ij}[k] \leq \frac{c_{ij}[k]}{K}$$

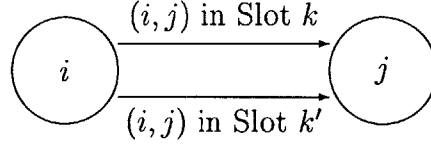
However, with the above constraint alone, there is nothing to discourage the routing algorithm with the objective function given in Eq. 3.11 to set some  $f_{ij}[k]$  equal to or arbitrarily close to  $\frac{c_{ij}[k]}{K}$ . But if that happens, then the average number of packets waiting to be transmitted on  $(i, j)$  in Slot  $k$  will become arbitrarily large, resulting in a buffer overflow. In other words, pure minimum-energy routing can congest certain links, which presumably is an undesirable effect. Therefore, it is necessary to find a way to incorporate delay into minimum-energy routing.

To start, we need to develop a cost function associated with each link. As in optimal-delay routing for point-to-point networks, we use an M/M/1 approximation for each link. The actual delays experienced by packets are due to queuing delays, as well as frame delays (i.e., time until next transmission slot). Exact analysis of the delay experienced by packets therefore is likely to be hard.

Having made the distinction between nominal data rates and flow rates, it ought to be straightforward to simply incorporate delay into the cost function. The only catch is that we need to model each link on a slot by slot basis. Let  $D_{ij}[k]$  be the average number of packets waiting to be transmitted on  $(i, j)$  in Slot  $k$ .

$$D_{ij}[k] = \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]}$$

We emphasize the division of  $c_{ij}[k]$  by  $K$ , to account for the fact that the transmission



**Figure 3-5:** A link that is enabled in two different slots is modeled as two separate links.

in that slot occurs once every  $K$  slots.

$$Z = w_1 \sum_{k=1}^K \sum_{(i,j)} f_{ij}[k] \frac{P_{ij}(\mathbf{C}[k])[k]}{c_{ij}[k]} + w_2 \sum_{k=1}^K \sum_{(i,j)} \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]} \quad (3.12)$$

where  $w_1$  and  $w_2$  are weights that reflect the relative importance of delay and power.

We have implicitly assumed multiple links between nodes, with each of the multiple links corresponding to a different slot in which a node transmits to its neighbor, as we show in Fig. 3-5. Each link among the multiple links is modeled to have a separate queue. This may or may not be the case in practice, as a node may place all the packets going to the same neighbor in one queue. However, since we are trying to approximate average number of packets in the network, this model is likely to be adequate, much like the M/M/1 approximation is considered adequate.

The cost function in Eq. 3.12 is a combination of the aggregate average transmission power and the average number of packets in the networks. If we divide it by the total traffic rate into the PRNET, we will get a combination of the average energy per packet and the average delay experienced by a packet.



### 3.3 Minimum-Combined-Energy-and-Delay Routing Formulation

In this section we summarize and formalize minimum-combined-energy-and-delay routing (MCED routing), along the same lines of the optimal-delay routing formulation in [BeG92]. Let

- $\mathbf{R} = [r_{ij}]$ , where  $r_{ij}$  is required end-to-end rate (from Node  $i$  to Node  $j$ .)
- $K =$  total number of slots in the frame.
- $\mathbf{F}[k] = [f_{ij}[k]]$ , where  $f_{ij}[k]$  is the flow rate on  $(i, j)$  in Slot  $k$ .
- $\mathbf{C}[k] = [c_{ij}[k]]$ , where  $c_{ij}[k]$  is the nominal data rate on  $(i, j)$  in Slot  $k$ .
- $\mathbf{P}(\mathbf{C}[k])[k] = [P_{ij}(\mathbf{C}[k])[k]]$ , where  $P_{ij}[k]$  is the nominal transmission power used by Node  $i$  when transmitting on  $(i, j)$  in Slot  $k$ .
- $\Pi_{ij}$  is the set of paths connecting an origin node, Node  $i$ , to a destination node, Node  $j$ . We note that if a path uses two links between the same pair of nodes (that occur in two different slots,) then that path will be considered to be two distinct paths.
- $x_\pi$  is the flow rate over path  $\pi$ .
- $w_1$  and  $w_2$  are the weights that reflect the relative importance of delay and power.

The objective is then to minimize  $Z$  where

$$Z = w_1 \sum_{k=1}^K \sum_{(i,j)} f_{ij}[k] \frac{P_{ij}(\mathbf{C}[k])[k]}{c_{ij}[k]} + w_2 \sum_{k=1}^K \sum_{(i,j)} \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]}$$

subject to the constraints

$$\sum_{\pi \in \Pi_{ij}} x_\pi = r_{ij}, \quad \forall i \text{ and } j$$

$$\begin{aligned}
f_{ij}[k] &= \sum_{\substack{\{\pi : (i,j) \text{ enabled} \\ \text{in Slot } k \in \pi\}}} x_{\pi}, & \forall i, j, \text{ and } k \\
f_{ij}[k] &\leq \frac{c_{ij}[k]}{K}, & \forall i, j, \text{ and } k \\
x_{\pi} &\geq 0, & \forall \pi
\end{aligned}$$

### 3.4 Decomposition of Routing and Power Control

In this section, we discuss how the distinction between link flow rates and nominal data rates amounts to an additional decomposition in the problem of minimum-energy communication over a PRNET. Originally the problem of how to set the nominal transmission powers for signals was viewed as an inherent part of the minimum-energy routing problem. This is because the nominal transmission powers are determined directly by the nominal data rates, the  $c_{ij}$ 's for the different signals; and before making the distinction between link flow rates, the  $f_{ij}$ 's, and nominal data rates, we assumed that the nominal data rates were the variables to be adjusted in order to meet the end-to-end rate requirements and the flow conservation constraints.

However, making a distinction between link flow rates and nominal data rates is necessary. With packets arriving into the network according to some random process, a link flow rate needs to be less than the nominal data rate for that link, or else that link can become congested.

As a result of separating link flow rates from nominal data rates, we are able to model the nominal data rates for links, and hence their respective nominal transmission powers, as being constant. In addition we are able to change the minimum-energy routing problem to that of determining the flow rates on different links, making the minimum-energy routing problem independent of the power control problem. The power control problem in turn becomes part of a new problem, namely the problem of determining the initial values for the nominal data rates, and how to make subsequent adjustments. This is an interesting new problem that we address in Section 3.6, although further research is needed.

### 3.5 Distributed Implementation of MCED Routing

Let

$$Z_{ij}[k] = w_1 f_{ij} \frac{P_{ij}(\mathbf{C}[k])[k]}{c_{ij}[k]} + w_2 \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]}$$

In a distributed implementation of the optimal routing algorithm, the routing information that nodes need to exchange with each other is the set of first derivatives

$$\frac{\partial Z_{ij}[k]}{\partial f_{ij}[k]}$$

for all  $(i, j)$  and  $k$  [Gal77], [BeG92]. We assume that every node is continually optimizing the flow rates on its outgoing links. Suppose some link  $(i, j)$  is enabled in two slots,  $k$  and  $k'$ , and the flow rate in Slot  $k$  is  $f_k$  and that in Slot  $k'$  is  $f_{k'}$ . If  $f_k$  and  $f_{k'}$  are both positive then we must have

$$\left. \frac{\partial Z_{ij}[k]}{\partial f_{ij}[k]} \right|_{f_{ij}[k]=f_k} = \left. \frac{\partial Z_{ij}[k']}{\partial f_{ij}[k']} \right|_{f_{ij}[k']=f_{k'}}$$

Otherwise Node  $i$  will be able to reduce the total network cost,  $Z$ , by shifting a small amount of flow from the slot with the higher first derivative to the slot with the smaller first derivative. If  $f_k$  is positive and  $f_{k'}$  is 0, then we must have

$$\left. \frac{\partial Z_{ij}[k]}{\partial f_{ij}[k]} \right|_{f_{ij}[k]=f_k} < \left. \frac{\partial Z_{ij}[k']}{\partial f_{ij}[k']} \right|_{f_{ij}[k']=0}$$

The point of the above argument is that any given link  $(i, j)$  can be thought of as a set of sub-links, with each sub-link corresponding to a different slot in which  $(i, j)$  is enabled within the frame. For the purpose of minimizing the total network cost, Node  $i$  locally allocates the flows on the different sub-links such that the subset of sub-links with positive flows on them share the same value for the first derivatives with respect to the flow rates on them. Therefore, Node  $i$  need only pass that single value to its neighboring nodes. This implies that in the distributed implementation of optimal

routing in PRNETs, the issue of slots becomes moot, and the routing information that a node passes is a single first derivative measure.

### 3.6 Adjusting the Nominal Data Rates

So far in this chapter we have implicitly assumed that the nominal data rates, the  $c_{ij}[k]$ 's, are constant. In PRNETs we can adjust the  $c_{ij}[k]$ 's, as is done in the DARPA PRNET [Ka+78]. This can be achieved by adjusting the nominal transmission power to maintain the required  $E_b/I_0$  as the rate increases or decreases. In this section we relax the assumption of fixed  $c_{ij}[k]$ 's, and allow the nodes to alter the nominal data rates on their outgoing links. This is not entirely consistent with our static model for the PRNET in this chapter, but we temporarily relax that assumption in order to investigate the problem of adjusting the  $c_{ij}[k]$ 's.

There are several reasons for wanting to adjust the nominal data rates. First, the nominal transmission powers are set at conservative levels responsive to the nominal data rates of potential interferers. If a certain link becomes underutilized, lowering its nominal data rate will lower the nominal transmission powers of other users, and consequently lower the interference in the network as a whole. On the other hand, if a link becomes congested with flow, then increasing the nominal data rate will lower the congestion on the link.

We suggest that nominal data rates be adjusted at a slow enough frequency to allow the link flow rates (i.e., the  $f_{ij}[k]$ 's) to stabilize. We address the issue of how frequently to adjust the  $c_{ij}[k]$ 's more carefully in Chapter 4. Recall that the system cost function, as a function of both the flow rates and the nominal data rates, is

$$Z = w_1 \sum_{k=1}^K \sum_{(i,j)} f_{ij} \frac{P_{ij}(\mathbf{C}[k])[k]}{c_{ij}[k]} + w_2 \sum_{k=1}^K \sum_{(i,j)} \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]}$$

Despite difficulties in practice, we assume that the  $c_{ij}[k]$ 's can be adjusted incrementally (i.e., the  $c_{ij}[k]$ 's are continuous). We assume that at every adjustment of the nominal data rates, the flow rates have stabilized, and therefore the  $f_{ij}[k]$ 's can be

treated as constants. At every adjustment of the nominal data rates, the  $c_{ij}[k]$ 's are altered such that the total cost,  $Z$ , is reduced (assuming the flow rates are constant). As we show in Appendix C, the first term in the cost function is not generally a convex function of the nominal data rates. Therefore, the cost function as a whole is not convex, making it hard to solve for a global minimum. However the cost function is differentiable, and therefore gradient methods can be applied to find cost-improving directions.

$P_{ij}(\mathbf{C}[k])[k]$  is an increasing function of the  $c_{ij}[k]$ 's. If  $w_2 = 0$  (i.e., if delay is omitted from the cost function) then  $Z$  will be an increasing function of the  $c_{ij}[k]$ 's. Therefore, for  $w_2 = 0$ , minimizing  $Z$  with respect to the  $c_{ij}[k]$ 's drives each  $c_{ij}[k]$  towards  $f_{ij}[k]$ . This drives delay to infinity. Therefore, incorporating delay into the cost function will be a necessity if we want to adjust the nominal data rates in a sensible and useful manner.

One concern about the above method of adjusting the nominal data rates is potential oscillations. After adjusting the  $c_{ij}[k]$ 's, new values for the  $f_{ij}[k]$ 's are computed. The  $c_{ij}[k]$ 's are then adjusted with respect to the new  $f_{ij}[k]$ 's. The concern is that we may return to the old values of  $c_{ij}[k]$ 's, which would lead to the return to the old values of  $f_{ij}[k]$ 's, and so on. However since with every adjustment we are making a descent (i.e., decreasing  $Z(\mathbf{C}[k], \mathbf{F}[k])$ ), a point in the space of  $\mathbf{C}[k]$  and  $\mathbf{F}[k]$  is not revisited unless it is a local minimum. Therefore, once the flow rates are optimized for the new set of nominal data rates, it is not cost-improving to return to the old set of nominal data rates, and hence oscillations do not occur.

Another insight developed from the above analysis is that if  $c_{ij}[k]$  is 0 for a link, then it will never be cost-improving to marginally increase it to a positive value even though further increases can eventually lead to a lower network cost. This is because increasing  $c_{ij}[k]$  from 0 to any positive value causes the aggregate average transmission power to increase, regardless of how small that positive value is. The increase includes at least the increase in transmission energy over  $(i, j)$  in Slot  $k$  from 0. At the same time, since there is no previous flow on  $(i, j)$  in Slot  $k$  there is no direct benefit in terms of reducing aggregate delay. This creates an undesirable effect, as links with nominal

data rates of zero are never allowed to carry data. This undesirable effect is overcome by setting the nominal data rate between any two neighbors to be at least some minimum nominal data rate  $c_{\min}$ , which allows for the introduction of flow on that link, and hence for the possibility of increasing the nominal data rate. Having some links initially adopt  $c_{\min}$ , just to allow for the possibility of increasing their nominal data rate, could be viewed as creating unnecessary potential interference. However, the value of  $c_{\min}$  can be arbitrarily small, so that the additional cost incurred by the network for having such links is also small.

The above analysis indicates that even under the simplifying assumptions of a quasi-static PRNET model and the ability to adjust the nominal data rates incrementally (i.e., that the  $c_{ij}[k]$ 's are continuous), the  $c_{ij}[k]$ 's can only be adjusted such that  $Z$ , the overall cost, can be improved, as opposed to be globally optimized. We also consider that the link flow rates are adjusted between adjustments of the  $c_{ij}[k]$ 's. This implies that the global minimum (with respect to the  $c_{ij}[k]$ 's) at one adjustment is different from that at the next adjustment, because the values of the  $f_{ij}[k]$ 's are different. On the other hand,  $Z$  is a global cost function, and each adjustment of the  $c_{ij}[k]$ 's, using gradient methods, requires sharing information among all the nodes in the network. Since global optimality with respect to the nominal data rates is not guaranteed to be achieved, it might not be worthwhile running such a complex distributed algorithm. In practical PRNETs it may be adequate for each node to adjust its nominal data rates independently of other nodes, and seek to improve the overall cost using simple rules of thumb, that likely improve the overall cost. One possible rule of thumb, that requires further study, is the following: if  $f_{ij}[k]$  is greater than a certain fraction of  $c_{ij}[k]$ , then increase  $c_{ij}[k]$  to the next level, and if it is lower than another fraction then decrease  $c_{ij}[k]$ .

Another somewhat complicated practical problem is how to set the initial nominal data rate for  $(i, j)$  in Slot  $k$ . In practice, there is likely to be a limit on how small  $c_{\min}$  can be made, implying that if a new link is severely underutilized it will create unnecessarily high potential interference. However, presumably the topological organization algorithm establishes new links with a preconception that a new link is to be

sufficiently utilized (e.g., if that link provides a short-cut for some path, and costs less in terms of power and delay). In that case, the topological organization algorithm may recommend a suitable  $c_{ij}$  for the new link, based on how much traffic it foresees going over it. It also seems appropriate to monitor the utilization of new links. If, for some reason, the routing algorithm does not introduce the anticipated flow rate on the new link, then one option to consider will be to reduce the nominal data rate on that link, or remove it altogether. If the method for choosing initial values for the nominal data rates is along the lines of the above description, then the topological algorithms will be significantly involved in setting the  $c_{ij}[k]$ 's, having to handle some very interesting problems, such as estimating the amount of flow on a new link.

### 3.7 Summary

Using the static PRNET model, we were able to develop several results and insights for the minimum-energy routing problem. First, we were able to construct a meaningful convex cost function for the minimum-energy routing problem. The construction hinged on making the distinction between the flow rate on a link and the nominal data rate. It was also based on the assumption that when nodes transmit signals, they need to use conservative and hence non-optimal levels of nominal transmission powers in order to overcome potential interference from other transmitters. We also discovered that minimum-energy routing can lead to congesting certain links in the network. This implied that delay needed to be incorporated into minimum-energy routing for the latter to be useful. Incorporating delay was greatly simplified by the convex cost function developed for minimum-energy routing. We referred to the resulting routing algorithm as the minimum-combined-energy-and-delay routing (MCED routing) algorithm.

We also looked into the potential benefits of adjusting the nominal data rates, and found that there are some. Unfortunately, it is difficult to globally optimize the MCED routing cost function with respect to the nominal data rates. The problem of adjusting nominal data rates in wireline networks also yields non-convex cost functions

[BeG92] (pp. 439-445). Based on our analysis, we proposed practical methods for adjusting the nominal data rates, although further research is needed.



## Chapter 4

# Minimum-Combined-Energy-and-Delay Routing in Dynamically Varying PRNETs

In this chapter our goal is to use the insight developed for minimum-combined-energy-and-delay (MCED) routing for static PRNETs to establish a framework for MCED routing in dynamically varying PRNETs. The first step in the process is to analyze the dynamic nature of the PRNET and how it affects routing.

### 4.1 The Effect of PRNET Dynamics on Routing

We have mentioned that the dynamic nature of the PRNET is primarily due to two physical factors, namely the mobility of the nodes and the changing characteristics of the radio channels. The PRNET also needs to dynamically accommodate the changing nature of traffic demand.

The mobility of nodes affects routing in two main ways. First, as the nodes move around, each node modifies its set of neighbors, implying that the topology changes. Second, the change in the distance between two nodes, Node  $i$  and Node  $j$ , results in a change in  $\alpha_{ij}$ , the path propagation loss between the two nodes. Furthermore, if Node  $j$  suddenly moves behind an obstructing object, then  $\alpha_{ij}$  can suddenly change

drastically.

The changing characteristics of a radio channel are caused by multipath interference which can affect the channel upon the slightest movement of a node or change in the surrounding environment. It too has the effect that the  $\alpha_{ij}$ 's change.

The significance of changing  $\alpha_{ij}$ 's is that, through the power control algorithm, nominal transmission powers also change. This in turn implies that the costs (in terms of energy) of transmitting bits on the links change, something that a MCED routing algorithm needs to react to.

The routing algorithm in a PRNET needs to adapt to changes other than the ones caused by the dynamic nature of the PRNET. It needs to adapt to changes in end-to-end rate requirements and changes in nominal data rates (i.e., the  $c_{ij}$ 's). Both classes of change generally require adjusting the flow rates on some paths. In addition, changes in end-to-end rate requirements may result in nodes adding or dropping neighbors to improve the PRNET's ability to satisfy these requirements.

Our approach to managing the different changes listed above is to classify them according to the frequencies with which they occur. Below we list the changes that the MCED routing algorithm needs to cope with, discussing the frequency at which each type of change occurs. The list below is in order of non-increasing frequency of occurrence:

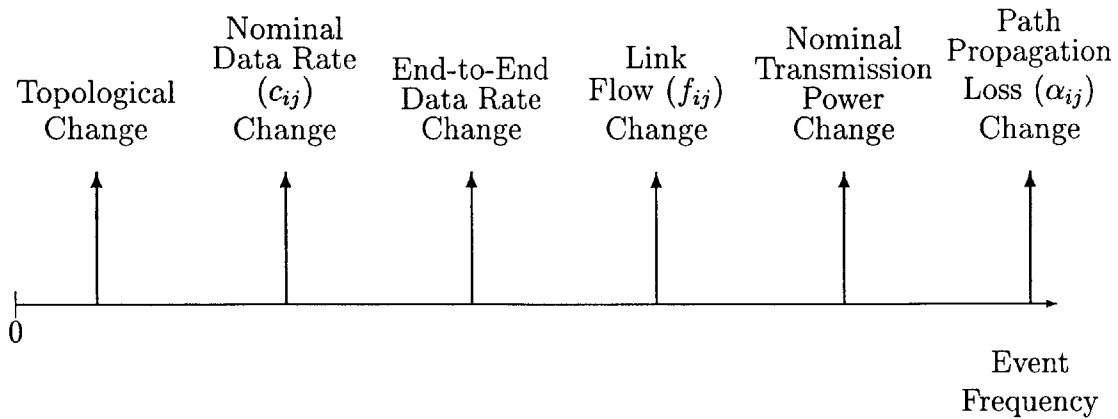
1. Changes in Path Propagation Losses ( $\alpha_{ij}$ 's): We identified three types of changes that occur to an  $\alpha_{ij}$ :
  - Type 1: Changes due to the changing characteristics of the radio channel.
  - Type 2: Changes due to a significant change in distances between nodes.
  - Type 3: Changes due sudden introduction of obstructing objects.

The highest frequency at which an  $\alpha_{ij}$  changes, is the frequency at which Type 1 changes occur, because they can be caused by the slightest movement of the node or the slightest change in the environment. The time between measurable changes in an  $\alpha_{ij}$  can be on the order of milliseconds.

2. Changes in Nominal Transmission Powers ( $P_{ij}[k]$ 's): The nominal transmission powers are altered by the power control algorithm in response to changes in the  $\alpha_{ij}$ 's. Ideally, the power control algorithm can adjust the nominal transmission powers at the same rate as changes in the  $\alpha_{ij}$ 's. However, the nominal transmission power for a link in a certain slot can only be changed once every frame length, which could be on the order of fractions of a second.
3. Changes in Link Flow Rates ( $f_{ij}[k]$ 's): The MCED routing algorithm determines the amounts of flow on different paths and links in the PRNET. The MCED routing algorithm adjusts those flow rates, as nominal transmission power levels change. However, given that the routing algorithm needs to determine flow rates along entire paths, the adjustment of link flow rates (i.e.  $f_{ij}[k]$ 's) requires network-wide coordination. Hence it is not likely that the routing algorithm can adjust the  $f_{ij}[k]$ 's at the same rate as changes in the  $P_{ij}[k]$ 's, and remain stable. Therefore we conclude that  $f_{ij}[k]$  changes likely occur at a lower frequency than  $P_{ij}[k]$  changes.
4. Changes in End-to-End Rate Requirements: The frequency at which end-to-end rate requirements change can be anything, as it depends on the behavior of the end users, the applications they are running, and how the flow control algorithm regulates the traffic into the PRNET.<sup>1</sup> Assuming that the flow control algorithm prevents rapid increases in the actual end-to-end data rates, we assume that end-to-end rate changes occur less frequently than changes to flow rates.
5. Changes in Nominal Data Rates ( $c_{ij}[k]$ 's): As discussed in the last chapter, it is beneficial to adjust the nominal data rates occasionally. There are at least two reasons for wanting the  $c_{ij}[k]$ 's to change less frequently than the  $f_{ij}[k]$ 's. The first is delay. If the  $c_{ij}[k]$ 's change at the same or higher frequency than the  $f_{ij}[k]$ 's, then it will not be likely that flows in the network can stabilize into paths with good delay performance. Second, if the  $c_{ij}[k]$ 's change rapidly

---

<sup>1</sup>See [BeG92] for more information about flow control.



**Figure 4-1:** Relative frequencies of events related to PRNET routing.

relative to the  $f_{ij}[k]$ 's, the entire structure of the routing algorithm will have to be changed. This creates an entirely new routing problem which we have not addressed.

6. Topological Changes: Topological changes are brought about by the mobility of nodes, and also decisions by nodes to seek out more appealing neighbors. The “rate” of mobility depends on the type of the PRNET, but we believe it is reasonable to assume that changes due to mobility of nodes are infrequent compared to the other events. As for seeking out more appealing neighbors, at least part of the decision making process for a node depends on observed patterns in routing and adjustments of nominal data rates. Such patterns have to be monitored for periods of time that allow multiple changes in end-to-end rate requirements,  $c_{ij}[k]$ 's, and  $f_{ij}[k]$ 's.

Fig. 4-1 summarizes how we view the relative frequencies of changes. Separating the events by frequency of occurrence helps us carry over most of the results developed for MCED routing in static PRNETs to dynamic PRNETs.

The MCED routing algorithm is continuously running in a PRNET, adjusting flow rates in response to the different types of changes outlined above. Frequency

separation allows us to look at the network at different time-scales. According to the view illustrated in Fig. 4-1, the MCED routing algorithm can make many adjustments to the link flow rates, the  $f_{ij}[k]$ 's, before a  $c_{ij}[k]$  adjustment occurs.

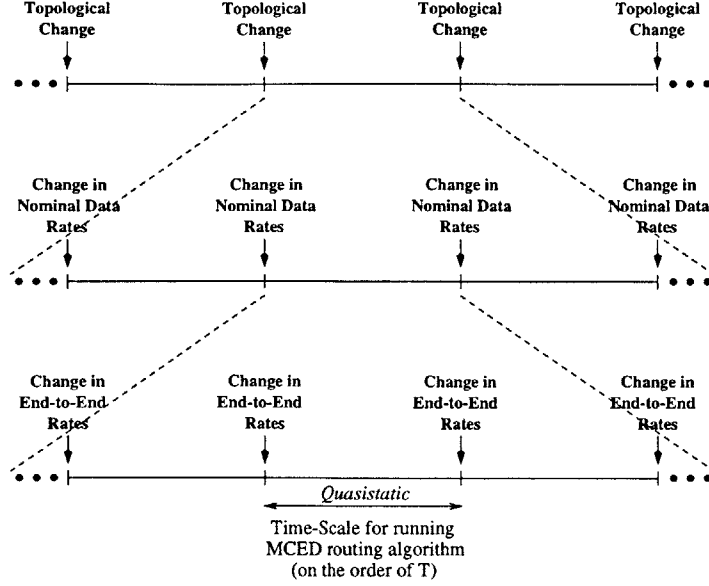
Similar statements apply to topological changes and changes to end-to-end rates. Using the  $c_{ij}[k]$ 's as an example, the MCED routing algorithm may converge or come close to converging to the set of  $f_{ij}[k]$ 's appropriate for the current values of  $c_{ij}[k]$ 's, before an adjustment to the set of  $c_{ij}[k]$ 's occurs, creating a new optimal point for the link flow rates, towards which the MCED routing algorithm subsequently tries to converge.

We can think of the continuous running of the MCED routing algorithm as a series of executions. Over the time period of a single execution, the  $c_{ij}[k]$ 's can be treated as quasi-static. Indeed the topology and end-to-end rate requirements can also be treated as quasi-static, as their frequencies of change are assumed to be distinctly lower than the frequency of adjustments to the  $f_{ij}[k]$ 's. The benefit of frequency separation is that over the time-scale of a single execution of the MCED routing algorithm, infrequent events (relative to the adjustments of link flow rates) can be discounted for the sake of making the routing problem tractable. Fig. 4-2 illustrates the time-scale at which we envision the MCED routing algorithm to run, relative to other time-scales. We note that the use of frequency separation to simplify complex problems is an established technique in at least one other field, namely manufacturing systems [Ger89].

## 4.2 Completing The Model

Quasi-static problems are usually solved using static models. Hence we model as static the dynamic aspects that appear quasi-static to the MCED routing algorithm. Accordingly, as in Chapter 3, we are able to model the topology, nominal data rates, and end-to-end rates as static. However as Fig. 4-1 indicates, the path propagation losses and the nominal transmission powers cannot be assumed to be static.

First we deal with the path propagation losses. We assume that the path prop-



**Figure 4-2:** Time-scale for MCED routing relative to other time-scales.

agation loss for a link  $(i, j)$  is a stationary stochastic process  $\alpha_{ij}(t)$ . This will be a reasonable assumption if the changes in  $\alpha_{ij}$ 's are solely of Type 1. However, if we consider Type 2 and Type 3 changes, then stationarity will no longer apply. For example as the distance between Node  $i$  and Node  $j$  increases, we expect the signal between them to experience more attenuation on average.

To get around the issue of stationarity, we again take advantage of frequency separation, albeit in a slightly different manner. We argue in the previous section that the frequencies at which Type 2 and Type 3 changes occur are much lower than that of Type 1 changes. We assume that those frequencies are low enough over the time-scale of an execution of the MCED routing algorithm that they can be assumed to be quasi-static. So in analyzing the MCED routing algorithm, we model  $\alpha_{ij}(t)$  as a stationary stochastic process.

For the nominal transmission powers, which are our primary concern, consider a particular slot, Slot  $k$ . Let  $\mathcal{L}[k]$  be the set of links enabled in Slot  $k$ .<sup>2</sup> Also let  $\mathcal{T}[k]$  be the set of transmitting nodes corresponding to the links in  $\mathcal{L}[k]$ . We consider a particular link in  $\mathcal{L}[k]$ ,  $(i, j)$ . Recall that the power transmitted by Node  $i$  to

<sup>2</sup>Recall that when Link  $(i, j)$  is enabled, Node  $i$  may transmit to Node  $j$ , although it does not necessarily have to.

send a signal to node  $j$  in Slot  $k$  is set to overcome potential interference from the transmitters in  $\mathcal{T}[k]$ . Therefore,  $P_{ij}[k]$  is a function of

- all  $c_{lm}[k]$  and  $\alpha_{lm}$  such that  $(l, m) \in \mathcal{L}[k]$ , and
- all  $\alpha_{ij}$  such that  $l \in \mathcal{T}[k]$ .

Due to power control,  $P_{ij}[k]$  changes from frame to frame. The changes in  $P_{ij}[k]$  are due solely to the changes in the  $\alpha_{ij}$ 's, because over the time-scale of consideration the  $c_{ij}[k]$ 's are modeled as static. Therefore, we model the changing  $P_{ij}[k]$  as a discrete stochastic process  $\{P_{ij}^n[k]\}$ . Since  $\{P_{ij}^n[k]\}$  is a function of a stationary stochastic process, we model it as a stationary discrete stochastic process. We assume further that  $\{P_{ij}^n[k]\}$  is ergodic.

#### 4.2.1 Power Control in a Dynamic PRNET

Up to this point we have implied that the nominal transmission power levels are the solution to the governing equations for meeting the  $E_b/I_0$  threshold requirements with equality (see Examples 3.1 - 3.3 in Section 3.1). In a dynamic PRNET that would mean that the equations need to be solved repeatedly at each node. Solving the set of equations requires global knowledge of information. More likely than not, the power control algorithm is distributed, so repeatedly solving the equations at each node with global and current information may not be feasible. Indeed it is more likely for the power control algorithm to be implemented on a link-by-link basis. Each transmitter sets its nominal transmission power based on feedback from the receiver. The receiver instructs the transmitter to either increase or decrease its transmission power based, at least in part, on the measured  $E_b/I_0$  ratio for that signal. We observe that the receiver lumps all interference together, and tries to adjust the power such that its required  $E_b/I_0$  threshold, whatever value it is, is met. This observation implies that the details of the changing  $E_b/I_0$  threshold with nominal data rates and the fact that signals from the same transmitter are not completely orthogonal at the receiver are no longer issues in this model.

The receiver also needs to account for the fact that the nominal transmission power needs to be set conservatively to combat interference from potential transmitters. Therefore in a PRNET with changing channel conditions, it is likely that the best one can hope for is the convergence of the power levels to within a neighborhood of the changing target power levels (i.e., the levels dictated by the governing equations). Indeed it is not clear whether the distributed power control algorithm would seek to converge to those target power levels, as practical considerations may result in a different strategy, although the nominal transmission powers would still need to clear the  $E_b/I_0$  threshold. However whatever the strategy of the power control algorithm, we believe it would be responsive to the  $\alpha_{ij}(t)$ 's, which by modeling assumption are stationary stochastic processes. Therefore the  $\{P_{ij}^n[k]\}$ 's can be regarded as stationary discrete stochastic processes, which we further assume to be ergodic.

### 4.3 Development of A Cost Function for MCED Routing

The key to the development of a convex cost function in the MCED routing formulation for static PRNETs is that the formulation made the nominal transmission powers, the  $P_{ij}[k]$ 's, independent of the link flow rates, the  $f_{ij}[k]$ 's. Below we propose a similar strategy for developing a convex cost function for dynamically varying PRNETs.

In Chapter 3 one of the key observations is that we are really interested in minimizing average power. We focus again on that goal. Let  $\tilde{P}_{ij}[k]$  be the time-average power over a time period  $T$ , due to the transmission from Node  $i$  to Node  $j$  in the  $k^{\text{th}}$  slot of the frame. Let the frames, over the period  $T$ , in which a transmission over  $(i, j)$  in Slot  $k$  occurs be numbered  $1, \dots, S$ . Note that there are other frames over the same period in which no transmissions occur over  $(i, j)$  in Slot  $k$ , and the corresponding nominal transmission powers are zero. Let the time duration of each slot be  $\tau$ . We assume that the nominal transmission power over the duration of a slot



is constant. Let  $P_{ij}^s[k]$  be the nominal transmission power in Slot  $k$  in the  $s^{\text{th}}$  frame in which Node  $i$  transmits to Node  $j$ . Then

$$\begin{aligned}\tilde{P}_{ij}[k] &= \frac{1}{T} \sum_{s=1}^S \tau P_{ij}^s[k] \\ &= \frac{S\tau}{T} \frac{\sum_{s=1}^S P_{ij}^s[k]}{S}\end{aligned}$$

The fraction  $\frac{S\tau}{T}$  is the fraction of time that a transmission occurs over Link  $(i, j)$  in Slot  $k$ . By the ergodicity of  $\{P_{ij}^n[k]\}$ , the ensemble average  $\frac{\sum_{s=1}^S P_{ij}^s[k]}{S}$  approximates the expected value of  $P_{ij}[k]$  in any given frame (if a transmission occurs). Therefore,

$$\begin{aligned}\tilde{P}_{ij}[k] &\approx \phi_{ij} E[P_{ij}[k]] \\ &= \frac{f_{ij}[k]}{c_{ij}[k]} E[P_{ij}[k]]\end{aligned}\tag{4.1}$$

The above derivation assumes that  $\{P_{ij}^n[k]\}$  is independent of the  $f_{ij}[k]$ 's. In theory, that can indeed be true. The stochastic process  $\{P_{ij}^n[k]\}$  depends on nominal data rates of the links in the set  $\mathcal{L}[k]$ , and on the path propagation losses in PRNET. The former is assumed to be fixed over the time-scale of consideration, and independent of the  $f_{ij}[k]$ 's.<sup>3</sup> The  $\alpha_{ij}$ 's are physical variables that we do not control anyway.

However, the practical implementation of the power control may cause the nominal transmission powers to depend on the flow rates. The power control algorithm needs to set the power levels conservatively, such that signals can overcome interference from potential transmitters. However, the receiver must sense the interference from other nodes before it can instruct the transmitter to increase its transmission power to overcome the interference. The closer  $f_{ij}[k]$  is to  $c_{ij}[k]$  for some  $(i, j)$  in Slot  $k$ , the more frequently Node  $i$  transmits to Node  $j$  in Slot  $k$ , and the more likely its interference is registered by other receivers, and hence the more likely their respective transmitters adjust their nominal transmission powers. A potential dependency of the nominal transmission powers on the flow rates is an important problem to address. If

---

<sup>3</sup>For larger time-scales, which we are not considering here, the independence of the  $c_{ij}[k]$ 's from the  $f_{ij}[k]$ 's is not valid, as discussed in Section 3.6.

there is a strong dependency, the result in Eq. 4.1 will be invalid. For the remainder of this discussion, however, we assume that the power control algorithm sets the nominal transmission powers conservatively, based solely on the  $c_{ij}[k]$ 's and the  $\alpha_{ij}$ 's.

Let  $\tilde{P}_T$  be the aggregate average transmission power. Then

$$\tilde{P}_T = \sum_{k=1}^K \sum_{(i,j)} f_{ij}[k] \frac{E[P_{ij}[k]]}{c_{ij}[k]} \quad (4.2)$$

The form of Eq. 4.2 is very similar to that of Eq. 3.11, except that  $P_{ij}[k]$  is replaced by  $E[P_{ij}[k]]$ . However,  $\tilde{P}_T$  is still a convex function of the link flow rates, the  $f_{ij}[k]$ 's. Therefore, for MCED routing in dynamic PRNETs we can employ the following convex cost function

$$Z = w_1 \sum_{k=1}^K \sum_{(i,j)} f_{ij}[k] \frac{E[P_{ij}[k]]}{c_{ij}[k]} + w_2 \sum_{k=1}^K \sum_{(i,j)} \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]}$$

Using the expected values of the nominal transmission powers in the cost function allows us to “slow down” the changes in the  $\alpha_{ij}$ 's. In practice Node  $i$  can approximate  $E[P_{ij}[k]]$  by employing a sliding window average of the actual nominal transmission powers.

## 4.4 Summary

In this chapter we outlined how we propose to deal with the dynamic nature of PRNETs in order to be able to formulate the MCED routing problem for dynamically varying PRNETs. We relied on the concept that different types of changes occur at different frequencies. That view helped us construct a meaningful cost function for the MCED routing problem that is convex as a function of the link flow rates.

# Chapter 5

## Analysis of a Neighbor Management Procedure

Since the communication nodes in a PRNET sometimes make large moves in their locations, it is necessary for the nodes to sometimes change their sets of neighbors. Neighbor changes might also be desirable due to changes in channel conditions or changes in end-to-end traffic requirements. It is likely that different topological algorithms need to run in parallel to handle responses to different changes. It is beyond the scope of our work to give a comprehensive survey of topological algorithms. However, we focus on a procedure that we believe will be a building block for several topological algorithms. The procedure is of particular interest to us because it deals with neighbor management, which in turn is of concern to us because neighbors form the basic structure on which scheduling and routing are based.

Below we describe the procedure in detail. First however, we introduce the concept of the control channel.

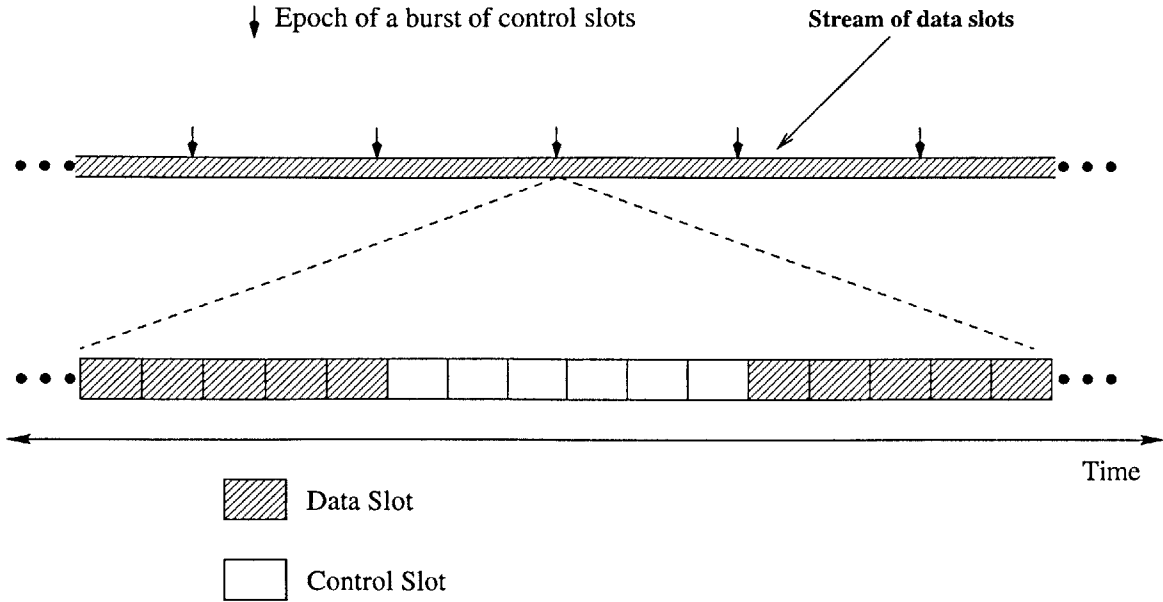
### 5.1 The Data Channel and The Control Channel

As previously described, in our PRNET model time is divided into slots which in turn are grouped into frames. Two neighbors send data packets to each other in certain slots, within a frame, using agreed upon spreading codes. We refer to such

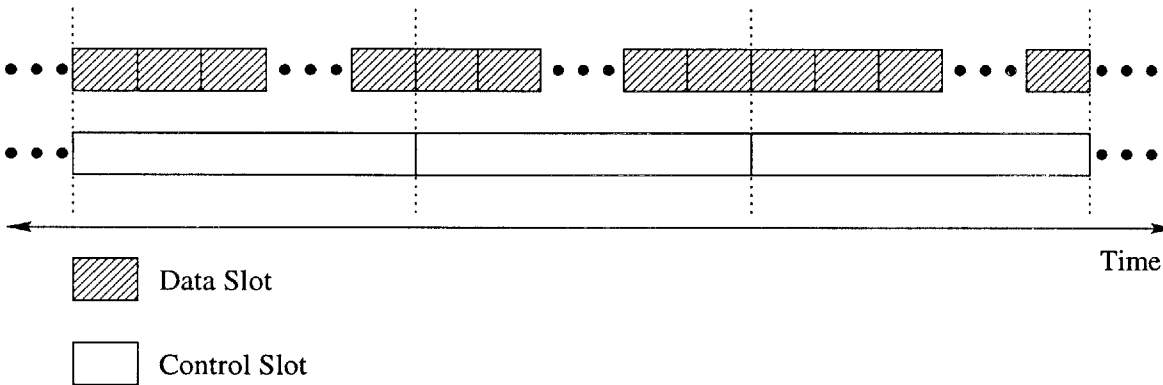
slots as data slots. Data slots form the data channel. We note that data packets may contain information to be used in topological, routing, and other network management algorithms. We refer to such information as administrative information. Data packets containing administrative information can be exchanged among neighbors and among non-neighbors with multi-hop paths established between them. However, if two non-neighbors do not initially know about each other's existence, and hence have no multi-hop path between them, they will not be able to use the data channel. They will need a different channel to announce themselves to each other, and possibly exchange administrative information. Accordingly a control channel, over which non-neighbors can communicate directly, is proposed to complement the data channel.

In this section we describe two methods by which the control channel can be implemented. The first is to interleave the data channel and the control channel as shown in Fig. 5-1. In this method, the continuous stream of data slots is interleaved every now and then with a burst of control slots. During the control slots the PRNET ceases to transmit data as usual, and communication over the control slots is limited to administrative communication. The control slots need not be the same length as the data slots. Indeed, one expects them to be shorter since the amount of information carried in a control slot is likely to be relatively small. The number of slots in each burst and the duration between bursts are to be determined and may even be pseudo-random. The disadvantage of this method is that control slots are not continuous in time. The network topology may change considerably between bursts of control slots, so a topological algorithm requiring several control slots may need to make appropriate adjustments.

The second method of implementing a control channel is to have it exist in parallel with the data channel as shown in Fig. 5-2. The control channel is again divided in time into slots. A control slot however can be made much longer than a data slot, to permit low bit-rates over the control channel. This allows signals transmitted over the control channel to have negligibly low power. To overcome interference from signals in the data channel, low-rate error-correction codes can be used. A potential difficulty in the parallel method is that if a node is transmitting in a data slot, it will



**Figure 5-1:** Data channel and control channel interleaved.



**Figure 5-2:** Data channel and control channel in parallel.

not be able to hear incoming signals in the parallel control slot, and vice versa.

We do not address which of the two above methods, or indeed an alternative method, is better to implement the control channel. However we henceforth assume the existence of a control channel that consists of a stream of slots. In the method where control slots occur in bursts, and are therefore non-adjacent, the stream is a virtual one. The control slots in the stream are allotted to different topological algorithms. Therefore, the stream of control slots can be divided into virtual sub-streams, each used by a different topological algorithm. Within each sub-stream the slots are grouped into frames. With each topological algorithm requiring a portion

of the control slots, it is important that the topological algorithms be efficient in the number of slots they need for execution (i.e., use as few slots as possible), in order to be able to accommodate as many topological algorithms as possible. In addition, for some topological algorithms, large delays can be very detrimental.

## 5.2 Hidden Node Discovery Procedure

The *Hidden Node Discovery* (HND) procedure is useful when a lost node or a new node entering the PRNET wants to connect to a node that is already part of the PRNET. (We assume there are other nodes in the network.) The other nodes in the network are hidden to the lost/new node. The purpose of the HND procedure is to allow the lost/new node to discover some of the existing nodes and connect to one of them. We refer to a lost/new node as a *lost node*, and to a node that is already in the PRNET as an *existing node*.

The HND procedure can be used as a building block for other topological algorithms. For example, it can be employed in the self-initialization of the PRNET (i.e., when the PRNET first comes into existence but nodes do not have neighbors). Each node can employ the HND procedure to find another node, agree to become neighbors of each other, and then elect one of them to use the HND procedure to find other nodes or groups of nodes. Similarly, a group of isolated nodes, forming a subnetwork with no path to a supernode/base-station, can elect one of them to assume the role of a lost node and carry out the HND procedure, with the purpose of finding and connecting to a node (outside that subnetwork) that has a path to a supernode/base-station. The role of the lost node can be rotated among the nodes in the lost subnetwork according to some algorithm. The HND procedure can also be used to allow disconnected subnetworks to discover and connect with each other.

To facilitate the development of the HND procedure, we focus on the case of a single lost node trying to connect to an existing node in the PRNET.

The HND frame consists of two slots.<sup>1</sup> In the first slot the lost node transmits a

---

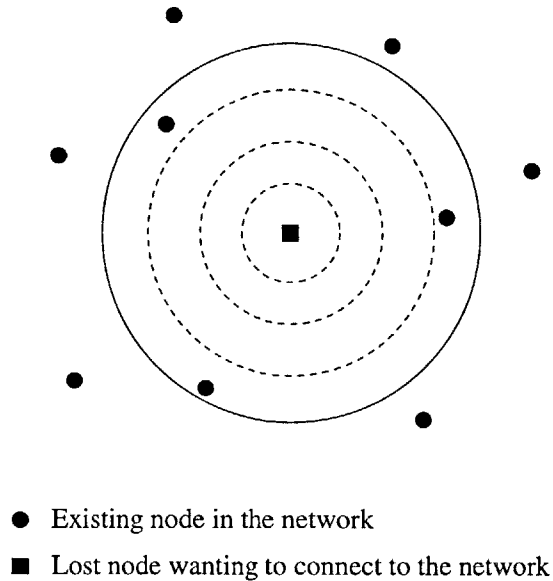
<sup>1</sup>Later we consider increasing the number of slots in the HND frame.

*beacon signal*. If an existing node correctly receives the beacon signal (i.e., receives it at a power level that is above a certain threshold), then we will say that the existing node is covered. The spreading code for the beacon signal is pre-set to allow existing nodes to know what spreading code to look for. The beacon signal contains the UID (user ID) of the lost node and the power level at which it was transmitted. Existing nodes monitor the HND slot designated for the beacon signal. If an existing node is covered, then it will send a *response signal* to the lost node in the second slot of the HND frame, which is monitored by the lost node. The existing node uses the information about power in the beacon signal to set the transmission power of its response signal appropriately. The response signal uses a pre-set spreading code and carries information about how the lost node can communicate with the existing node over the data channel. Such information includes the data slots and the spreading codes to use. At that point the lost node can stop using the control channel, becomes an existing node and can use topological algorithms for existing nodes to find other neighbors and become fully integrated into the PRNET.

There are two problems in the implementation of the HND procedure as described so far. First, it is necessary for an existing node to be covered by the lost node's beacon signal. Second, if two or more existing nodes are covered then they will all transmit response signals which will collide at the lost node.

To overcome the two obstacles, we propose a strategy, in which the lost node starts with an initial coverage area, and repeatedly increases the coverage area until it covers at least one existing node. We refer to this process as increasing scope, and it is depicted in Fig. 5-3. The lost node will know that there are no existing nodes in its coverage area if it does not hear a response signal in the second slot of the HND frame. The lost node may start with a small initial coverage area, incrementing it by small amounts, in order to increase the likelihood that it covers exactly one node (when a it first covers existing nodes).

We briefly analyze the increasing scope process, using a simple model. We assume that when a lost node transmits a beacon signal at some power  $P$ , the beacon signal covers a circular region centered at the lost node, with an area equal to  $A$ . We assume



**Figure 5-3:** A lost node attempts to find other nodes using increasing scope. Eventually its beacon message is heard by some nodes.

that the area  $A$  is a function of  $P$ , and therefore that the lost node can control the coverage area of its beacon signal. Existing nodes inside the circle can correctly receive the beacon signal, whereas existing nodes outside are assumed not to receive the signal at all. We assume that the existing nodes in the PRNET are distributed according to a Poisson process with some rate  $\lambda$ .

The model described above ignores important issues such as the mobility of nodes, the fading nature of the wireless channel, and the possibility of nodes forming physical clusters. However, this model enables us to do a preliminary mathematical analysis of the procedure.

Let  $X$  be the number of existing nodes that are covered by the beacon signal on the first trial that the beacon signal covers existing nodes. In the example given in Fig. 5-3,  $X = 3$ . Also let  $T$  be the number of trials up to and including the trial in which the lost node's beacon signal covers existing nodes. In Fig. 5-3,  $T = 4$ .  $X$  and  $T$  are random variables.

Suppose the strategy the lost node uses is to have an initial transmission area of



$A_0$  and to increase the transmission area by  $A_0$  in each trial. For this strategy,

$$E[X] = \sum_{t=1}^{\infty} E[X|T = t]P_T(t)$$

where,

$$\begin{aligned} E[X|T = t] &= E[X|\text{area} = A_0 \text{ and } X > 0] \\ &= \frac{\lambda A_0}{1 - e^{-\lambda A_0}} \end{aligned}$$

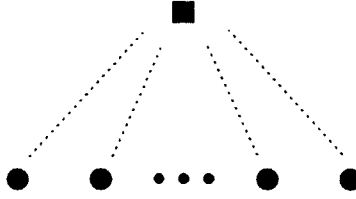
Therefore

$$\begin{aligned} E[X] &= \frac{\lambda A_0}{1 - e^{-\lambda A_0}} \sum_{t=1}^{\infty} P_T(t) \\ &= \frac{\lambda A_0}{1 - e^{-\lambda A_0}} \end{aligned}$$

On the other hand,  $T$  is a discrete random variable with a geometric probability mass function. Therefore,

$$E[T] = \frac{1}{1 - e^{-\lambda A_0}}$$

It is of interest to have  $E[X]$  and  $E[T]$  be as small as possible. Large values of  $T$  are undesirable since they leave the lost node unconnected for an unnecessarily long period. The smaller  $E[X]$  is the more likely the lost node covers exactly one existing node the first time it covers existing nodes. Moreover, the smaller  $E[X]$ , the generally less complicated it is for the lost node to work out which of the covered nodes to connect to. However, there is a trade-off between the values of  $E[X]$  and  $E[T]$ , that is driven by the parameter  $\lambda A_0$ . The larger  $\lambda A_0$ , the smaller  $E[T]$ , but the larger  $E[X]$ . This trade-off is not surprising. The larger the incremental area, the faster one expects the lost node to find existing nodes, but also the more likely it is to have more of them in the covered area. Another trade-off to consider is that between lowering  $E[T]$  and lowering the expected amount of energy used by the lost node in finding existing nodes. For instance, the lost node can minimize  $E[T]$  by blasting its beacon signal at maximum power on the first trial. However, in doing so it may



**Figure 5-4:** Lost soul must filter out one of the nodes that respond.

unnecessarily waste transmission energy, not to mention increase the likelihood of it being intercepted. The design of the increasing scope process depends on the desired balance between the different trade-offs.

Regardless of how the increasing scope process is designed, it is almost certain that it cannot always guarantee that  $X = 1$ . This will be especially true if existing nodes tend to be clustered (as opposed to being distributed according to a Poisson process). Therefore, it is critical for the lost node to have the ability to filter out one of multiple covered existing nodes, when that situation arises. We refer to an algorithm that gives a lost node this ability as a splitting algorithm. In the next section, we propose several splitting algorithms.

## 5.3 Splitting Algorithms for The HND Procedure

The situation that a splitting algorithm needs to deal with is depicted in Fig. 5-4. The lost node must filter out and connect to one out of several covered nodes. We assume that the lost node can detect a collision if it occurs. The situation in Fig. 5-4 is reminiscent of the multiple access problem. The difference here is that it is sufficient for the lost node to receive a signal correctly from one of the covered nodes, as opposed to all. Nevertheless, some of the established random access techniques may prove useful here.

### 5.3.1 MALOHA

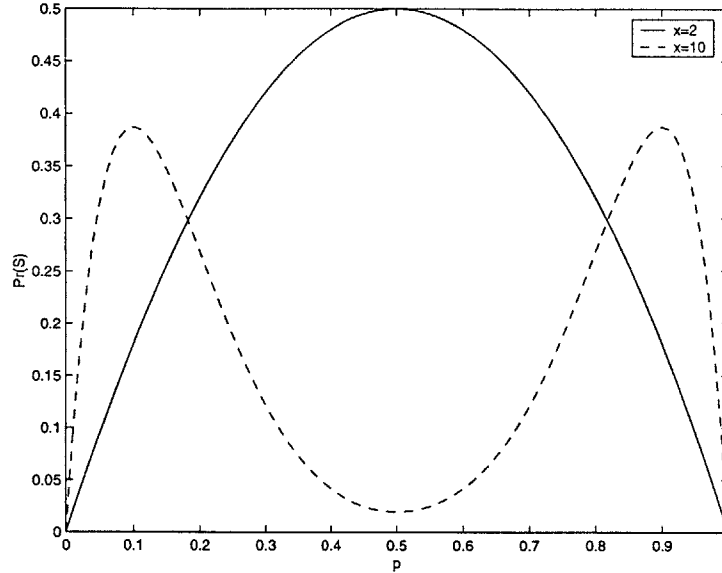
The first splitting algorithm we consider is inspired by the ALOHA random access protocol. We refer to it as Modified ALOHA (MALOHA). In MALOHA the HND

frame is elongated to contain  $K$  slots. The first slot is still reserved for the lost node to transmit its beacon signal. In each of the next  $K - 2$  slots, each covered node transmits a response signal with some probability  $p$ . Any covered node that does not transmit a response signal in those  $K - 2$  slots transmits a response signal in the  $K^{\text{th}}$  slot with probability 1. The reason for forcing a transmission in the last slot is that otherwise the lost node may not hear any response signals, leading it to think that no existing nodes are covered and hence to increase its coverage area in the next trial. Increasing the area likely results in more covered nodes, worsening the contention. We will say the HND frame is successful if in one of its slots exactly one covered node transmits a response signal. If the frame is unsuccessful (due to collisions), in the following frame's first slot the lost node will send a *repeat-response signal* at the same power level as that of its last beacon signal. We assume that the same covered nodes receive the repeat-response signal. Upon receiving it the covered nodes understand that collisions occurred, and repeat their probabilistic transmissions in the same manner as the first frame. The process is repeated until a successful frame occurs. At that point the lost node gets off the control channel, and the covered nodes understand that the lost node connected successfully.

We need to study what values of  $K$  and  $p$  maximize the probability of success (i.e., of a successful frame),  $\Pr(S)$ . For  $K = 3$ ,

$$\Pr(S) = \begin{cases} 2p(1-p) & \text{for } x = 2 \\ xp(1-p)^{x-1} + x(1-p)p^{x-1} & \text{for } x > 2 \end{cases}$$

where  $x$  is the number of contending existing nodes. For  $K = 4$ , the calculation of  $\Pr(S)$  is possible yet cumbersome, and for  $K > 4$  it becomes almost necessary to use approximations. However, just based on analyzing the case for  $K = 3$ , it seems that MALOHA is not a promising approach. Looking at Fig. 5-5 we see that the probability of success for a given  $p$  varies with the number of users. Therefore, for MALOHA to work well the covered nodes need to adjust  $p$  based on the feedback from the lost node. This can be a long process requiring complicated calculations. Therefore, we look to different approaches.



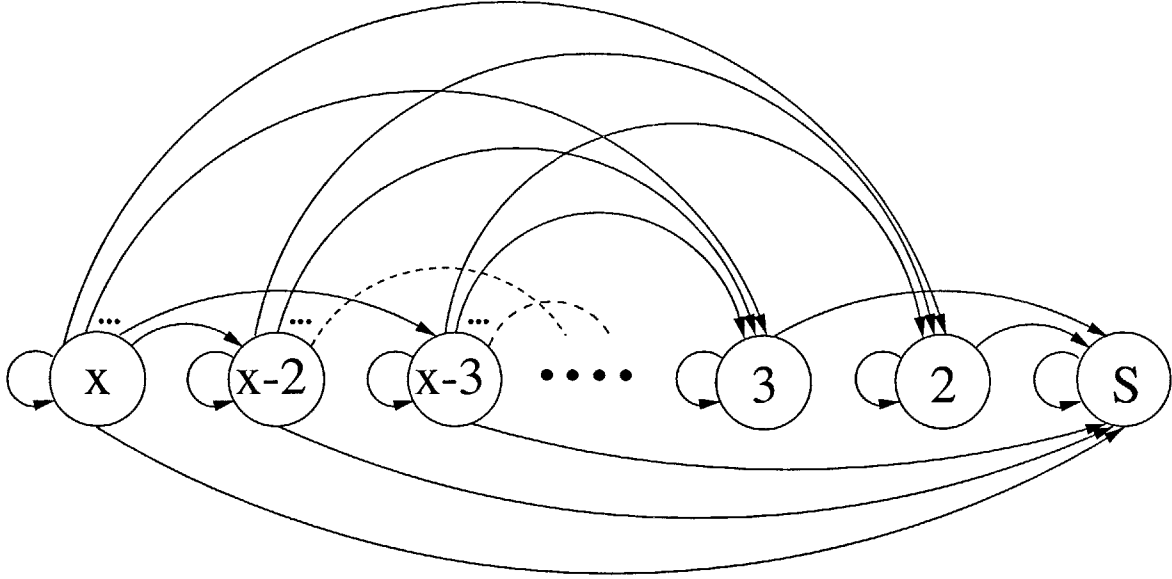
**Figure 5-5:** MALOHA: Comparing  $\Pr(S)$  for the cases of  $x = 2$  and  $x = 10$ , as  $p$  varies from 0 to 1.

### 5.3.2 Uniform Splitting

Like MALOHA, in uniform splitting the HND frame consists of  $K$  slots, with the first slot reserved for transmissions by the lost node. Each covered node selects exactly one of the last  $K - 1$  slots, with equal probability, and transmits its response signal in that slot. As in MALOHA, the frame will be successful if any of the slots has exactly one response signal in it. The difference with uniform splitting is that if the frame is unsuccessful, the lost node will select one of the collision slots at random, and in its repeat-response signal will indicate which slot is chosen. Only the covered nodes that transmitted in that slot transmit again. The idea here is that typically after each trial the number of contending covered nodes decreases. The process is repeated until a successful frame occurs. We want to investigate what  $K$  ought to be, comparing the performance of  $K = 3$  and  $K = 4$ .

#### Binary Uniform Splitting

When  $K = 3$  the covered nodes select among two slots. Therefore, we refer to the case of  $K = 3$  as the binary uniform splitting. The Markov chain in Fig. 5-6 depicts the evolution of binary uniform splitting. Each state in the Markov chain



**Figure 5-6:** Markov chain depicting the evolution of binary and ternary uniform splitting. The transition probabilities for the two cases are different.

is characterized by the number of remaining covered nodes trying to connect to the lost node.  $S$  denotes the event of a successful frame. The transition probabilities are given below:

$$P_{2,2} = \frac{1}{2}, \quad P_{2,S} = \frac{1}{2}$$

$$P_{3,3} = \frac{1}{4}, \quad P_{3,S} = \frac{3}{4}$$

and for  $x \geq 4$ ,

$$P_{x,i} = \begin{cases} \left(\frac{1}{2}\right)^{x-1} & \text{if } i = x \\ \binom{x}{i} \left(\frac{1}{2}\right)^x & \text{if } 2 \leq i \leq x-2 \\ x \left(\frac{1}{2}\right)^{x-1} & \text{if } i = S \\ 0 & \text{otherwise} \end{cases}$$

Let  $F_x$  be the number of frames up to and including the first successful frame, given that the starting number of responding nodes is  $x$ .

$$E[F_x] = 1 + P_{x,x}E[F_x] + \sum_{i=2}^{x-2} P_{x,i}E[F_i] \quad (5.1)$$

Eq. (5.1) above is a difference equation with the initial conditions

$$E[F_2] = 2 \text{ and } E[F_3] = \frac{4}{3}$$

Using the two initial conditions above we can numerically evaluate  $E[F_x]$  for  $x > 3$ , which we do later.

We might also be interested in the variance of  $F_x$ . Accordingly, we evaluate  $E[F_x^2]$ .

$$\begin{aligned} E[F_x^2] &= P_{x,x}E[(1 + F_x)^2] + \sum_{i=2}^{x-2} P_{x,i}(1 + E[F_i])^2 + P_{x,S} \\ &= P_{x,x}(1 + 2E[F_x] + E[F_x^2]) + \sum_{i=2}^{x-2} P_{x,i}(1 + 2E[F_i] + E[F_i^2]) + P_{x,S} \\ &= 1 + 2P_{x,x}E[F_x] + \sum_{i=2}^{x-2} 2P_{x,i}E[F_i] + P_{x,x}E[F_x^2] + \sum_{i=2}^{x-2} P_{x,i}E[F_i^2] \\ &= P_{x,x}E[F_x^2] + \sum_{i=2}^{x-2} P_{x,i}E[F_i^2] + 2E[F_x] - 1 \end{aligned}$$

where  $E[F_2^2] = 6$  and  $E[F_3^2] = \frac{20}{9}$

### Ternary Uniform Splitting

$K = 4$  corresponds to ternary uniform splitting. The potential advantage of ternary uniform splitting is that it probably requires fewer frames on average to resolve the contention.

The Markov chain in Fig. 5-6 also applies to ternary uniform splitting. However the transition probabilities are different. The transition probabilities are

$$\begin{aligned} P_{2,2} &= \frac{1}{3}, & P_{2,S} &= \frac{2}{3} \\ P_{3,3} &= \frac{1}{9}, & P_{3,S} &= \frac{8}{9} \end{aligned}$$

and for  $x \geq 4$ ,

$$P_{x,i} = \begin{cases} 3 \left(\frac{1}{3}\right)^x & \text{if } i = x \\ \left(\frac{1}{3}\right)^x \left[ \frac{3}{2} \mathcal{I}_1(i, 0, x) \binom{x}{i, 0} + \sum_{j=1}^{\lfloor \frac{x-i}{2} \rfloor} \mathcal{I}_1(i, j, x) \binom{x}{i, j} \right] & \text{if } 2 \leq i \leq x-1 \\ \left(\frac{1}{3}\right)^x \sum_{j=0}^{\lfloor \frac{x-1}{2} \rfloor} \mathcal{I}_2(j, x) \binom{x}{1, j} & \text{if } i = S \\ 0 & \text{otherwise} \end{cases}$$

where

$$\mathcal{I}_1(i, j, x) = \begin{cases} 0 & \text{if } j = 1 \text{ or } x - i - j = 1 \\ 1 & \text{if } j = x - i - j \\ 2 & \text{otherwise} \end{cases}$$

and

$$\mathcal{I}_2(j, x) = \begin{cases} 3 & \text{if } j = 1 \text{ or } x - 1 - j = 1 \text{ or } j = x - 1 - j \\ 6 & \text{otherwise} \end{cases}$$

Again, letting  $F_x$  be the number of frames up to and including a successful resolution then, because we have the same Markov chain, the difference equation is the same as in the case of binary uniform splitting, namely

$$E[F_x] = 1 + P_{x,x}E[F_x] + \sum_{i=2}^{x-2} P_{x,i}E[F_i]$$

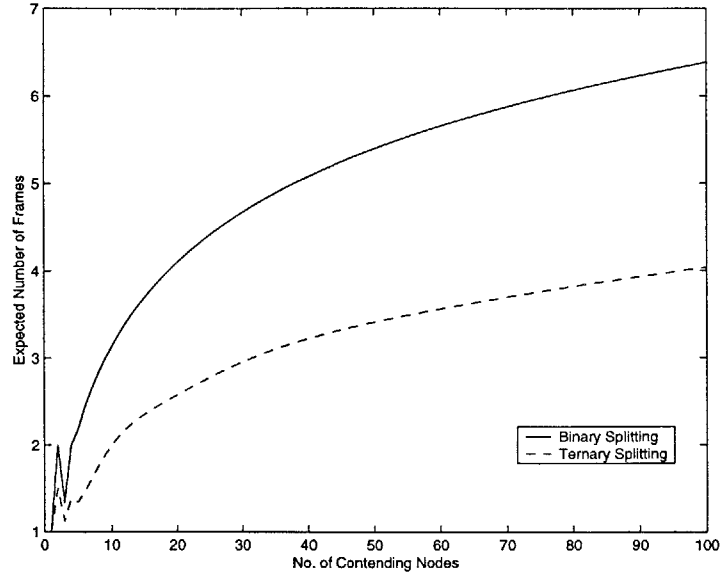
with the different initial conditions of

$$E[F_2] = \frac{3}{2} \text{ and } E[F_3] = \frac{9}{8}$$

Similarly,

$$E[F_x^2] = P_{x,x}E[F_x^2] + \sum_{i=2}^{x-2} P_{x,i}E[F_i^2] + 2E[F_x] - 1$$

where  $E[F_2^2] = 3$  and  $E[F_3^2] = \frac{45}{32}$ .



**Figure 5-7:** The expected number of frames until successful resolution versus number of contending users.

### Comparison of Binary and Ternary Uniform Splitting

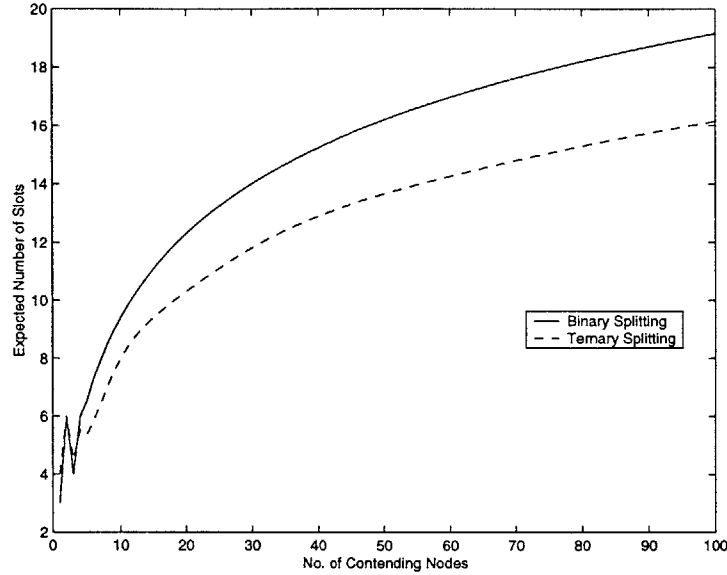
Fig. 5-7 compares the expected number of frames until successful resolution among the covered nodes under binary and ternary uniform splitting. We note however that the critical resource is the control slot. Therefore to make the comparison more meaningful, we compare the expected number of slots until a successful resolution in Fig. 5-8.

If the number of contending users is less than 4, then binary uniform splitting will do better than ternary uniform splitting in terms of the expected number of slots needed. For  $x \geq 4$ , ternary uniform splitting outperforms binary uniform splitting. In any case, with an appropriate increasing scope strategy we expect the number of contending users to typically be fewer than ten. Over that range, binary uniform splitting and ternary uniform splitting are comparable.

The expected number of slots until a successful resolution is a good first-order measure for comparing the different splitting schemes. In Subsection 5.3.5 we propose other performance measures for comparing the different splitting schemes, that take the variance in the number of slots into account.

First we want to address an interesting question that presents itself. What if





**Figure 5-8:** Comparing the expected number of slots until successful resolution versus number of contending users.

the lost node is able to determine the slot that has the fewest respondents? This is likely to remain a hypothetical question, as the determination of the slot with the fewest respondents is likely to be a daunting task in terms of the hardware and the processing. Nevertheless, it is likely that there is an advantage in selecting that slot.

### 5.3.3 Minimum Splitting

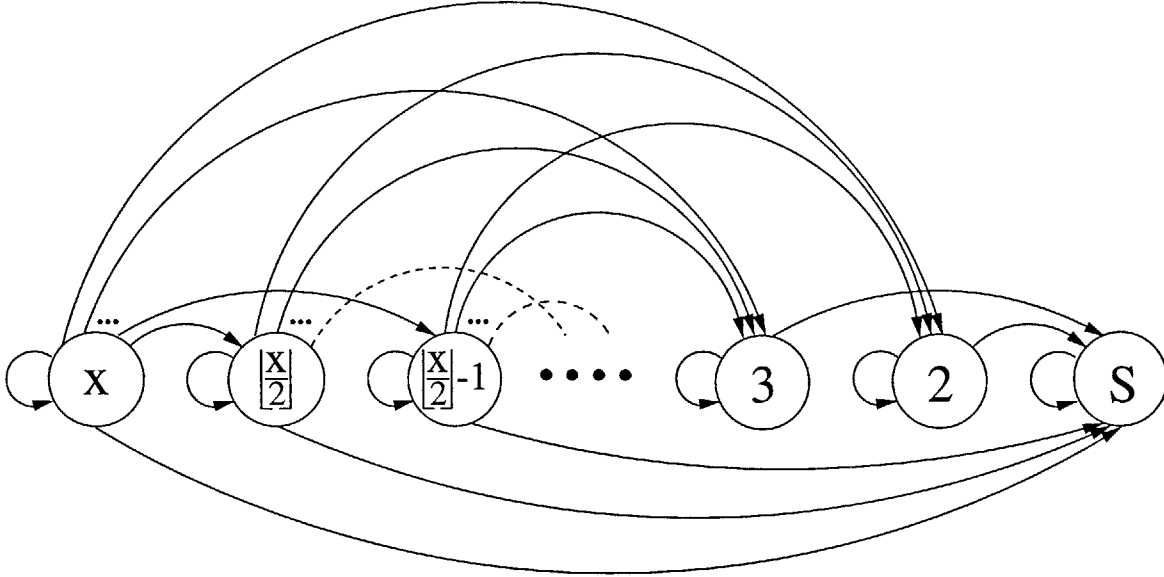
Minimum splitting is a variation of uniform splitting, where we assume that the lost node has the capability to determine which slot has the fewest response signals. Below we analyze binary and ternary minimum splitting.

#### Binary Minimum Splitting

In this case the Markov chain is modified as shown in Fig. 5-9. The new transition probabilities are as follows:

$$P_{2,2} = \frac{1}{2}, \quad P_{2,S} = \frac{1}{2}$$

$$P_{3,3} = \frac{1}{4}, \quad P_{3,S} = \frac{3}{4}$$



**Figure 5-9:** Markov chain depicting the evolution of binary and ternary minimum splitting.

and for  $x \geq 4$ ,

$$P_{x,i} = \begin{cases} = \frac{1}{2}^{x-1} & \text{if } i = x \\ \binom{x}{i} \frac{1}{2}^x & \text{if } i = \frac{x}{2} \\ \binom{x}{i} \frac{1}{2}^{x-1} & \text{if } i \neq \frac{x}{2} \text{ and } 2 \leq i \leq \lfloor \frac{x-i}{2} \rfloor \\ x \frac{1}{2}^{x-1} & \text{if } i = S \\ 0 & \text{otherwise} \end{cases}$$

The expected value of  $F_x$  is

$$E[F_x] = 1 + P_{x,x}E[F_x] + \sum_{i=2}^{\lfloor \frac{x}{2} \rfloor} P_{x,i}E[F_i]$$

where  $E[F_2] = 2$  and  $E[F_3] = \frac{4}{3}$ . The second moment of  $F_x$  is given by

$$E[F_x^2] = P_{x,x}E[F_x^2] + \sum_{i=2}^{\lfloor \frac{x}{2} \rfloor} P_{x,i}E[F_i^2] + 2E[F_x] - 1$$

where  $E[F_2^2] = 6$  and  $E[F_3^2] = \frac{20}{9}$ .

### Ternary Minimum Splitting

The Markov chain in 5-9 also depicts the evolution of ternary minimum splitting, with the following transition probabilities.

$$P_{2,S} = \frac{2}{3}, \quad P_{2,2} = \frac{1}{3}$$

$$P_{3,S} = \frac{8}{9}, \quad P_{3,3} = \frac{1}{9}$$

and for  $x \geq 4$ ,

$$P_{x,i} = \begin{cases} 3 \left(\frac{1}{3}\right)^x & \text{if } i = x \\ \left(\frac{1}{3}\right)^x \left[ \mathcal{I}_3(i, x) \binom{x}{i} + \sum_{j=i}^{\lfloor \frac{x-i}{2} \rfloor} \mathcal{I}_4(i, j, x) \binom{x}{i, j} \right] & \text{if } 2 \leq i \leq \lfloor \frac{x}{2} \rfloor \\ \left(\frac{1}{3}\right)^x \sum_{j=0}^{\lfloor \frac{x-1}{2} \rfloor} \mathcal{I}_2(j, x) \binom{x}{1, j} & \text{if } i = S \\ 0 & \text{otherwise} \end{cases}$$

where as before,

$$\mathcal{I}_2(j, x) = \begin{cases} 6 & \text{if } j \neq x-1-j \neq 1 \\ 3 & \text{if } j = 1 \text{ or } x-1-j = 1 \\ 3 & \text{if } j = x-1-j \end{cases}$$

and

$$\mathcal{I}_3(i, x) = \begin{cases} 3 & \text{if } i = x-i \\ 6 & \text{otherwise} \end{cases}$$

and

$$\mathcal{I}_4(i, j, x) = \begin{cases} 1 & \text{if } i = j = x-i-j \\ 6 & \text{if } i \neq j \neq x-i-j \\ 3 & \text{otherwise} \end{cases}$$

The expressions for  $E[F_x]$  and  $E[F_x^2]$  are the same as in the case of binary minimum splitting, but with different initial conditions:  $E[F_2] = \frac{3}{2}$ ,  $E[F_3] = \frac{9}{8}$ ,  $E[F_2^2] = 3$ , and

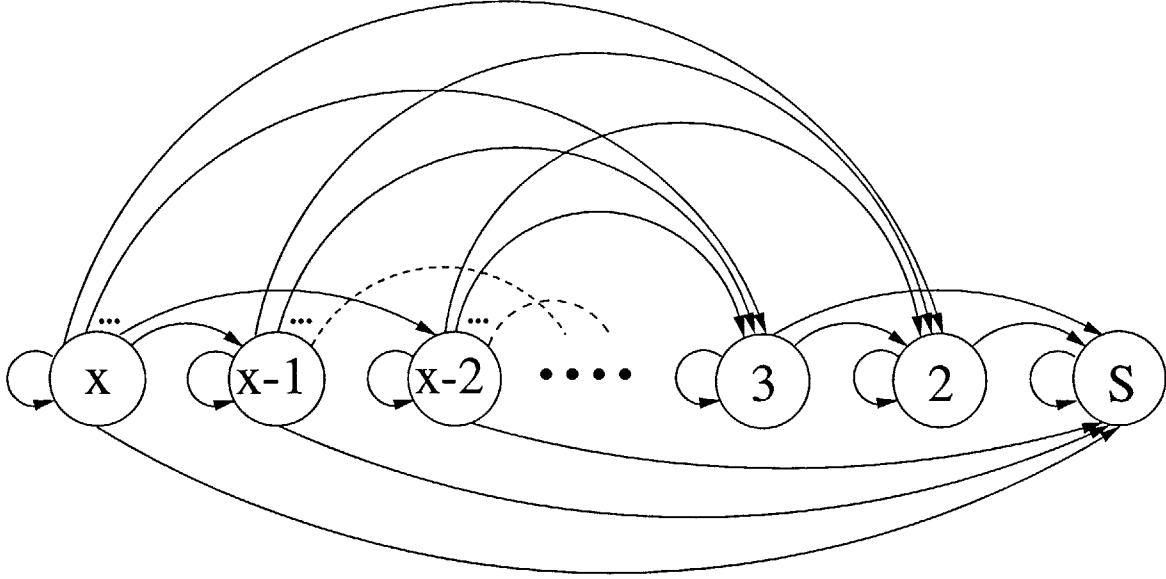
$$E[F_3^2] = \frac{45}{32}.$$

### 5.3.4 Single-Slot Splitting

In the single-slot splitting algorithm, the HND frame consists of two slots. As before, in the first slot the lost node transmits its beacon signal. The first time an existing node hears the beacon signal, it transmits a response signal in the second slot. This ensures that the lost node knows that there is at least one covered node. If a collision occurs, then the lost node will send a repeat-response signal. Upon hearing the repeat-response signal, a covered node transmits a response signal, with probability  $1/2$ . If, as a result, no covered node transmits a response signal, the lost node will send a repeat-response signal in the next frame indicating an idle slot, and each covered node again will transmit with probability  $1/2$ . On the other hand, if a collision occurs, then the lost node will transmit a repeat-response signal, indicating the collision. This is when the splitting occurs. Only covered nodes that transmitted response signals in the previous frame attempt to retransmit, again with probability  $1/2$ . Therefore with each trial, on average half the remaining contending nodes drop out. The process continues as described above, until a single covered node transmits a response signal. Although the number of frames required to filter out one covered node may typically be larger than those in uniform and minimum splitting, the motivation behind this scheme is the recognition that the cost is measured in slots as opposed to frames. In this scheme there are only two slots per frame, as opposed to three in binary splitting and four in ternary splitting, which may compensate for the possibly larger typical number of frames.

The Markov chain in Fig. 5-10 depicts the evolution of single-slot splitting, where each state is characterized by the number of contending covered nodes. The transition probabilities for the Markov chain are:

$$P_{2,2} = \frac{1}{2}, \quad P_{2,S} = \frac{1}{2}$$



**Figure 5-10:** Markov chain depicting the evolution of single-slot splitting.

and for  $x \geq 3$ ,

$$P_{x,i} = \begin{cases} \left(\frac{1}{2}\right)^{x-1} & \text{if } i = x \\ \binom{x}{i} \left(\frac{1}{2}\right)^x & \text{if } 2 \leq i \leq x-1 \\ x \left(\frac{1}{2}\right)^x & \text{if } i = S \end{cases}$$

The calculation of the expected number of frames is slightly different in this case than in uniform and minimum splitting. There is a fixed cost of one frame, which is the first frame in which all the nodes transmit. Therefore,

$$F_x = 1 + G_x$$

where  $G_x$  is the number of frames, from the first frame in which covered nodes transmit probabilistically, up to and including the successful frame.

$$E[G_x] = 1 + \sum_{i=2}^x P_{x,i} E[G_i] \quad (5.2)$$

where,

$$E[G_2] = 2$$

$E[F_x]$  can be simply computed as follows

$$E[F_x] = 1 + E[G_x]$$

As for the second moment,

$$E[G_x^2] = \sum_{i=2}^x P_{x,i} E[G_i^2] + 2E[G_x] - 1$$

where,  $E[G_2^2] = 6$ . Again,  $E[F_x^2]$  can be derived simply as follows

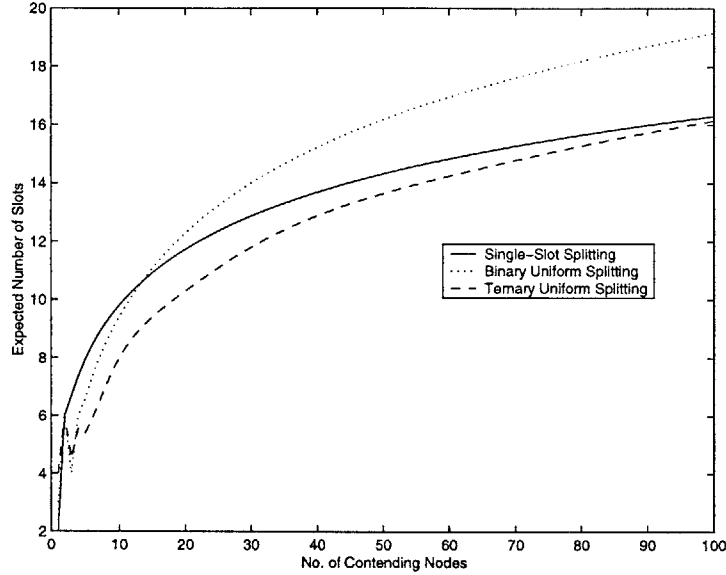
$$E[F_x^2] = 1 + 2E[G_x] + E[G_x^2]$$

Fig. 5-11 uses the expected number of slots until successful resolution as the performance measure to compare single-slot splitting with binary and ternary uniform splitting. Fig. 5-11 suggests that if the increasing scope process can be designed such that  $X$  is typically 1, then single-slot splitting will be the most promising splitting algorithm. Fig. 5-11 also indicates that only for a relatively large number of contending nodes (around 15) is single-slot splitting better than binary uniform splitting. On the other hand, for extremely large numbers of contending users, single-slot splitting approaches the performance of ternary uniform splitting.

Again, the expected number of slots provides a first-order comparison between the splitting schemes. In the next subsection, we propose what we believe to be more meaningful performance measures to compare the different splitting schemes.

### 5.3.5 Comparing The Different Splitting Schemes

Perhaps the most meaningful question to ask is how many slots the splitting algorithm *typically* needs to complete the splitting. Certainly that number depends on the expected number of slots needed to complete the splitting, but it probably also depends on the variance. In addition, it depends on the number of covered nodes that the lost node needs to split.



**Figure 5-11:** Comparing single-slot splitting with binary and ternary uniform splitting.

To make sense of all of that, we propose to model the number of covered nodes as a random variable  $X$  with a conditional Poisson PMF, characterized by the expected value of  $X$ , namely  $E[X]$ . By that we mean, given  $E[X]$ , let  $E[Y]$  be such that

$$E[X] = \frac{E[Y]}{1 - e^{-E[Y]}}$$

where  $Y$  is a discrete random variable with a Poisson PMF. Then,

$$P_X(x) = \begin{cases} 0 & \text{if } x = 0 \\ P_Y(x|Y > 0) & \text{otherwise} \end{cases}$$

Since we do not know what an appropriate value for  $E[X]$  is, we propose to look at the range  $1 < E[X] \leq 15$ . We do not start with 1 because there is at least one covered node, implying that the expected value ought to be greater than 1. If the increasing scope process is designed to cover one or two existing nodes most of the time, then we will expect  $E[X]$  to be less than 2. In any case, we believe that  $E[X]$  is not larger than 10. Therefore, we consider the range  $11 \leq E[X] \leq 15$  only for the sake of being conservative.

Let  $S$  be the random variable representing the number of slots required for suc-

cessful splitting. To compare the different schemes, we propose to use a linear combination of  $E[S]$  and  $\sigma_S$  as the performance measure.  $E[S]$  is calculated in the following manner,<sup>2</sup>

$$E[S] = \sum_{x=1}^{\infty} E[S|x]P_X(x)$$

Similarly,

$$E[S^2] = \sum_{x=1}^{\infty} E[S^2|x]P_X(x)$$

We conservatively look at three versions of the linear combination of  $E[S]$  and  $\sigma_S$ . Table 5.1 uses  $E[S] + \sigma_S$  to compare the different splitting algorithms. Table 5.2 uses  $E[S] + 1.5\sigma_S$ , and Table 5.3 uses  $E[S] + 2\sigma_S$ . Tables 5.1 - 5.3 indicate that if the increasing scope process produces a number of covered nodes whose average is very close to 1, then single-slot splitting will clearly have the best performance. For  $2 \leq E[X] \leq 10$ , all three versions of the performance measure agree that ternary uniform splitting performs best. For  $11 \leq E[X] \leq 15$  ternary minimum splitting performs best. It is surprising, however, that minimum splitting does not perform better than uniform splitting throughout.

Given the complexity involved in implementing ternary minimum splitting, the fact that it performs best for what we expect to be unlikely values for  $E[X]$ , and that even for those values the performance of ternary uniform splitting is comparable, we discount ternary minimum splitting as an option for the splitting algorithm. Overall, it seems that ternary uniform splitting is the most promising splitting algorithm. Even for small values of  $E[X]$ , it has comparable performance to single-slot splitting. However, if the increasing scope process does produce mostly one or two covered nodes, such that  $E[X]$  is close to one, then single-slot splitting apparently will be the best option. Moreover, performances of binary uniform splitting and single-slot splitting are generally comparable to ternary uniform splitting. Therefore, if either binary uniform splitting or single-slot splitting is simplest to implement in practice, then it will be acceptable to use it for splitting.

---

<sup>2</sup>Actually for the purposes of numerical calculations, we set  $\Pr(X = x) = 0$  for  $x > 100$ . This is quite adequate, because for the values of  $E[X]$  we are interested in  $\Pr(X = x)$  is very close to zero for  $x > 100$ .



$E[X]$	$E[S] + \sigma_S$				
	Binary Uniform Splitting	Binary Minimum Splitting	Ternary Uniform Splitting	Ternary Minimum Splitting	Single- Slot Splitting
1.2	5.5144	5.5149	5.8876	5.8878	4.6945 *
1.5	6.6896	6.6973	6.7259	6.7287	6.3062 *
1.8	7.3355	7.3620	7.1453 *	7.1557	7.3331
2	7.6482	7.6925	7.3271 *	7.3456	7.8516
3	8.7618	8.8933	7.8568 *	7.9427	9.5060
4	9.6714	9.7815	8.2887 *	8.4590	10.4689
5	10.4948	10.4628	8.7813 *	9.0243	11.1536
6	11.2299	11.0024	9.3190 *	9.6050	11.6933
7	11.8787	11.4678	9.8596 *	10.1478	12.1429
8	12.4514	11.9002	10.3686 *	10.6148	12.5299
9	12.9602	12.3104	10.8271 *	10.9910	12.8704
10	13.4164	12.6915	11.2296 *	11.2806	13.1745
11	13.8292	13.0335	11.5802	11.4993 *	13.4496
12	14.2060	13.3317	11.8873	11.6668 *	13.7007
13	14.5526	13.5883	12.1608	11.8019 *	13.9317
14	14.8733	13.8116	12.4094	11.9203 *	14.1455
15	15.1719	14.0123	12.6405	12.0331 *	14.3446

**Table 5.1:** Comparison of different splitting schemes using the parameter  $E[S] + \sigma_S$ . \* indicates the best option.

$E[X]$	$E[S] + 1.5\sigma_S$				
	Binary Uniform Splitting	Binary Minimum Splitting	Ternary Uniform Splitting	Ternary Minimum Splitting	Single- Slot Splitting
1.2	6.5263	6.5270	6.6700	6.6702	5.6810 *
1.5	8.0517	8.0622	7.7794	7.7833	7.6692 *
1.8	8.8597	8.8955	8.3187 *	8.3331	8.8701
2	9.2426	9.3023	8.5489 *	8.5744	9.4570
3	10.5645	10.7394	9.2045 *	9.3218	11.2424
4	11.6015	11.7502	9.7259 *	9.9564	12.2291
5	12.5093	12.4830	10.3093 *	10.6355	12.9181
6	13.2963	13.0370	10.9310 *	11.3117	13.4585
7	13.9744	13.5069	11.5380 *	11.9190	13.9083
8	14.5624	13.9466	12.0911 *	12.4156	14.2954
9	15.0788	14.3678	12.5722 *	12.7903	14.6358
10	15.5386	14.7590	12.9805 *	13.0555	14.9399
11	15.9531	15.1056	13.3258	13.2364 *	15.2150
12	16.3307	15.4010	13.6220	13.3615 *	15.4661
13	16.6776	15.6485	13.8832	13.4564 *	15.6971
14	16.9985	15.8588	14.1213	13.5411 *	15.9110
15	17.2971	16.0458	14.3450	13.6290 *	16.1101

**Table 5.2:** Comparison of different splitting schemes using the parameter  $E[S] + 1.5\sigma_S$ .

$E[X]$	$E[S] + 2\sigma_S$				
	Binary Uniform Splitting	Binary Minimum Splitting	Ternary Uniform Splitting	Ternary Minimum Splitting	Single- Slot Splitting
1.2	7.5382	7.5391	7.4524	7.4527	6.6674 *
1.5	9.4137	9.4270	8.8329 *	8.8380	9.0322
1.8	10.3838	10.4290	9.4921 *	9.5106	10.4070
2	10.8370	10.9122	9.7706 *	9.8031	11.0624
3	12.3673	12.5855	10.5522 *	10.7008	12.9787
4	13.5316	13.7189	11.1631 *	11.4539	13.9894
5	14.5239	14.5032	11.8373 *	12.2467	14.6826
6	15.3627	15.0716	12.5430 *	13.0184	15.2237
7	16.0700	15.5459	13.2164 *	13.6903	15.6737
8	16.6734	15.9930	13.8135 *	14.2165	16.0608
9	17.1974	16.4253	14.3173 *	14.5896	16.4012
10	17.6609	16.8265	14.7314 *	14.8303	16.7053
11	18.0771	17.1777	15.0713	14.9735 *	16.9804
12	18.4554	17.4704	15.3566	15.0562 *	17.2315
13	18.8026	17.7087	15.6056	15.1109 *	17.4625
14	19.1237	17.9060	15.8331	15.1619 *	17.6764
15	19.4224	18.0792	16.0496	15.2250 *	17.8756

Table 5.3: Comparison of different splitting schemes using the parameter  $E[S] + 2\sigma_S$ .

Finally, we note that a back-of-the-envelope calculation reinforces our conclusion that ternary splitting does better than binary splitting. Given  $x$  nodes, if we split them (deterministically) by a half each time, then we will need  $\log_2 x$  splits to reduce them to 1. Similarly, if we take a fraction  $1/b$  of the nodes each time, we will need  $\log_b x$  splits to reduce the nodes to 1. The way uniform splitting is implemented is such that each split costs  $b + 1$  slots. Therefore, what we want to minimize is  $(b + 1) \log_b x$ , which is minimized at  $b = 3.59$ . This implies that ternary splitting is better than binary splitting (3.59 is closer to 3 than 2). However, the result also implies that quaternary splitting might perform better than ternary splitting. This point requires further investigation.

## 5.4 Summary

In this chapter we investigated the Hidden Node Discovery (HND) procedure, which is a neighbor management procedure to be employed by lost nodes aiming to connect to an existing node in the PRNET. The HND procedure can also be used as a building block for other topological algorithms. We proposed that the lost node emit a beacon signal in increasing strengths, until existing nodes in the PRNET come under the coverage of the beacon signal. If there are multiple covered nodes, then the lost node will need to filter out one of them. We proposed several splitting algorithms for this purpose, and found one in particular, the ternary uniform splitting algorithm, to generally be the most promising, although the performances of binary uniform splitting and single-slot splitting are comparable.

# Chapter 6

## Conclusion

In this chapter we summarize the contributions of this work and outline directions for future research.

### 6.1 Thesis Contributions

Our goal for this work was to contribute to the understanding of the design of low-energy PRNETs by studying the problem of minimum-energy communication over a PRNET. Given the important design consideration of having the PRNET be resistant to jamming, intercept, and listening-in, we decided to use DS-CDMA as the main component of the medium access scheme.

Our methodology was to establish a mathematical framework for studying minimum-energy communication over a PRNET in order to gain insight into the issues and trade-offs involved in the problem. Our first insight was that minimum-energy communication over a PRNET involves solving three problems. The first is the topological organization problem, through which the links in the PRNET are established. The second problem, minimum-energy routing, requires finding paths, along those links, from sources to destinations and determining the flow rates on the paths. The third problem is the scheduling problem, which is how to ensure that no node transmits and receives simultaneously.

The three components, routing, scheduling, and topological organization, are in-

tertwinced. We concluded that the joint routing-scheduling-topological problem is a complex problem. Therefore, we proposed that the joint problem be decomposed into the three components. Solving for each component separately, although very likely to yield a sub-optimal solution, simplifies the overall problem, enabling us to gain insight into minimum-energy communication over a PRNET.

We argued that the topological organization problem is likely to be influenced by factors outside the scope of the problem of minimum-energy communication. However, we decided that it is of benefit to the overall goal of low-energy PRNETs to put an upper bound on the number of neighbors each node has. Having an upper bound on the number of neighbors implies that nodes have small transmission radii, and therefore use low transmission power levels, and creating low interference for other nodes.

We focused on how to best decompose the routing and scheduling components, and concluded that it is better to solve the scheduling problem first, and then solve the minimum-energy routing problem based on the solution to the scheduling problem. Such a decomposition allows the minimum-energy routing algorithm the chance to avoid using links that have high interference with each other. We proposed several practical scheduling algorithms that are based on having an upper bound on the number of neighbors, as well as two performance measures to help compare the different scheduling algorithms.

The minimum-energy routing formulation we developed does not require that we use one of the scheduling algorithms we proposed. It works with any schedule consisting of recurring identical frames. In formulating the minimum-energy routing problem, our aim was to apply the optimal routing approach, which in the past had been applied to minimum-delay routing in wireline networks. First we looked at minimum-energy routing in static PRNETs. An immediate challenge was to find an appropriate global cost function that reflects the goal of minimum-energy routing and

at the same time is convex. That global cost function is

$$Z = \sum_{(i,j)} Z_{ij}$$

where  $Z_{ij}$  is the cost associated with Link  $(i, j)$ . It is

$$Z_{ij} = \sum_{k=1}^K f_{ij}[k] \frac{P_{ij}[k]}{c_{ij}[k]}$$

where  $f_{ij}[k]$ ,  $P_{ij}[k]$ , and  $c_{ij}[k]$  are the flow rate, nominal transmission power level, and the nominal data rate, respectively, for transmissions over Link  $(i, j)$  in Slot  $k$  of the frame.

It became apparent that pure minimum-energy routing has no mechanism for preventing congestion in the network. Hence for minimum-energy routing to be meaningful, minimum-delay routing needs to be incorporated into it. Accordingly, we formulated the minimum-combined-energy-and-delay (MCED) routing problem by constructing a cost function that is a linear combination of the convex cost functions reflecting energy and delay. Accordingly, in MCED routing

$$Z_{ij} = w_1 \sum_{k=1}^K f_{ij}[k] \frac{P_{ij}[k]}{c_{ij}[k]} + w_2 \sum_{k=1}^K \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]}$$

where  $w_1$  and  $w_2$  are weights reflecting the relative importance of energy and delay. In addition, we concluded that it may be beneficial to adjust the nominal data rates occasionally, in response to the levels of congestion on the different links.

We extended the MCED routing formulation to dynamically varying PRNETs. The extension relies on the idea of separating events that cause the PRNET to be dynamic into different frequencies of occurrence. The frequencies of occurrence of some of the events are low enough that they can be ignored by the MCED routing algorithm. However we argued that changes in the nominal transmission power levels are visible to the MCED routing algorithm. Under certain ergodicity and stationarity assumptions, the following link cost function accounts for changes in nominal

transmission power levels, without requiring the MCED routing algorithm to adjust the link flow rate at the same frequency of those changes:

$$Z_{ij} = w_1 \sum_{k=1}^K f_{ij}[k] \frac{E[P_{ij}[k]]}{c_{ij}[k]} + w_2 \sum_{k=1}^K \frac{f_{ij}[k]}{\frac{c_{ij}[k]}{K} - f_{ij}[k]}$$

Finally, given the importance of neighbors to the scheduling algorithms we proposed, we investigated several approaches to a neighbor management procedure and analyzed their performance. The neighbor management procedure deals with how a new node entering the PRNET can seek out and establish a connection with an existing node in the PRNET. Through this investigation we recognized that a control channel is needed to carry out some of the network management algorithms, and proposed some of ways in which the control channel may be realized.

## 6.2 Directions for Future Research

Perhaps the most interesting problem for future research is to try to develop a practical MCED routing algorithm based on our results. We hypothesize that for a routing algorithm to be practical for PRNETs, it should be of the shortest-path type, because optimal routing may be computationally too cumbersome to implement. If a shortest-path routing algorithm is adopted, an appropriate distance metric will be needed for the link lengths. Although a link may be enabled in several slots in the frame,<sup>1</sup> it is useful to have a single consolidated link length associated with each link. We briefly propose such a link length. First, we note that if  $(i, j)$  is enabled in slots  $k$  and  $k'$  then according to optimal routing Node  $i$  will need to set the flow rates such that

$$\left. \frac{\partial Z_{ij}[k]}{\partial f_{ij}[k]} \right|_{f_{ij}[k]=f_k} = \left. \frac{\partial Z_{ij}[k']}{\partial f_{ij}[k']} \right|_{f_{ij}[k']=f_{k'}}$$

---

<sup>1</sup>When Link  $(i, j)$  is enabled, Node  $i$  may transmit to Node  $j$ , although it does not necessarily have to.



We assume that Node  $i$  can easily equate the two first derivatives, since it is a local operation.<sup>2</sup> Let  $f_{ij}[k]$  and  $f_{ij}[k']$  be the outcome of equating the derivatives, then the consolidated link length we propose is

$$C_{ij} = \frac{f_{ij}[k]Z_{ij}[k] + f_{ij}[k']Z_{ij}[k']}{f_{ij}[k] + f_{ij}[k']}$$

The above proposal generalizes for any number of slots over which  $(i, j)$  is enabled.

A second interesting problem for future research is power control. Power control has received much attention in the literature for application to cellular networks. We submit that power control in PRNETs is a harder problem, especially if it needs to observe the requirement that nodes transmit conservatively (i.e., overcome potential interference). The complication with overcoming potential interference is that if a node adjusts its nominal transmission power based on past interference levels, then the power control algorithm will face the interesting problem of estimating the worst-case interference level based on previously *measured* interference levels (which may not include the worst-case interference level). The problem will be even more interesting if instead of setting the nominal transmission power level to overcome the worst possible interference (which is not likely to happen often), we require the nominal transmission power level to overcome (within some probability) the actual interference to be experienced.

We suggested that the nominal data rates be adjusted occasionally to react to the levels of congestion on different links. However, the problem of congestion is already dealt with using flow control, which avoids congestion by regulating the amount of traffic entering the network. If the congestion problem is also to be handled by adjusting the nominal data rates, it will be interesting to study how the two mechanisms might interact with each other.

We proposed two methods for implementing the control channel. We did not study in detail the advantages and disadvantages of each method. Comparing the two methods, investigating in more detail how they can be implemented, and proposing

---

<sup>2</sup>As explained in Chapter 3, the two first derivatives will be equal if there is flow in both slots.

different methods for implementing the control channel, are some of the issues that need to be studied.

In terms of topological algorithms, we only scratched the surface with the Hidden Node Discovery (HND) procedure investigated in Chapter 5. One important issue that needs to be resolved is the criteria that a node uses in assessing the value of another node as a neighbor. Assigning values to other nodes can help a node determine which neighbors to keep, which neighbors to drop, and which nodes to seek to become a neighbor of. We believe that the criteria for evaluating a node include the path propagation loss and the accessibility of certain destination nodes and/or supernodes from that node. Associated with the problem of selecting new neighbors is determining what the initial nominal data rate for the new link ought to be. Another topological problem is how a group of isolated nodes can coordinate with each other to locate and connect with another group of nodes that preferably have access to a supernode. Topological problems will be even more interesting if GPS technology is assumed, and nodes can broadcast their geographic locations.

In the HND procedure, we suggested some ways of filtering out multiple responding nodes. Further investigation of this topic is needed in conjunction with the algorithm for first finding such a set of nodes.

Finally, there are some parameters whose values, we believe, can be chosen through experimentation. The first parameter is  $M$ , the optimal upper limit on the number of neighbors. It is likely that this number depends on certain characteristics of the PRNET, such as the geographic distribution of the nodes. In addition, appropriate values for the weights used in the MCED routing cost function,  $w_1$  and  $w_2$ , need to be found. It is likely that simulations using different values for the weights under different conditions are the most convenient way to find appropriate values for  $w_1$  and  $w_2$ .

# Appendix A

## Initial Approach: Mathematical Programming

In this appendix we describe an initial approach that we attempted for solving the problem of minimum-energy communication over a PRNET. Since the goal is to minimize the aggregate transmission energy (over a period of time) for a network while satisfying given end-to-end rates, it seemed natural to formulate the problem as a mathematical programming problem; whose solution is “optimal” in the sense no assumptions, such as limiting the number of neighbors, are made to reduce the complexity of the optimization problem and contend with a sub-optimal solution. We assumed having a centralized algorithm with global knowledge of the PRNET. The hope of this approach was to find the optimal solution for a static network (i.e., fixed end-to-end rates and an unchanging wireless channel), and use the solution to gain insight into how to construct fast and good heuristic methods for dynamic PRNETs.

Unfortunately, the mathematical programming approach proved to be unfruitful. The purpose of this appendix is to explain the difficulties involved in this approach. In particular, we want to show why scheduling makes the problem so complex that we need to consider decomposing the problem.

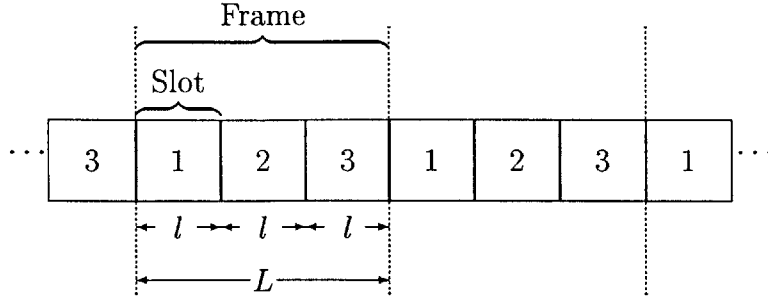
## A.1 PRNET Model for Mathematical Programming

The PRNET model used in this appendix is slightly different from the one developed in Chapter 1. In this section we go over the differences. One main difference is that we do not limit the number of neighbors a node has, so the network topology is described by a complete symmetric digraph [ChL96]. A complete symmetric digraph has both links  $(i, j)$  and  $(j, i)$  for every pair of nodes  $i$  and  $j$ . The assumption here is that every node can adjust its transmission power to reach any other node, therefore we allow a link to exist between every pair of nodes. If a destination node, Node  $j$ , happens to be absolutely outside the transmission range of a source node, Node  $i$ , then the path propagation loss for that link,  $\alpha_{ij}$  will be set to zero, such that the power received at Node  $j$ ,  $\alpha_{ij}G_{ij}$ , is zero. Accordingly, in this model the topological organization problem is a non-problem, as it is predetermined that all the nodes are neighbors of each other.

There is also a difference in how we describe end-to-end rates. Here we have  $S$  end-to-end sessions among the  $N$  nodes. Associated with Session  $s$  ( $1 \leq s \leq S$ ), there is a required average end-to-end transmission rate of  $r^s$  bits/second. Furthermore, if Node  $i$  is the origin node of Session  $s$ , and Node  $j$  is the destination node of Session  $s$  then we will define:

- $O(s) = i$
- $D(s) = j$

A traffic vector summarizes the required steady-state end-to-end data rates. We denote the traffic vector by  $\mathbf{r} = [r^1 \ r^2 \ \dots \ r^S]^T$ , where  $r^s$  is the end-to-end data rate for Session  $s$ . In a multi-hop network, several sessions may be using a particular link as a hop. In that case, the link carries bits belonging to all those sessions. We subsequently refer to bits belonging to Session  $s$  as Type  $s$  bits. In the same vein, if  $v_{ij}$  be the total bit-rate on  $(i, j)$ , then  $v_{ij} = \sum_{s=1}^S x_{ij}^s$ , where  $x_{ij}^s$  is the bit-rate of Type  $s$  bits on  $(i, j)$ .

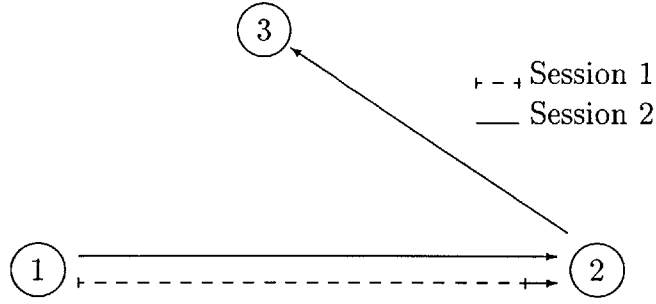


**Figure A-1:** Illustration of frames and slots.

The channel access scheme is still slotted DS-CDMA. Accordingly we modify the above variables to reflect the existence of slots. Fig. A-1 shows two successive and identical frames. Each frame consists of 3 slots. The length of each slot is  $l$ , and the length of a frame is  $L$ . We use the term frame-average to refer to averages taken over the frame length. For example, the frame-average power of a node is the total energy radiated by that node over the length of the frame, divided by  $L$ , the length of the frame. Similarly, the frame-average bit-rate on  $(i, j)$  is the total number of bits transmitted on  $(i, j)$ , over the length of the frame, divided by  $L$ . Moreover, the term slot-average refers to averages taken over the slot length. The slot-average rate on  $(i, j)$  is similar to the nominal data rate, however we do not think of it as being exactly the same. For example, here we do not preclude transmitting the bits in bursts within a slot, whereas in Chapter 3, we implicitly assume that the bits are transmitted uniformly over a slot. Similarly, slot-average power, frame-average rate, and frame-average power are not the same as (and may even be very different from) nominal transmission power, flow rate, and average transmission power, respectively. We modify the variables such that  $v_{ij}[k]$  is the total slot-average bit-rate on  $(i, j)$  in Slot  $k$ , and  $x_{ij}^s[k]$  is the slot-average bit-rate of Type  $s$  bits on  $(i, j)$  in Slot  $k$ .

### A.1.1 Simple Example

In this subsection, we provide a simple example to illustrate the concepts and definitions discussed so far. Consider the network in Fig. A-2. The network contains three nodes 1, 2, and 3. By assumption, data traffic is in steady-state, and the network



**Figure A-2:** Paths of Session 1 and Session 2.

uses the slotted DS-CDMA channel access scheme.

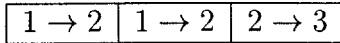
Suppose there are two ongoing end-to-end sessions in the network in Fig. A-2:

1. Session 1:  $O(1) = 1$ ,  $D(1) = 2$ , and  $r^1 = 10$  bits/second.
2. Session 1:  $O(2) = 1$ ,  $D(2) = 3$ , and  $r^2 = 20$  bits/second.

Let the slot length be  $l = 2$  seconds. Recall that  $x_{ij}^s[k]$  is the slot-average rate of Type  $s$  bits on  $(i, j)$  in Slot  $k$ . Suppose we have the following transmissions in the network:

- $x_{12}^1[1] = 30$  bits/second (Session 1 bits from Node 1 to Node 2 in Slot 1.)
- $x_{12}^2[1] = 15$  bits/second (Relay Session 2 traffic through Node 2.)
- $x_{12}^2[2] = 45$  bits/second
- $x_{23}^2[3] = 60$  bits/second (Forward Session 2 traffic to Node 3.)

The frame length is  $L = 3 \times 2 = 6$  seconds. Fig. A-3 shows the active links in each slot. Link  $(i, j)$  will be active if a transmission is occurring on it. Note that the first slot and the second are identical from the perspective of which links are active. Fig. A-2 shows the paths of the two sessions. Session 2 is realized by multi-hop communication. The frame-average bit-rate for Session 1 is the number of bits delivered in one frame divided by the frame length which is equal to  $\frac{30 \times 2}{6} = 10$  bits/second. Similarly the



**Figure A-3:** Frame for the simple network example.

frame-average bit-rate for Session 2 is  $\frac{60 \times 2}{6} = 20$  bits/second. Therefore we clearly satisfy the required session rates  $r^1$  and  $r^2$ .

Finally, note that the first hop in Session 2 is spread over multiple slots (i.e.,  $x_{12}^2[1]$  and  $x_{12}^2[2]$ ). We may want to spread the traffic over multiple slots if the capacity of the channel could not accommodate the required slot-average bit-rate if we were to transmit in one slot.

## A.2 Formulation of The Mathematical Program

In order to formulate the problem of minimum-energy communication over a PRNET as a mathematical program, we need to ask ourselves what information the solution to the mathematical program should contain. First, it should give us the number of slots in the frame. In each slot, it should tell us which nodes are transmitting and which nodes are receiving. Finally, it should tell us the transmission rate along each active link in each slot. The solution must be such that end-to-end average rate requirements are met.

A convenient formulation for the mathematical programming problem is to find the optimal frame-average powers for different frame lengths, and then to find the overall optimal solution among them. Accordingly, the formulation has the following objective function:

$$\underbrace{\min}_{K \in \mathcal{Z}^+} \left\{ \min \frac{1}{K} \sum_{k=1}^K \sum_{i=1}^N G_i[k] \right\} \quad (\text{A.1})$$

minimize over  
positive integers

where,

- $K$  is the number of slots in the frame.

- $G_i[k]$  is the slot-average power radiated by Node  $i$  in Slot  $k$ .  $G_i[k]$  is a function of  $\mathbf{V}[k]$ , where  $\mathbf{V}[k] = [v_{ij}[k]]$ , and  $v_{ij}[k]$  is the total slot-average bit-rate of bits transmitted on  $(i, j)$  in Slot  $k$ .

The minimization would be over  $\mathbf{V}[k]$ . Before we even proceed to list the constraints for this optimization problem, we note that the above objective function does not yield a unique optimal solution. Suppose we find that  $K^*$  is the length of a frame (in slots) that achieves minimum-energy routing. Then we can construct a frame of length  $2K^*$ , that also achieves minimum-energy routing. This is done by duplicating each slot in the frame of length  $K^*$ , to create the new frame of length  $2K^*$ . Furthermore, it is not unlikely that the optimal solution  $K^*$  is a very large number, that is found as  $K \rightarrow \infty$ .

We need to find another formulation to the optimization problem, because the solution to this optimization problem does not provide much insight. The new formulation of the optimization problem requires us to temporarily amend some of the ideas established thus far. First, in the new formulation, we no longer restrict the slots to have a fixed length. On the other hand, a slot (characterized by which nodes are transmitting and which are receiving) may only appear once in a frame. Second, we fix the frame length,  $L$ , to be equal to 1 second, and require it to include as many slots as there are combinations of transmitters and receivers (i.e., enumerate all the possible combinations of transmitters and receivers, and assign a slot for each combination). Consequently the number of variable-length slots in the frame is  $q(N)$ , where

$$q(N) = \sum_{i=1}^{N-1} \binom{N}{i} [2^{(N-i)} - 1]^i$$

as derived in Appendix D (see Eq. D.4). We note the  $q(N)$  grows exponentially with  $N$ . This is precisely why the problem of minimum-energy communication over a PR-NET is complex. We need at least one variable for every combination of transmitters and receivers, and the number of combinations simply increases exponentially with the number of nodes. As a side note, if we consider limiting the number of neighbors each node has to  $M$ , and assuming that the neighbors are already selected, then we



will still have a complex problem. The number of combinations of transmitters is at least on the order of  $q(M + 1)$ . Therefore with  $M = 4$ , we still have at least on the order of one thousand combinations, and with  $M = 5$  at least on the order of ten thousand combinations (see Table D.1 in Appendix D).

The optimal solution consists of a frame with  $q(N)$  slots. Slot  $k$  has length  $l^*[k]$ . If the presence of a particular slot is non-optimal, then its length will be set to zero. Since the length of the frame is fixed to be 1,  $l^*[k]$  represents the fraction of time that slot occupies in the frame.

When we revert to the scheme with equal-length slots, and variable-length frame, we allow multiple occurrences of the same slot in the frame. The number of identical slots divided by the total number of slots in a frame is then equal to the fraction given by the optimal solution to the optimization problem presented here.

Before giving the mathematical formulation for the new optimization problem, we enumerate the different variables. Let

- $K$  be the total number of slots in the frame.  $K$  is equal to the total number of different combinations of transmitters and receivers (i.e.,  $q(N)$ ).
- $l[k]$  ( $1 \leq k \leq K$ ) be the length of Slot  $k$ .  $0 \leq l[k] \leq 1$ , and  $\sum_{k=1}^K l[k] = L = 1$ .
- $x_{ij}^s[k]$  be the slot-average rate (bits/sec) of transmission of bits of Type  $s$  on  $(i, j)$  in Slot  $k$ .
- $\mathbf{V}[k] = [v_{ij}[k]]$ , where  $v_{ij}[k]$  is the slot-average rate (bits/sec) of transmission of bits of all types on  $(i, j)$  in Slot  $k$ .
- $\mathbf{G}[k] = [G_i[k]]$ , where  $G_i[k]$  is the slot-average power radiated by Node  $i$  in Slot  $k$ .
- $G_T[k]$  be the total slot-average power radiated by the network in Slot  $k$ .  $G_T[k] = \sum_{i=1}^N G_i[k]$ .
- $\mathcal{Q}_1(\cdot)$  be a function mapping the matrix  $\mathbf{V}[k]$  to the vector  $\mathbf{G}[k]$ , in such a way that the sum of the powers in the vector  $\mathbf{G}[k]$  is minimized, while maintaining

acceptably small error probabilities in the transmission of  $\mathbf{V}[k]$ .

The problem can now be formulated as:

$$\min \sum_{k=1}^K \sum_{i=1}^N G_i[k]l[k] = \sum_{k=1}^K G_T[k]l[k] \quad (\text{A.2})$$

such that,

$$\sum_{k=1}^K l[k] = 1 \quad (\text{A.3})$$

and such that, for  $i, j = 1, \dots, N$ ;  $k = 1, \dots, K$ ; and  $s = 1, \dots, S$ ,

$$x_{ij}^s[k] \begin{cases} \geq 0 & \text{if transmission on } (i, j) \text{ in Slot } k \text{ is permitted} \\ = 0 & \text{if transmission on } (i, j) \text{ in Slot } k \text{ is not permitted} \end{cases} \quad (\text{A.4})$$

and such that, for  $i, j = 1, \dots, N$  and  $k = 1, \dots, K$

$$v_{ij}[k] = \sum_{s=1}^S x_{ij}^s[k] \quad (\text{A.5})$$

and such that, for  $k = 1, \dots, K$ ,

$$\mathbf{G}[k] = \mathcal{Q}_1(\mathbf{V}[k]) \quad (\text{A.6})$$

$$l[k] \geq 0 \quad (\text{A.7})$$

and such that, for  $i = 1, \dots, N$ ; and  $s = 1, \dots, S$

$$\underbrace{\sum_{k=1}^K \sum_{j=1}^N x_{ij}^s[k]l[k]}_{\text{Flow out}} - \underbrace{\sum_{k=1}^K \sum_{j=1}^N x_{ji}^s[k]l[k]}_{\text{Flow in}} = \begin{cases} r^s & \text{if } i = O(s) \\ -r^s & \text{if } i = D(s) \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.8})$$

### A.2.1 Solving The Mathematical Formulation

The above mathematical formulation is a multi-commodity flow problem. Solving it is a difficult task for a couple of reasons. The first reason is that the number of variables is proportional to the number of different combinations of transmitters and receivers. As previously mentioned this number grows exponentially with the number of nodes in the network, rendering the optimization problem hard to solve by virtue of the number of variables involved.

The second reason is that we have a serious problem with the function  $\mathcal{Q}_1(\mathbf{V}[k])$ . We can assume that  $\mathcal{Q}_1(\mathbf{V}[k])$  finds the information theoretic minimum aggregate power for achieving the set of rates  $(\mathbf{V}[k])$  while maintaining acceptably low error probabilities. However finding this power is an unsolved problem in information theory (The Interference Channel Problem). Alternatively we can assume, as we do in Section 1.3 of Chapter 1, the use of conventional detection strategies under which each interfering user appears as white noise over the DS-CDMA band to the reception of other users. In that case, the power along each link is governed by the  $E_b/I_0$  requirement, and as mentioned in Section 1.3 we make the simplifying assumption that the  $E_b/I_0$  threshold is independent of the slot-average rate on the link. We need to maintain that simplifying assumption for us to be able to solve for the powers. However, this assumption implies that, in the absence of interference from other transmitters, the power for a signal along a link is linear with the slot-average rate on that link. With the assumption of global knowledge and a centralized algorithm, it is highly likely that the optimal solution is one in which interference is avoided. In other words, it tends towards a round-robin scheme in which nodes alternate in blasting out all their data. This is not reasonable as we want the solution to at least take advantage of spatial diversity.

Not knowing how to handle  $\mathcal{Q}_1(\mathbf{V}[k])$  is problematic for us, since if we are to have any hope of solving the optimization problem above (optimally), we will need that function. That being said, we emphasize that the exponential growth in the number of variables (with the number of nodes  $N$ ), is the main lesson we should take from

this exercise.

Finally we note that in this initial approach, we do not make the distinction between flow rate and slot-average rate, like we make the distinction between flow rate and nominal data rate in Chapter 3.<sup>1</sup> The formulation in this appendix assumes that if Node  $i$  is allowed to transmit to Node  $j$  in a particular slot (and has bits to transmit to Node  $j$ ), then it will transmit in that slot in every frame. Without the distinction, it is not apparent to us how to incorporate delay into the problem. We can possibly reformulate the problem to incorporate that distinction, but then we essentially end up with the approach discussed in Chapter 3, albeit with more complexity due to scheduling.

## A.2.2 Convexity of Power in The Interference Channel

Even though there is no known expression for a function  $\mathcal{Q}_1(\mathbf{V}[k])$  that finds the information theoretic power, we show in this section that it is at least convex. We start by giving the definitions for convex sets and convex functions.<sup>2</sup>

**Definition A.1** *A set  $\mathbf{X}$  in  $\mathcal{R}^n$  is said to be convex if for each  $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{X}$ , and for each  $\beta \in [0, 1]$ ,  $\beta\mathbf{x}_1 + (1 - \beta)\mathbf{x}_2 \in \mathbf{X}$ .*

**Definition A.2** *Let  $f : \mathbf{X} \rightarrow \mathcal{R}$ , where  $\mathbf{X}$  is a nonempty convex set in  $\mathcal{R}^n$ . The function  $f$  is said to be convex on  $\mathbf{X}$  if for each  $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{X}$ , and for each  $\beta \in (0, 1)$*   

$$f(\beta\mathbf{x}_1 + (1 - \beta)\mathbf{x}_2) \leq \beta f(\mathbf{x}_1) + (1 - \beta)f(\mathbf{x}_2)$$

Let  $\underline{\phi}$  be a vector in  $\mathcal{R}^M$ , where  $M$  is the number of active links in the network. The  $m^{\text{th}}$  element of  $\underline{\phi}$  is the average bit-rate along the  $m^{\text{th}}$  active link. That is, if we are in Slot  $k$ , and Link  $m$  is active going from Node  $i$  to Node  $j$ , then  $\phi_m = v_{ij}[k]$ . The reason we use  $\phi_m$ 's instead of  $v_{ij}[k]$ 's is to simplify the notation in this subsection. A vector  $\underline{\phi}$  will be achievable, if there exists a transmission strategy such that the rates indicated in the vector can be achieved with arbitrarily small probabilities of error.

---

<sup>1</sup>Frame-average rate is not the same as the flow rate as introduced in Chapter 3.

<sup>2</sup>The definitions are adapted from [BSS79].

**Theorem A.1** Let  $\Phi$ , a set in  $\mathcal{R}^M$ , be the set of achievable network rate vectors.  $\Phi$  is a convex set.

**Proof:** Suppose we have two achievable rate vectors  $\underline{\phi}_1$  and  $\underline{\phi}_2$ . Consider a time interval of length  $T$ . For any  $\beta \in [0, 1]$ , let  $t_1 = \beta T$ . If for the first  $t_1$  seconds the network transmits at network rate vector  $\underline{\phi}_1$  and for the last  $T - t_1$  seconds at rate vector  $\underline{\phi}_2$ ,<sup>3</sup> then the following rate vector  $\frac{t_1 \underline{\phi}_1 + (T - t_1) \underline{\phi}_2}{T} = \beta \underline{\phi}_1 + (1 - \beta) \underline{\phi}_2$  will be achieved. Therefore  $\beta \underline{\phi}_1 + (1 - \beta) \underline{\phi}_2 \in \Phi$ . This implies that for each  $\underline{\phi}_1, \underline{\phi}_2 \in \Phi$ , and for each  $\beta \in [0, 1]$ ,  $\beta \underline{\phi}_1 + (1 - \beta) \underline{\phi}_2 \in \Phi$ . Therefore,  $\Phi$  is a convex set. **Q.E.D.**

What the proof to the above theorem implies is that to achieve certain average bit-rates within a slot, one may need to implement time-sharing, which may lead to dividing the slot into mini-slots, where in each mini-slot a different transmit-receive combination is active.

**Theorem A.2** Let  $\Phi$  be a nonempty convex set in  $\mathcal{R}^M$ . Let  $G_T(\underline{\phi})$  be the minimum network power required to achieve  $\underline{\phi}$ , where  $\underline{\phi} \in \Phi$ .  $G_T(\underline{\phi})$  is convex.

**Proof:** For each pair  $\underline{\phi}_1, \underline{\phi}_2 \in \Phi$  and for each  $\beta \in (0, 1)$ , let  $\underline{\phi}_0 = \beta \underline{\phi}_1 + (1 - \beta) \underline{\phi}_2$ . Now consider a time interval of length  $T$  seconds long. Let  $t_1 = \beta T$ . If we transmit at network rate vector  $\underline{\phi}_1$  for the first  $t_1$  seconds, and at network rate vector  $\underline{\phi}_2$  for the last  $T - t_1$  seconds,<sup>4</sup> then the average network rate vector will be  $\frac{t_1 \underline{\phi}_1 + (T - t_1) \underline{\phi}_2}{T} = \beta \underline{\phi}_1 + (1 - \beta) \underline{\phi}_2 = \underline{\phi}_0$ . Furthermore, the average power over the interval of length  $T$  will be  $\frac{t_1 G_T(\underline{\phi}_1) + (T - t_1) G_T(\underline{\phi}_2)}{T} = \beta G_T(\underline{\phi}_1) + (1 - \beta) G_T(\underline{\phi}_2)$ . This implies that the network can achieve the network rate vector  $\underline{\phi}_0$  with average power of  $\beta G_T(\underline{\phi}_1) + (1 - \beta) G_T(\underline{\phi}_2)$ . By definition  $G_T(\underline{\phi})$  is the minimum average power required to achieve the network rate vector  $\underline{\phi}$ . This implies that  $G_T(\underline{\phi}_0) \leq \beta G_T(\underline{\phi}_1) + (1 - \beta) G_T(\underline{\phi}_2)$ . Therefore,  $G_T(\underline{\phi})$  is convex. **Q.E.D.**

<sup>3</sup>We assume that  $t_1$  and  $T - t_1$  are sufficiently long to achieve those rate vectors.

<sup>4</sup>Again we assume that  $t_1$  and  $T - t_1$  are sufficiently long to achieve those rate vectors.

### A.2.3 Reducing The Number of Variables

The number of slots in the frame is equal to the total number of transmit-receive configurations (i.e.,  $q(N)$ ). The number of variables is proportional to the number of slots. Accordingly, it will be worthwhile if we manage to reduce the number of slots.

Suppose we have a network with three nodes, and consider three slots. In the first Node 1 transmits to Node 3, in the second Node 2 transmits to Node 3, and in the third both Node 1 and Node 2 transmit to Node 3. The third slot is a union of the first two. Therefore, the transmit-receive configuration in the third slot is a “superset” of the transmit-receive configurations in each of the first two slots.<sup>5</sup>

One option for reducing the number of slots is to only use superset slots. Then within each slot, the optimization algorithm is allowed to divide the slot into mini-slots, where each mini-slot is characterized by a subset of the superset. Recall from the previous subsection, that the algorithm minimizing the power may do time-sharing (which may lead to dividing slots into mini-slots) anyway. The number of supersets in a network of  $N$  nodes is

$$p(N) = 2^N - 2$$

as derived in Appendix D (see Eq. D.5). Note that  $p(N)$  still grows exponentially, but not as rapidly as  $q(N)$ . Further we note that when using the  $p(N)$  superset transmit-receive configurations, we are sweeping the complexity arising from the additional variables (i.e., the non-superset transmit-receive combinations) under the rug of computing the minimum aggregate power within the function  $\mathcal{Q}_1(\mathbf{V}[k])$ . The computation implicitly needs to enumerate the different mini-slots.

## A.3 Summary and Conclusion

First we emphasize, that the main goal in the mathematical programming approach was to use the optimal solution to give us insight and intuition into what constitutes good minimum-energy communication over a PRNET. It was established, a priori,

---

<sup>5</sup>See Appendix D for more information about superset slots.

that the mathematical programming approach could not be used in real life networks, since it assumes global knowledge of the network, and centralized control. Furthermore, the formulation assumes a snapshot (in time) of the network, which implies that the nodes are treated as stationary nodes, the radio channel as being time-invariant, and the required end-to-end rates are treated as steady state rates.

Even with our modest aspirations for the mathematical programming approach, it turned out that this approach is a blind alley. This is because even in an off-line setting, the mathematical programming problem is very hard to solve. One of the main problems is the exponential growth in the number of variables (with the number of nodes). The other main problem is that we have no known way to intelligently express aggregate power as a function of the slot-average rates. Finally, it is not clear how one might, within such a framework, consider delay, which was neglected in this formulation, despite its relevance.

However, all is not lost, as we gained insight into how not to approach the problem. First we need to ask ourselves why this mathematical program is difficult even though on the surface it appears to be just another network optimization problem. The fundamental difference here is the need to account for scheduling. The exponential growth in the number of variables arises due to the need to enumerate all possible scheduling combinations in a PRNET.<sup>6</sup> This suggests that we try a sub-optimal approach in which routing and scheduling are solved separately. The challenge in the sub-optimal approach is finding the right way to separate the two. We want to separate them in such a way that they can be solved independently, and yet have the connection between them maintain the spirit of the original problem.<sup>7</sup>

---

<sup>6</sup>Even if we enumerate the slots containing the supersets of transmit-receive configurations, we still end up with an exponentially growing number of variables.

<sup>7</sup>There is a third component, namely the topological organization problem, which is a non-problem in this approach because it is assumed that all nodes are neighbors of each other.

# Appendix B

## Medium Access Schemes for PRNETs with Emphasis on DS-CDMA

In this appendix we discuss the multiple access techniques we considered in Chapter 1 as candidates for the PRNET medium access scheme. They are time division multiple access, frequency division multiple access, frequency hopping, and direct sequence code division multiple access. We focus more on DS-CDMA, because the medium access scheme in our PRNET model is slotted DS-CDMA.

### B.1 Frequency Division Multiple Access

In Frequency Division Multiple Access (FDMA), the channel bandwidth is partitioned into separate sub-bands. Signals transmitted in different sub-bands are orthogonal to each other. In the case of a single receiver, different transmitters need to be allocated different sub-bands. We assume that the receiver is continuously tuned in to all the sub-bands.<sup>1</sup> In the multiple-receiver case, which arises in PRNETs,<sup>2</sup> one

---

<sup>1</sup>This assumption is not necessarily true. Without that assumption FDMA is complicated by the need to have the receiver know when to tune in to a sub-band to receive a transmission in it.

<sup>2</sup>In this discussion we make the simplifying assumption that there is a fixed set of receivers.



needs to take advantage of spatial diversity in order to use the channel efficiently. If transmitters are sufficiently distant from each other, then their signals will interfere with each other negligibly, and they can be assigned the same sub-band. Therefore, in the multiple-receiver case, transmitters need to be partitioned into subsets, where transmitters in the same subset are selected such that their signals negligibly interfere with each other, allowing them to use the same sub-band.

## **B.2 Time Division Multiple Access**

In Time Division Multiple Access (TDMA) the division is over time rather than frequency. Time is divided into non-overlapping intervals. The intervals are typically of equal lengths, and referred to as slots. Signals transmitted in different slots are orthogonal to each other. Slots are typically grouped into identical and periodically recurring frames. A slot in a frame is analogous to a sub-band in FDMA.

## **B.3 Frequency Hopping**

Frequency hopping is a combination of TDMA and FDMA [Ste95a]. In frequency hopping, the bandwidth is again divided into many sub-bands. In addition time is divided into slots. In each slot, the transmitting node hops to a new frequency-band. The hopping pattern is dictated by a pseudo-random sequence. Pseudo-random sequences for different users can be designed such that the probability that two users collide (i.e., use the same frequency sub-band in the same time slot) is small. If a collision occurs in a sub-band, that part of the signal will be lost, but can be recovered through error-correction codes and interleaving. Hopping patterns are designed such that signals may occasionally collide in some sub-bands, but do not totally destroy each other.

## B.4 Direct Sequence Code Division Multiple Access

Direct Sequence Code Division Multiple Access (DS-CDMA) has emerged as a viable and favorable alternative to FDMA and TDMA in cellular networks. Before DS-CDMA became a popular multiple access technique in cellular networks, its primary use was in military communications. This is because DS-CDMA has anti-jamming, anti-listening-in, and anti-intercept capabilities [Ste95a] (pp. 45-51), [Pro95] (pp. 695-697), [PuR93]. Moreover, DS-CDMA exhibits anti-multipath-fading capability, which can be realized using a RAKE receiver [Gal94], [Pro95]. The anti-multipath-fading capability is particularly useful in wireless communications.

In DS-CDMA, each transmitted signal occupies the entire channel bandwidth simultaneously with all other transmitted signals. Typically the channel bandwidth is much larger than the minimum required bandwidth needed to transmit the signal, assuming BPSK or QPSK modulation with a given bit-rate [Vit95] (pp. 1-8), [Sou90], [Pur87], [Ste95a] (pp. 45-51). The narrow-band signal (i.e., the signal that would be used if BPSK or QPSK modulation were employed) is spread over the bandwidth via a signature/spreading code sequence unique to the signal. A receiver can de-spread and separate the signals by cross-correlating the total received signal with each spreading code.

In DS-CDMA, it is possible for a transmitter to transmit different signals to several receivers simultaneously. We refer to this case as the one-to-many case. It is also possible for a receiver to receive from multiple transmitters simultaneously. This is the many-to-one case. In the one-to-many case, we assume that the signals emanating from the single transmitter do not interfere with each other. This assumption is based on the observation that the spreading codes for different transmitted signals can be chosen such that the spread signals are orthogonal to each other. Assuming single-path propagation for each signal, the transmitted signals remain orthogonal everywhere, implying they do not interfere with each other at any receiver. With multipath propagation, the aggregate signal arriving at a receiver is distorted in a

manner similar to passing the signal through a linear filter [Gal94]. As a result of the distortion, individual signals are not completely orthogonal, and do interfere with each other to some extent at the receivers. Nevertheless, we assume that we can ignore the interference among signals that are originally orthogonal at the transmitter.

In the many-to-one case, it is very difficult to make the received signals orthogonal, even if we ignore multipath propagation. An essential component in having orthogonal DS-CDMA signals is the ability to maintain time synchronization of the spreading codes, among the spread signals at the receiver. In the case of many-to-one, this synchronization is very difficult. Even if the clocks of the transmitters are perfectly synchronized, each transmitted signal will take a different path to the receiver. Since each path takes a different amount of time to traverse, the signals almost surely arrive at the receiver unsynchronized, unless the transmitters engage in the difficult task of offsetting their transmission times such that the signals arrive at the receiver synchronized. Due to the lack of synchronization, interference among signals is inevitable. Therefore, in the many-to-one case the question becomes of how to shape the inevitable interference, so that it can be dealt with. The spreading code sequences are designed to have relatively small cross-correlations, regardless of the offsets between them. In other words, signals are designed such that they partially interfere with each other without completely destroying each other. The interference caused by one signal to another can be approximated as additive white Gaussian noise (AWGN), with spectral density equal to the power of the interfering signal divided by the channel bandwidth [Sou90], [Vit95] (pp. 4-6). The wider the bandwidth over which the narrow-band signal is spread, the better the AWGN approximation of the interference [Sou90].

#### **B.4.1 Power Control in DS-CDMA**

The value of  $E_b/I_0$  for a received signal determines the bit-error rate [Sou90], [Vit95] (pp. 4-6). Typically one wants the value of  $E_b/I_0$  to be above a certain threshold, in order to achieve a desired bit-error rate. Suppose Node  $i$  is transmitting to Node  $k$ ,

then we want

$$\left(\frac{E_b}{I_0}\right)_{ik} \geq \gamma_{ik} \quad (\text{C-B.1})$$

$$\Rightarrow \frac{\alpha_{ik}P_{ik}/c_{ik}}{I_0} \geq \gamma_{ik} \quad (\text{C-B.2})$$

where  $P_{ik}$  is the power transmitted by Node  $i$  to Node  $k$ ,  $\gamma_{ik}$  is the required  $E_b/I_0$  threshold,  $c_{ik}$  is the transmission rate in bits per second, and  $\alpha_{ik}$  is the path propagation loss, such that the signal arrives at the receiver with power  $\alpha_{ik}P_{ik}$ .  $I_0$  is the density of the interference to Node  $i$ 's signal.

$$I_0 = N_0 + \frac{\sum_{j \in \mathcal{T}, j \neq i} \alpha_{jk}P_j}{W}$$

where  $\mathcal{T}$  is the set of nodes transmitting concurrently with Node  $i$ , and  $P_j$  is the total power radiated by Node  $j$ . Therefore Condition C-B.2 can be expressed as

$$\frac{\alpha_{ik}P_{ik}/c_{ik}}{N_0 + \frac{\sum_{j \in \mathcal{T}, j \neq i} \alpha_{jk}P_j}{W}} \geq \gamma_{ik} \quad (\text{C-B.3})$$

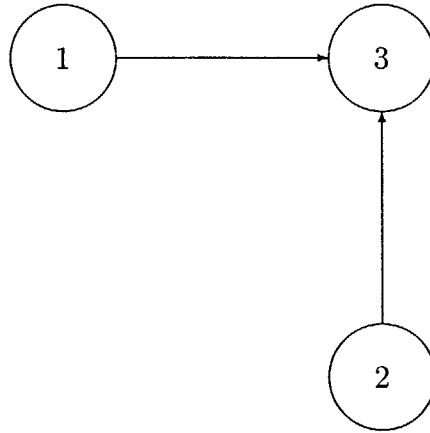
Each transmitted signal needs to satisfy a condition corresponding to Condition C-B.3.

The conditions can be summarized using matrix notation as we demonstrate using an example. We revisit the simple network in Example 3.3, shown again in Fig. B-1. The conditions corresponding to Condition C-B.3 are

$$\frac{\alpha_{13}P_{13}/c_{13}}{N_0 + \frac{\alpha_{23}P_{23}}{W}} \geq \gamma_{13} \quad (\text{C-B.4})$$

and

$$\frac{\alpha_{23}P_{23}/c_{23}}{N_0 + \frac{\alpha_{13}P_{13}}{W}} \geq \gamma_{23} \quad (\text{C-B.5})$$



**Figure B-1:** A simple network.

Let

$$\mathbf{p} = \begin{bmatrix} P_{13} \\ P_{23} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & \frac{\alpha_{23} \gamma_{13} c_{13}}{W \alpha_{13}} \\ \frac{\alpha_{13} \gamma_{23} c_{23}}{W \alpha_{23}} & 0 \end{bmatrix}$$

and

$$\nu = \begin{bmatrix} \frac{\gamma_{13} c_{13}}{\alpha_{13}} N_0 \\ \frac{\gamma_{23} c_{23}}{\alpha_{23}} N_0 \end{bmatrix}$$

then the Conditions C-B.4 and C-B.5 can be summarized by the following matrix inequality:

$$\mathbf{p} \geq \mathbf{I}(\mathbf{p})$$

where

$$\mathbf{I}(\mathbf{p}) = \mathbf{B}\mathbf{p} + \nu$$

We note that for  $\mathbf{p} \geq 0$ ,  $\mathbf{I}(\mathbf{p})$  satisfies the following three properties:

1. Positivity -  $\mathbf{I}(\mathbf{p}) > 0$ : If  $\mathbf{p} = 0$ , then  $\mathbf{B}\mathbf{p} = 0$ , but  $\nu > 0$  so  $\mathbf{I}(\mathbf{p}) > 0$ . If  $\mathbf{p} > 0$ , then since each element in  $\mathbf{B}$  is nonnegative,  $\mathbf{B}\mathbf{p} > 0$ .

2. Monotonicity - If  $\mathbf{p} \geq \mathbf{p}'$ , then  $\mathbf{I}(\mathbf{p}) \geq \mathbf{I}(\mathbf{p}')$ :  $\mathbf{I}(\mathbf{p}) - \mathbf{I}(\mathbf{p}') = \mathbf{B}(\mathbf{p} - \mathbf{p}') \geq 0$ .
3. Scalability - For all  $\mu > 1$ ,  $\mu\mathbf{I}(\mathbf{p}) > \mathbf{I}(\mu\mathbf{p})$ :  $\mu\mathbf{I}(\mathbf{p}) - \mathbf{I}(\mu\mathbf{p}) = (\mu - 1)\nu > 0$ .

We note that the above properties of  $\mathbf{I}(\mathbf{p})$  generalize to any PRNET. Using the terminology from [Yat95], by satisfying the three properties above,  $\mathbf{I}(\mathbf{p})$  is referred to as a *standard interference function*. [Yat95] develops several results for standard interference functions. Those results allow us to make the following statements:

1. To minimize aggregate transmission power in the network, the transmission powers must satisfy Condition C-B.3 with equality. Intuitively, it is in the interest of a transmitting Node  $i$  to lower its own transmission power,  $P_{ik}$ , to the level where Condition C-B.3 is met with equality. Upon doing so however, Node  $i$  lowers its interference to other transmitters, allowing them to reduce their transmission powers. If the other transmitters lower their powers, then they will interfere less with Node  $i$ 's signal, allowing it to lower its power even further, and so on. Accordingly, the transmitting nodes can collectively lower their transmission powers until they all meet their required  $E_b/I_0$  thresholds with equality.
2. Starting with any initial power vector  $\mathbf{p}(0) \geq \mathbf{I}(\mathbf{p}(0))$ , if we apply the iterative power control algorithm

$$\mathbf{p}(t+1) = \mathbf{I}(\mathbf{p}(t))$$

then  $\mathbf{p}(t)$  will converge to the unique fixed point,  $\mathbf{p}^*$  (where  $\mathbf{p}^* = \mathbf{I}(\mathbf{p}^*)$ ), provided it exists.

So far we have implicitly assumed that the  $\alpha_{ik}$ 's are constant. However, as the wireless channel varies with time, the path propagation loss factors also vary with time. Accordingly, the transmitting nodes need to regulate their transmission powers to track the changes in the  $\alpha_{ik}$ 's, such that the  $E_b/I_0$  level of a signal at its intended receiver remains constant. The process by which transmitters adjust their radiated powers is referred to as power control. In Qualcomm DS-SS-CDMA systems, power control is implemented via a closed control loop, whereby the receiving node continuously

instructs the transmitting node to incrementally decrease or increase its transmission power.

### **B.4.2 Similarities Between DS-CDMA and Frequency Hopping**

Frequency hopping and DS-CDMA can be seen as equivalent in several ways. In both systems, signals are designed to partially interfere with each other without completely destroying each other. In frequency hopping, each signal is spread over the entire channel bandwidth (as it hops from one frequency sub-band to another). The hopping pattern of each transmitter is dictated by a pseudo-random sequence, which is analogous to the spreading code in DS-CDMA. Frequency hopping also has an anti-fading capability that is achieved through frequency diversity i.e. transmitting the signal over different frequency sub-bands [Pro95] (pp. 777-778). Finally the pseudo-random hopping patterns give frequency hopping anti-jamming, anti-intercept, and anti-listening-in capabilities.

## **B.5 Random Access Protocols**

In the three multiple access schemes mentioned above, coordination among nodes is required to avoid conflicts. In the case of TDMA, two nodes transmitting to the same receiver need to use different time slots. Random access protocols, such as ALOHA, have been frequently talked about in the literature (e.g., [SiK83], [Tob87], and [KIT75]) as a way to avoid the complexity of coordination among nodes when it comes to sharing the channel. In random access protocols, there is little or no coordination among the nodes, which comes at the price of occasional conflicts in sharing the channel. However, we do not think random access protocols are suitable for our design. Our objection to ALOHA [BeG92], for example, is that transmission energy is wasted in collisions. Moreover, even in the simple case of a single receiver, it has a relatively low channel throughput (18% for unslotted and 37% for slotted),

in terms of the percentage of time the channel is used for successful transmissions (i.e., not idle, and no collisions). We suspect ALOHA's inefficient use of the channel carries over to the multiple-receiver situation in PRNETs.

Another random access protocol is Carrier Sense Multiple Access (CSMA), which is really an enhancement of ALOHA, based on additional technological capability [BeG92]. In the single-receiver case, CSMA outperforms ALOHA in terms of throughput. However, it is not clear how that advantage is translated in PRNETs. One reason is the hidden terminal problem described in [LNT87].

Recently, two reservation-based protocols, motivated by the goal of overcoming the hidden terminal problem in CSMA, have been introduced in the literature. The two protocols are MACA (Multi-Access Collision Avoidance) [Kar90] and MACAW<sup>3</sup> [Bh+94]. One objection to MACA and MACAW is that, for correct operation, it is assumed that if Node 1 does not hear Node 2's transmission, then Node 2 cannot hear Node 1's transmission. This will not necessarily be true if nodes adjust their transmission powers based on the intended receivers.

---

<sup>3</sup>MACAW is essentially an enhancement of MACA.



# Appendix C

## Non-Convexity Results

In this appendix we use a counterexample to show that the aggregate nominal transmission power, as introduced in Section 1.3 (Eq. 1.1), is not a convex function of the nominal data rates. We also use a counterexample to demonstrate that the aggregate average transmission power, as introduced in Section 3.2 (Eq. 3.11), is also not a convex function of the nominal data rates (although it is a convex function of the flow rates).

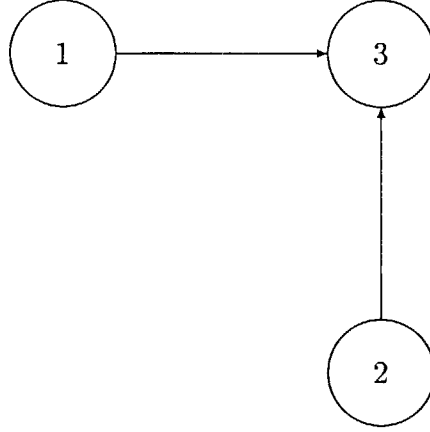
### C.1 Non-Convexity of Aggregate Nominal Transmission Power

Consider the simple network in Fig. C-1, where Node 1 and Node 2 are transmitting to Node 3. Let  $\gamma$  be the required  $E_b/I_0$  threshold for both signals. Then the following equations minimize the power assignment subject to meeting the  $E_b/I_0$  threshold,

$$\frac{\alpha_{13}P_{13}/c_{13}}{N_0 + \alpha_{23}P_{23}/W} = \gamma$$

$$\frac{\alpha_{23}P_{23}/c_{23}}{N_0 + \alpha_{13}P_{13}/W} = \gamma$$

where  $N_0$  is the background noise density,  $W$  is the DS-CDMA bandwidth,  $P_{ij}$  is the nominal transmission power radiated by Node  $i$  due to the signal from Node  $i$  to Node



**Figure C-1:** Transmission with two transmitters and one receiver.

$j$ , and  $c_{ij}$  is the nominal data rate of that signal. Finally,  $\alpha_{ij}$  is the path propagation loss (i.e., the power received at Node  $j$  is  $\alpha_{ij}P_{ij}$ ). Solving for the powers, we get

$$P_{13} = \frac{Wc_{13}\gamma N_0(W + c_{23}\gamma)}{\alpha_{13}(W^2 - \gamma^2c_{13}c_{23})}$$

$$P_{23} = \frac{Wc_{23}\gamma N_0(W + c_{13}\gamma)}{\alpha_{23}(W^2 - \gamma^2c_{13}c_{23})}$$

Accordingly, the aggregate nominal transmission power,  $P_T$ , is given by

$$\begin{aligned}
P_T(c_{13}, c_{23}) &= P_{13}(c_{13}, c_{23}) + P_{23}(c_{13}, c_{23}) \\
&= \frac{Wc_{13}\gamma N_0(W + c_{23}\gamma)}{\alpha_{13}(W^2 - \gamma^2c_{13}c_{23})} + \frac{Wc_{23}\gamma N_0(W + c_{13}\gamma)}{\alpha_{23}(W^2 - \gamma^2c_{13}c_{23})} \\
&= \left( \frac{1}{1 - \frac{\gamma^2}{W^2}c_{13}c_{23}} \right) \left( \frac{c_{13}\gamma N_0(W + c_{23}\gamma)}{\alpha_{13}W} + \frac{c_{23}\gamma N_0(W + c_{13}\gamma)}{\alpha_{23}W} \right) \\
&= \frac{\frac{\gamma N_0}{\alpha_{13}}c_{13} + \frac{\gamma N_0}{\alpha_{23}}c_{23} + \left( \frac{\gamma^2 N_0}{\alpha_{13}W} + \frac{\gamma^2 N_0}{\alpha_{23}W} \right) c_{13}c_{23}}{1 - \frac{\gamma^2}{W^2}c_{13}c_{23}} \\
&= \frac{k_1c_{13} + k_2c_{23} + k_3c_{13}c_{23}}{1 - k_4c_{13}c_{23}}
\end{aligned}$$

where  $k_1 = \frac{\gamma N_0}{\alpha_{13}}$ ,  $k_2 = \frac{\gamma N_0}{\alpha_{23}}$ ,  $k_3 = \frac{\gamma^2 N_0}{\alpha_{13} W} + \frac{\gamma^2 N_0}{\alpha_{23} W}$ , and  $k_4 = \frac{\gamma^2}{W^2}$ . Note that  $\gamma$ ,  $N_0$ ,  $W$ ,  $\alpha_{13}$ , and  $\alpha_{23}$  are all positive, which implies that  $k_1$ ,  $k_2$ ,  $k_3$ , and  $k_4$  are all positive. Furthermore we know that  $1 - k_4 c_{13} c_{23} > 0$ , otherwise  $P_T$  is negative or infinite.<sup>1</sup>

For  $P_T(c_{13}, c_{23})$  to be convex, the Hessian of  $P_T(c_{13}, c_{23})$  must be positive semi-definite. Let  $H_P$  be the Hessian of  $P_T(c_{13}, c_{23})$ .

$$H_P = \begin{bmatrix} \frac{-2k_4 c_{23}(k_1 + k_3 c_{23} + k_2 k_4 c_{23}^2)}{(-1 + k_4 c_{13} c_{23})^3} & \frac{-(k_3 + k_3 k_4 c_{13} c_{23} + 2k_2 k_4 c_{23} + 2k_1 k_4 c_{13})}{(-1 + k_4 c_{13} c_{23})^3} \\ \frac{-(k_3 + k_3 k_4 c_{13} c_{23} + 2k_2 k_4 c_{23} + 2k_1 k_4 c_{13})}{(-1 + k_4 c_{13} c_{23})^3} & \frac{-2k_4 c_{13}(k_2 + k_3 c_{13} + k_1 k_4 c_{13}^2)}{(-1 + k_4 c_{13} c_{23})^3} \end{bmatrix}$$

For  $H_P$  to be positive semi-definite, all of the following quantities must be non-positive:

Q1:

$$\frac{2k_4 c_{23}(k_1 + k_3 c_{23} + k_2 k_4 c_{23}^2)}{(1 - k_4 c_{13} c_{23})^3}$$

Q2:

$$\frac{4k_1 k_2 k_4^3 c_{13}^2 c_{23}^2 + 4k_1 k_3 k_4^2 c_{13}^2 c_{23} + 4k_1^2 k_4^2 c_{13}^2 + 4k_2 k_3 k_4^2 c_{13} c_{23}^2 + 4k_1 k_2 k_4^2 c_{13} c_{23}}{(1 - k_4 c_{13} c_{23})^5} + \frac{4k_2^2 k_4^2 c_{23}^2 + 3k_3^2 k_4 c_{13} c_{23} + 4k_1 k_3 k_4 c_{13} + 4k_2 k_3 k_4 c_{23} + k_3^2}{(1 - k_4 c_{13} c_{23})^5}$$

Q3:

$$\frac{2k_4 c_{13}(k_2 + k_3 c_{13} + k_1 k_4 c_{13}^2)}{(1 - k_4 c_{13} c_{23})^3}$$

Examining Q2 closely, we see that its denominator is positive (since  $1 - k_4 c_{13} c_{23} > 0$ ). The numerator in Q2 is also positive. Therefore, Q2 is positive. This implies  $H_P$  is not positive semi-definite, which in turn implies that  $P_T(c_{13}, c_{23})$  is not convex.

---

<sup>1</sup>If  $P_T$  is negative or infinite, then this will correspond to the infeasible case where the bandwidth cannot accommodate the required transmission rates and/or the required  $E_b/I_0$  threshold. We assume that we are not operating in such a regime.

## C.2 Non-Convexity of Aggregate Average Transmission Power

The simple network in Fig. C-1 also leads to a counterexample that demonstrates that the aggregate average transmission power is not a convex function of the nominal data rates. Let the flow rate on (1, 3) and (2, 3) be  $f_{13}$  and  $f_{23}$  respectively. The aggregate average transmission power, as a function of  $c_{13}$  and  $c_{23}$ , is

$$\begin{aligned}
\tilde{P}_T(c_{13}, c_{23}) &= \tilde{P}_{13}(c_{13}, c_{23}) + \tilde{P}_{23}(c_{13}, c_{23}) \\
&= \frac{W f_{13} \gamma N_0 (W + c_{23} \gamma)}{\alpha_{13} (W^2 - \gamma^2 c_{13} c_{23})} + \frac{W f_{23} \gamma N_0 (W + c_{13} \gamma)}{\alpha_{23} (W^2 - \gamma^2 c_{13} c_{23})} \\
&= \frac{\frac{\gamma N_0}{\alpha_{13}} f_{13} + \frac{\gamma N_0}{\alpha_{23}} f_{23} + \frac{\gamma^2 N_0}{\alpha_{13} W} f_{13} c_{23} + \frac{\gamma^2 N_0}{\alpha_{23} W} f_{23} c_{13}}{1 - \frac{\gamma^2}{W^2} c_{13} c_{23}} \\
&= \frac{k_1 + k_2 c_{13} + k_3 c_{23}}{1 - k_4 c_{13} c_{23}}
\end{aligned}$$

where  $k_1 = \frac{\gamma N_0}{\alpha_{13}} f_{13} + \frac{\gamma N_0}{\alpha_{23}} f_{23}$ ,  $k_2 = \frac{\gamma^2 N_0}{\alpha_{23} W} f_{23}$ ,  $k_3 = \frac{\gamma^2 N_0}{\alpha_{13} W} f_{13}$ , and  $k_4 = \frac{\gamma^2}{W^2}$ .

For  $\tilde{P}_T(c_{13}, c_{23})$  to be convex, the following quantities must be non-positive:

Q1:

$$-2 \frac{k_4 c_{23} (k_3 + k_1 k_4 c_{23} + k_2 k_4 c_{23}^2)}{(1 - k_4 c_{13} c_{23})^3}$$

Q2:

$$\begin{aligned}
&\frac{k_4^2 (4k_2 k_3 k_4 c_{13}^2 c_{23}^2 + 4k_1 k_2 k_4 c_{13} c_{23}^2 + 4k_1 k_3 k_4 c_{13}^2 c_{23} + 3k_1^2 k_4 c_{13} c_{23} + 4k_2^2 c_{23}^2)}{(1 - k_4 c_{13} c_{23})^5} + \\
&\frac{k_4^2 (4k_2 k_3 c_{13} c_{23} + 4k_1 k_2 c_{23} + 4k_3^2 c_{13}^2 + 4k_1 k_3 c_{13} + k_1^2)}{(1 - k_4 c_{13} c_{23})^5}
\end{aligned}$$

Q3:

$$-2 \frac{k_4 c_{13} (k_2 + k_1 k_4 c_{13} + k_3 k_4 c_{13}^2)}{(1 - k_4 c_{13} c_{23})^3}$$

Since  $\gamma$ ,  $N_0$ ,  $W$ ,  $\alpha_{13}$ ,  $\alpha_{23}$ , and  $1 - k_4 c_{13} c_{23}$  are positive, Q2 will be zero (i.e., satisfy the

condition of non-positivity) only if  $f_{13} = f_{23} = 0$ . However, this corresponds to the uninteresting case of having no flows, and hence no transmissions. In this case, the average power is zero for all  $c_{13}$  and  $c_{23}$ . If either  $f_{13}$  or  $f_{23}$  is positive, then  $Q_2$  will be positive. Therefore except for the uninteresting case of  $f_{13} = f_{23} = 0$ ,  $\tilde{P}_T(c_{13}, c_{23})$  is not a convex function.

# Appendix D

## Transmission Matrices

A transmission matrix is a useful way of summarizing a transmit-receive configuration in a PRNET. In this appendix we introduce transmission matrices and analyze their properties. We also show how the minimum-frame-length scheduling problem discussed in Chapter 2 can be expressed in terms of transmission matrices.

Let each time slot have an associated transmission matrix. We denote the transmission matrix for Slot  $k$  by the 0-1 matrix  $\mathbf{A}[k] = [a_{ij}[k]]$ , where  $a_{ij}[k]$  is 1 if Node  $i$  is transmitting to Node  $j$  in Slot  $k$ , and 0 otherwise.

Suppose, for example, we have a network of three nodes (1, 2, and 3), where in the first slot Node 1 transmits to both Node 2 and Node 3, and in the second slot 2 and 3 both transmit to 1. Then,

$$\mathbf{A}[1] = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and

$$\mathbf{A}[2] = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Not every 0-1 matrix is an admissible transmission matrix. For example the following matrix

$$\mathbf{A}[k] = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

is not admissible because  $a_{11}[k] = 1$  implying Node 1 is transmitting to itself, which is nonsensical. Furthermore, according to this matrix, Node 2 is receiving and transmitting simultaneously, which by assumption is not possible. The following is a formal definition of an admissible transmission matrix:

**Definition D.1** Let  $\mathbf{A}[k]$  be an  $N \times N$  0-1 matrix.  $\mathbf{A}[k]$  is an admissible transmission matrix if and only if in the transmission pattern described by  $\mathbf{A}[k]$ ,

1. There is at least one transmission. Equivalently,  $\mathbf{A}[k] \neq \mathbf{0}$ .
2. No node is both transmitting and receiving (i.e., if  $a_{ij} > 0$  for any  $j$ , then  $a_{ki} = 0$  for all  $k$ ).
3. No node is transmitting to itself. Equivalently,  $a_{ii} = 0$  for  $1 \leq i \leq N$ .

The theorem below provides an easy test for the admissibility of a transmission matrix:

**Theorem D.1** Let  $\mathbf{A}[k]$  be an  $N \times N$  0-1 matrix.  $\mathbf{A}[k]$  is an admissible transmission matrix if and only if  $\mathbf{A}[k]\mathbf{A}[k] = \mathbf{0}$  and  $\mathbf{A}[k] \neq \mathbf{0}$ .

**Proof:** If  $\mathbf{A}[k]$  is an admissible transmission matrix, then  $\mathbf{A}[k] \neq \mathbf{0}$ . Further, let  $\mathbf{v}_r[k]$  be a row vector whose  $i^{\text{th}}$  element is the sum of the entries in the  $i^{\text{th}}$  column of  $\mathbf{A}[k]$ . If the  $i^{\text{th}}$  element of  $\mathbf{v}_r[k]$  is nonzero, then Node  $i$  will be receiving in Slot  $k$ . Similarly, let  $\mathbf{v}_c[k]$  be a column vector, whose  $i^{\text{th}}$  element is the sum of the entries in the  $i^{\text{th}}$  row of  $\mathbf{A}[k]$ . If the  $i^{\text{th}}$  element of  $\mathbf{v}_c[k]$  is nonzero, then Node  $i$  will be transmitting in Slot  $k$ . If  $\mathbf{A}[k]$  is an admissible transmission matrix, then the  $i^{\text{th}}$

element of  $\mathbf{v}_c[k]$  and the  $i^{\text{th}}$  element of  $\mathbf{v}_r[k]$  cannot both be nonzero. This implies that  $\mathbf{v}_r[k]\mathbf{v}_c[k] = 0$ . This can be rewritten as:

$$(\mathbf{u}^T \mathbf{A}[k])(\mathbf{A}[k]\mathbf{u}) = 0 \quad (\text{D.1})$$

where  $\mathbf{u} = [1 \ 1 \ \dots \ 1 \ 1]^T$ .

$$\Rightarrow \mathbf{u}^T(\mathbf{A}[k]\mathbf{A}[k])\mathbf{u} = 0 \quad (\text{D.2})$$

$$\Rightarrow \mathbf{A}[k]\mathbf{A}[k] = \mathbf{0} \quad (\text{D.3})$$

which follows since  $\mathbf{A}[k]$  is non-negative. Therefore, if  $\mathbf{A}[k]$  is an admissible transmission matrix, then  $\mathbf{A}[k]\mathbf{A}[k] = \mathbf{0}$ , and  $\mathbf{A}[k] \neq \mathbf{0}$ .

Now, if  $\mathbf{A}[k] \neq \mathbf{0}$  then the first condition in Definition D.1 will be directly satisfied. Further, if  $\mathbf{A}[k]\mathbf{A}[k] = \mathbf{0}$  then every row in  $\mathbf{A}[k]$  will be orthogonal to every column. In particular the product of the  $i^{\text{th}}$  row and the  $i^{\text{th}}$  column is 0, for  $1 \leq i \leq N$ . This implies that for  $1 \leq i \leq N$ ,

$$\begin{aligned} a_{ii}^2 + \text{other nonnegative terms} &= 0 \\ \Rightarrow a_{ii} &= 0 \end{aligned}$$

This satisfies the third condition in Definition D.1. Finally, if  $\mathbf{A}[k]\mathbf{A}[k] = \mathbf{0}$  then  $\mathbf{v}_r[k]\mathbf{v}_c[k] = 0$ . This in turn implies that no node is both transmitting and receiving in Slot  $k$ , which satisfies the second condition in Definition D.1. **Q.E.D.**

In a network with three nodes, Node 1, Node 2, and Node 3, there are 12 admissible transmission matrices. They are:



$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\{1 \rightarrow 2\}$        $\{1 \rightarrow 3\}$        $\{2 \rightarrow 1\}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$\{2 \rightarrow 3\}$        $\{3 \rightarrow 1\}$        $\{3 \rightarrow 2\}$

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$\{1 \rightarrow 2\}$        $\{2 \rightarrow 1\}$        $\{3 \rightarrow 1\}$   
 $\{1 \rightarrow 3\}$        $\{2 \rightarrow 3\}$        $\{3 \rightarrow 2\}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$\{2 \rightarrow 1\}$        $\{1 \rightarrow 2\}$        $\{1 \rightarrow 3\}$   
 $\{3 \rightarrow 1\}$        $\{3 \rightarrow 2\}$        $\{2 \rightarrow 3\}$

Note that in the first of the 12 matrices above, namely

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Node 3 is neither transmitting nor receiving. We may want to focus on the subset of admissible transmission matrices, in which each node is either transmitting or receiving; and if a node is transmitting, it will transmit to all the receivers, and

vice versa. We refer to that subset as the subset of *superset* admissible transmission matrices. In the example above, one such admissible transmission matrix is

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

It is a superset (i.e., contains) the two matrices

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

and

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

In the example of the three-node network, the subset of superset matrices consists of the matrices containing two 1's (i.e., the last six of the listed matrices).

For a general network of  $N$  nodes, it is of interest to know the number of distinct admissible transmission matrices and superset admissible transmission matrices, as they determine the sizes of certain PRNET optimization problems (see Appendix A). Let  $q(N)$  be the number of admissible transmission matrices, then

$$q(N) = \sum_{i=1}^{N-1} \binom{N}{i} [2^{(N-i)} - 1]^i \quad (\text{D.4})$$

To see why  $q(N)$  is as given in Eq. D.4, suppose there are  $i$  nodes actively receiving. Then we have at least 1 and at most  $(N - i)$  transmitters among the remaining nodes (some nodes may be idle). For each of the receivers, there are  $(2^{(N-i)} - 1)$  possible combinations of transmitters sending to that receiver. In other words, for each receiver there are  $(2^{(N-i)} - 1)$  possible ways of being a receiver, which implies

$N$	$q(N)$
2	2
3	12
4	86
5	840
6	11642
7	227892
$\vdots$	$\vdots$

**Table D.1:** Growth of the number of admissible matrices with  $N$ .

that for the  $i$  receivers there are  $(2^{(N-i)} - 1)^i$  ways. There are  $\binom{N}{i}$  ways of choosing  $i$  receivers among  $N$  nodes.

Table D.1 shows how the number of admissible matrices grows with  $N$ , the number of nodes in the network. Note that the number grows exponentially.

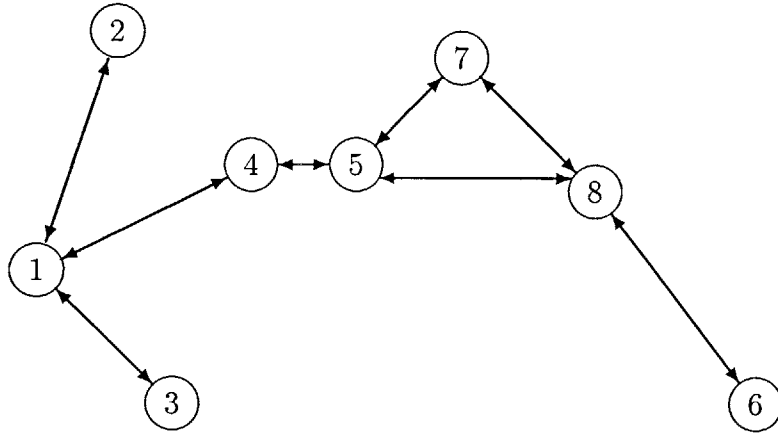
Similarly, let  $p(N)$  be the number of superset transmission matrices.

$$p(N) = 2^N - 2 \tag{D.5}$$

Eq. D.5 simply enumerates all the  $N$ -digit 0-1 strings, minus the all 0 and all 1 strings. With a 0 corresponding to a receiver and a 1 corresponding to a transmitter, it is easy to see that each 0-1 string corresponds to a different transmit-receive configuration in which all the nodes are involved. The all 0 and all 1 strings correspond to the all-receive and all-transmit configurations, which we are not interested in. We note that  $p(N)$  still grows exponentially, but not as rapidly as  $q(N)$ .

## D.1 Minimum-Frame-Length Scheduling

The minimum-frame-length (MFL) scheduling problem discussed in Chapter 2 can be represented compactly using transmission matrices. Consider the set of links in the PRNET shown in Fig. D-1. Those links can be represented by the non-admissible transmission matrix  $\mathbf{S}$  where



**Figure D-1:** Links in a frame as dictated by topology.

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The frame structure shown in Fig. D-2 can be viewed as decomposing  $\mathbf{S}$  into four admissible transmission matrices:

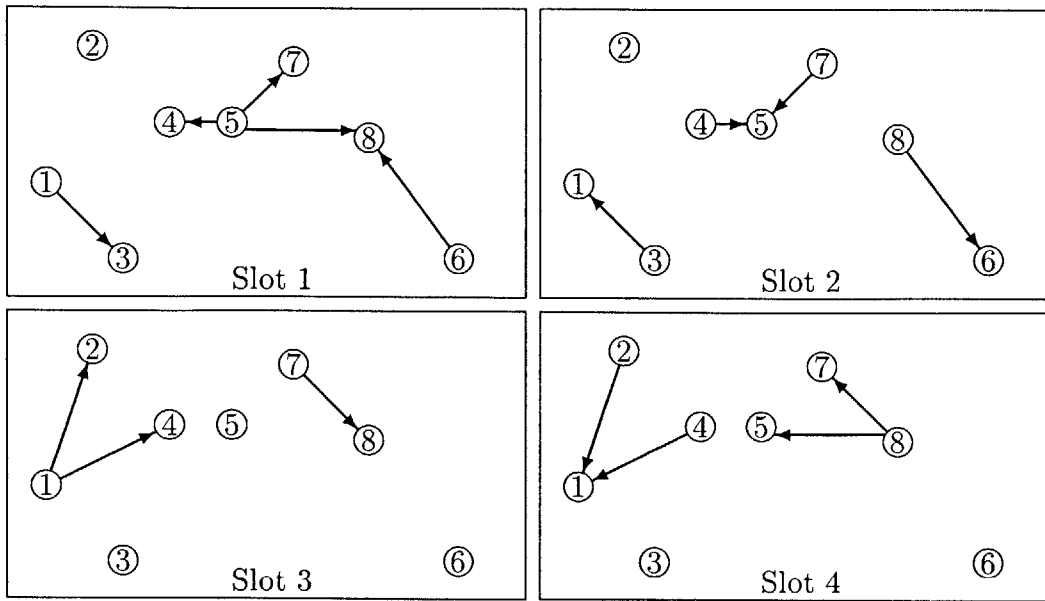
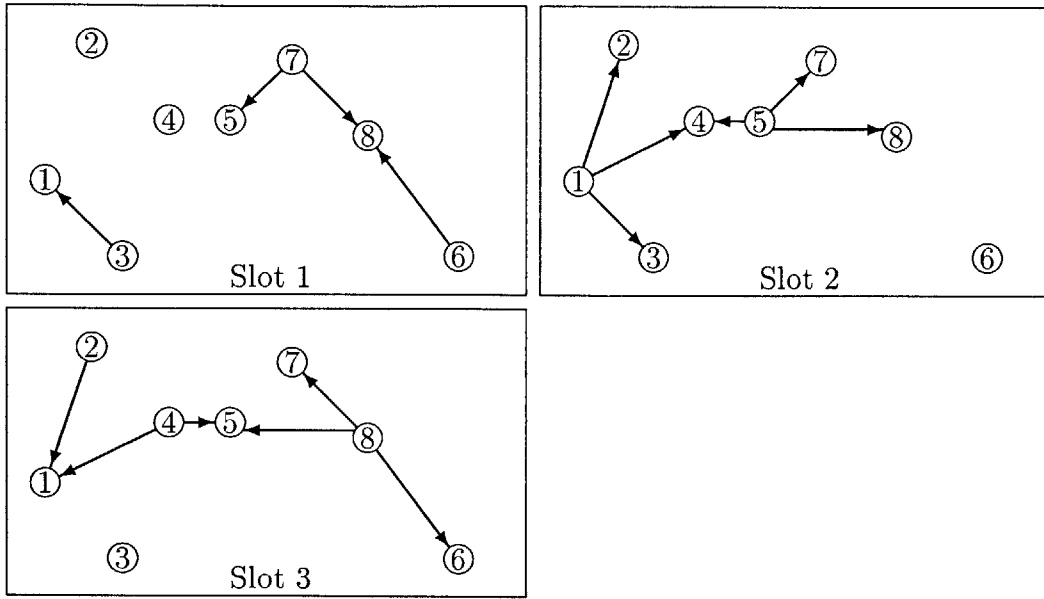


Figure D-2: A feasible frame consisting of four slots.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \\
+ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

whereas the minimum-length frame shown in Fig. D-3 corresponds to decomposing  $S$  into three admissible transmission matrices:



**Figure D-3:** A feasible frame consisting of minimum number of slots.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Therefore the minimum-frame-length scheduling problem can be reformulated as the problem of how to decompose a matrix  $S$ , that represents all the links in the PRNET, into the smallest possible number of admissible transmission matrices.

# Appendix E

## Proof of Complexity of Minimum-Frame-Length Scheduling

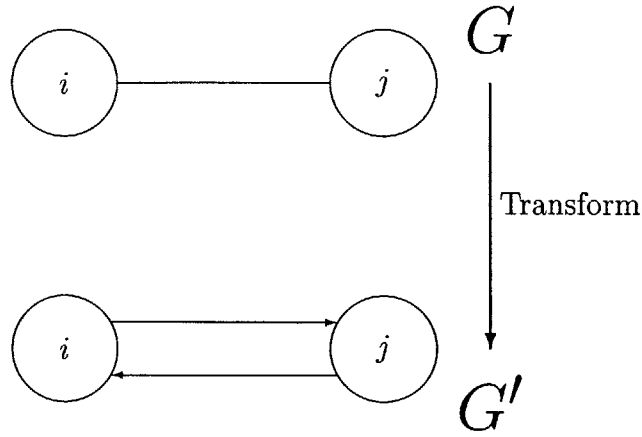
In this appendix we prove the following theorem from Chapter 2:

**Theorem E.1** *The problem of Minimum-Frame-Length Scheduling is NP-Complete.*

The proof of Theorem E.1 is based on the coloring of undirected graphs. A coloring of an undirected graph  $G$  is an assignment of colors to the vertices of the graph, such that no two vertices joined by an arc have the same color. Two non-adjacent vertices (i.e., not linked by an arc) may have the same color. The minimum number of colors required to color the graph is called the chromatic number of the graph and is denoted by  $\chi(G)$ . One problem of interest is determining whether or not  $\chi(G) \leq 3$ . This problem is referred to as the Graph 3-Colorability problem, and is known to be an NP-Complete problem [GaJ76], [GaJ79].

Next we link the problems of Minimum-Frame-Length Scheduling (MFL Scheduling) and Graph 3-Colorability. Let  $G$  be an undirected graph. We define  $G'$  to be

$$G' = \mathcal{F}(G)$$



**Figure E-1:** Transforming an undirected edge to two directed edges.

where  $\mathcal{F}$  transforms an undirected graph into a directed graph by replacing each undirected arc with two directed edges (in opposite directions) as shown in Fig. E-1. Note that this transformation can be done in polynomial time.

Let  $K(G')$  be the number of slots in the MFL scheduling of  $G'$ . We use the following lemmas to complete our proof:

**Lemma E.1**  $K(G') = 2$  if and only if  $\chi(G) = 2$ .

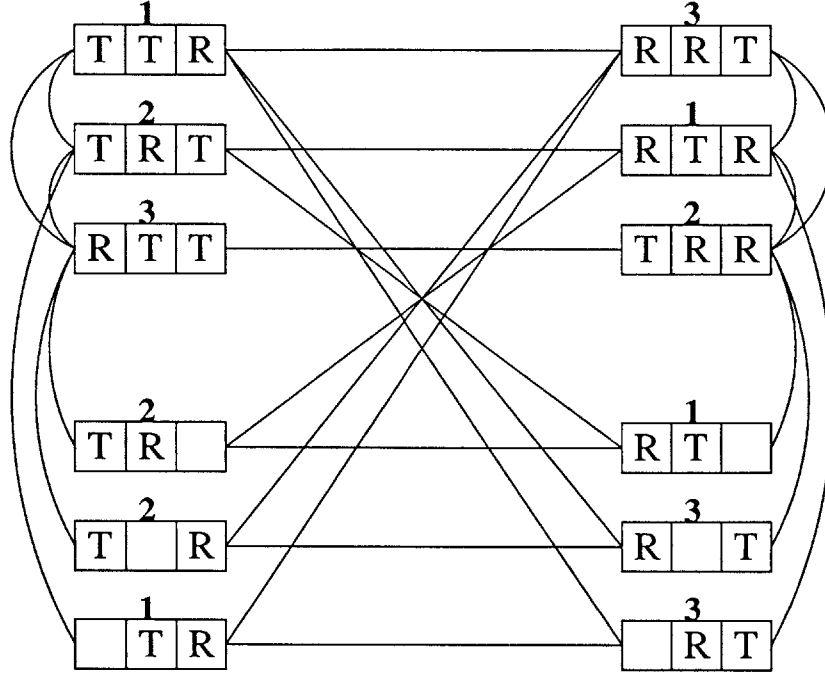
**Proof:** If  $\chi(G) = 2$  then we can construct a feasible transmission schedule which consists of two slots. In the first slot, nodes with Color 1 transmit, and in the second nodes with Color 2 transmit. It is not possible to have a schedule with one slot in it, since each node needs to transmit and receive, requiring at least two slots. Therefore if  $\chi(G) = 2$  then  $K(G') = 2$ .

If  $K(G') = 2$ , then we can give the nodes that transmit in the first slot the Color 1, and in the second the Color 2, implying that  $\chi(G) = 2$ . **Q.E.D.**

**Lemma E.2**  $K(G') = 3$  if and only if  $\chi(G) = 3$ .

**Proof:** If  $\chi(G) = 3$  then from Lemma E.1  $K(G') \geq 3$ . Suppose the colors are 1, 2, and 3. One feasible schedule for  $G'$  is to have three slots. In the  $k^{\text{th}}$  slot ( $k = 1, 2, 3$ ) the nodes with Color  $k$  transmit to their neighbors. Therefore,  $K(G') = 3$ .





**Figure E-2:** Activity configurations for nodes.

If  $K(G') = 3$  then from Lemma E.1  $\chi(G) \geq 3$ . We show that  $\chi(G) = 3$  by finding a feasible way to color  $G$  with three colors. We will say a node is engaged in a slot if it is either receiving or transmitting in that slot. Since each node needs to transmit and receive within the span of a frame, each node is engaged in at least two slots. If a node is engaged in three slots, it will either transmit in two and receive in one, or receive in two and transmit in one. Fig. E-2 shows all the possible engagement configurations for a node. Furthermore, the arcs in Fig. E-2 link engagement configurations permissible for two neighbors. Above each engagement configuration in Fig. E-2 is one of three colors. Note that engagement configurations that are linked have different colors. Therefore, by assigning the color associated with a node's engagement configuration to that node, we have feasible coloring for  $G$  that requires only three colors. **Q.E.D.**

Using Lemmas E.1 and E.2, proving Theorem E.1 becomes simple. We show that the MFL Scheduling problem is NP-Complete through the traditional method of transforming a known NP-Complete problem (in this case, the Graph 3-Colorability problem) to the problem of interest (in this case, the MFL Scheduling problem). The

transformation in this case is done in the following manner: To solve the Graph 3-Colorability problem for an undirected graph  $G$ , obtain  $G'$  through the transformation  $\mathcal{F}$  in polynomial time. Then solve the MFL Scheduling problem for  $G'$ . If  $K(G') \leq 3$  then  $\chi(G) \leq 3$ , otherwise  $\chi(G) > 3$ . Therefore, if the MFL Scheduling problem can be solved in polynomial time, then the Graph 3-Colorability problem can be solved in polynomial time.

What remains to be shown is that the MFL Scheduling problem  $\in NP$ . This is done by showing that the feasibility of a solution to an MFL Scheduling problem can be verified in polynomial time. Given the solution, we represent each slot by its transmission matrix (see Appendix D). We then test the admissibility of each transmission matrix using the test given in Theorem D.1. The final step of the verification is to ensure that each directed arc in  $G'$  is enabled within the span of the frame produced by MFL scheduling. The verification can be completed in polynomial time.

# Bibliography

- [Ari84] E. Arikan, "Some Complexity Results about Packet Radio Networks," *IEEE Transactions on Information Theory*, Vol. IT-30, No. 4, July 1984.
- [BaE81] D. J. Baker and A. Ephridemes, "The Architectural Organization of a Packet Radio Network via a Distributed Algorithm," *IEEE Transactions on Communications*, Vol. COM-29, pp. 1694-1701.
- [BBS98] "The Explaining of Packet Radio,"  
<<http://www.cuci.nl/mars/packet.htm>>.
- [BeG92] D. Bertsekas and R. Gallager, "Data Networks." Prentice Hall, New Jersey, 1992.
- [Be+89] D. Beyer, M. Frankel, J. Hight, D. Lee, M. Lewis, P. McKenney, J. Naar, R. Ogier, N. Shacham, and W. Zaumen, "Packet Radio Network Research, Development and Application." *Proceedings of the SHAPE Packet Radio Symposium*, 1989.
- [Be+98] R. Berry, S. Finn, R. Gallager, H. Kassab and J. Mills, "Multiple Access Techniques in Future Army Communications." *Proceedings of Second Annual FEDLAB Symposium*, pp. 122-125, College Park, Maryland, Feb. 1998.
- [Bh+94] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs." *Proceedings SIGCOMM '94*, pp. 212-225, London, U.K., September 1994.

- [Bla99] R. E. Blahut, "Theory and Practice of Data Transmission Codes." Cambridge University Press, 1999.
- [BSS79] M. Bazaraa, H. Sherali and C. M. Shetty, "Nonlinear Programming Theory and Algorithms." John Wiley & Sons, New York, 1979.
- [CaG74] D. G. Cantor and M. Gerla, "Optimal Routing in a Packet-Switched Computer Network," *IEEE Transactions on Computers*, Vol. c-23, No. 10, October 1974.
- [ChL96] C. Chartrand and L. Lesniak, "Graphs & Digraphs." Chapman & Hall, London, 1996.
- [EWB87] A. Ephridemes, J. E. Wieselthier and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proceedings of The IEEE*, Vol. 75, No. 1, January 1987.
- [GaB87] E. M. Gafni and D. P. Bertsekas, "Asymptotic Optimality of Shortest Path Routing Algorithms," *IEEE Transactions on Information Theory*, Vol IT-33, No. 1, January 1987.
- [Gal77] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," *IEEE Transactions on Communications*, Vol. COM-25, No. 1, January 1977.
- [Gal94] R. G. Gallager, "An Inequality on The Capacity Region of Multiaccess Multipath Channels," *Technical Report LIDS-P-2240*, Laboratory for Information and Decision Systems, MIT, April 1994.
- [GaJ76] M. R. Garey and D. S. Johnson, "The Complexity of Near-Optimal Graph Coloring," *Journal of the Association for Computing Machinery*, Vol. 23, No. 1, pp. 43-49, January 1976.
- [GaJ79] M. R. Garey and D. S. Johnson, "Computers and Intractability A guide to The Theory of NP-Completeness." W.H. Freeman & Company, San Francisco, 1979.

- [Ger89] S.B. Gershwin, "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems," *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 195-209, Jan. 1989.
- [GuK97] P. Gupta and P. R. Kumar, "A System and Traffic Dependent Adaptive Routing Algorithm for Ad Hoc Networks." *Proceedings of IEEE 36th Conference on Decision and Control*, San Diego, 1997, pp. 2375-2380.
- [HaS88] B. Hajek and G. Sasaki, "Link Scheduling in Polynomial Time," *IEEE Transactions on Information Theory*, Vol. 34, No. 5, September 1988.
- [HRf99a] "Welcome to the HomeRF Web Site!," <<http://www.homerf.org/>>.
- [HRf99b] "Technical Summary of the SWAP Specification," <<http://www.homerf.org/tech/hrfwgtec.pdf>>.
- [Ibm99] "Networked Home," <<http://www-3.ibm.com/pvc/nethome/>>.
- [JuT87] J. Jubin and J. D. Tornow, "The DARPA Packet Radio Network Protocols." *Proceedings of The IEEE*, Vol. 75, No. 1, January 1987.
- [Kar90] P.Karn, "MACA - A New Channel Access Method for Packet Radio." *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, September, 1990.
- [Kat96] R. H. Katz, "Introduction to Packet Radio Networks," "Routing in Packet Radio Networks," and "Advanced Routing in Packet Radio Networks." *CS 294-7: Wireless Communications and Mobile Computing*, Lecture notes, University of California, Berkeley, Spring 1996.
- [Ka+78] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in Packet Radio Technology." *Proceedings of The IEEE*, Vol. 66, No. 11, November 1978.

- [KIS78] L. Kleinrock and J.A. Silvester, "Optimal Transmission Radii in Packet Radio Networks or Why six is a Magic Number," *Proceedings of The National Telecommunication Conference*, Birmingham, AL, Dec. 1978.
- [KIT75] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I- Carrier Sense Multiple Access Modes and Their Throughput Delay Characteristics." *IEEE Transactions on Communications*, Vol. COM-24, pp. 1400-1416, Dec. 1975.
- [KPD85] P. R. Karn, H. E. Price and R. J. Diersing, "Packet Radio in The Amateur Service." *IEEE Journal on Selected Areas in Communications*, Vol. SAC-3, No. 3, May 1985.
- [Lau95] G. Lauer, "Packet-Radio Routing," in "Routing in Communications Networks" (ed. M. Steenstrup). Prentice-Hall, New Jersey, 1995.
- [Lee82] W. C. Y. Lee, "Mobile Communications Engineering." McGraw-Hill, New York, 1982.
- [LNT87] B. Leiner, D. L. Nielson and F. A. Tobagi, "Issues in Packet Radio Network Design," *Proceedings of The IEEE*, Vol. 75, No. 1, January 1987.
- [Pur87] M. B. Pursley, "The Role of Spread Spectrum in Packet Radio Networks." *Proceedings of The IEEE*, Vol. 75, No. 1, January 1987.
- [PuR93] M. B. Pursley and H. B. Russell, "Routing in Frequency-Hop Packet Radio Networks with Partial-Band Jamming." *IEEE Transactions on Communications*, Vol. 41, No. 7, July 1993.
- [Par85] A. K. Parekh, "Minimizing The Number of Clusters in Mobile Packet Radio Networks," LIDS Report LIDS-TH-1502, 1985. Available from MIT Laboratory for Information and Decision Systems, Cambridge, MA.
- [Pro95] J. G. Proakis, "Digital Communications." McGraw-Hill, New York, 1995.

- [RaL93] S. Ramanathan and E. L. Lloyd, "Scheduling Algorithms for Multihop Radio Networks," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 2, April 1993.
- [ScB95] K. Scott and Nicholas Bambos, "The Self-Organizing Wireless Network (SWAN) Protocol for Communication Among Mobile Users," *Proceedings of GLOBECOM '95*, pp. 355-359, Singapore, November 1995.
- [SeH97] A. Sen and M. L. Huson, "A New Model for Scheduling Packet Radio Networks," *Wireless Networks*, 3 (1997) pp. 71-82.
- [She96] T. J. Shepard, "A Channel Access Scheme for Large Dense Packet Radio Networks." *Proceedings of ACM SIGCOMM'96*, August 1996.
- [SiK83] J. A. Silvester and L. Kleinrock, "On the Capacity of Multihop Slotted ALOHA Networks with Regular Structure." *IEEE Transactions on Communications*, Vol. COM-31, No. 8, August 1983.
- [SIP87] B. M. Leiner, D. L. Nielson and F. A. Tobagi (Eds.), *Proceedings of the IEEE - Special Issue on Packet Radio Networks*, Vol. 75, No. 1, January 1987.
- [Sou90] E. S. Sousa, "Interference Modeling in a Direct-Sequence Spread-Spectrum Packet Radio Network", *IEEE Transactions on Communications*, Vol. 38, No. 9, September 1990.
- [Ste88] J. A. Stevens, "Least Interference Routing," *Proceedings of the Tactical Communications Conference*, May 1988, pp. 42-51.
- [Ste85] J. A. Stevens, "Spreading Connectivity Information out over Multiple PROP Periods, and Timelines of Information," *SURAN Program Technical Note (SRNTN) 21*, Richardson, TX: Rockwell Inc., May 1985. Available from Defense Technical Information Center (DTIC).
- [Ste95a] R. Steele, "Mobile Radio Communications." Pentech Press, London, 1995.

- [TaK84] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals." *IEEE Transactions on Communications*, Vol. COM-32, pp. 246-257, March 1984.
- [TAP99] "Tucson Amateur Packet Radio (TAPR) Home Page," <<http://www.tapr.org/tapr/html/>>.
- [Tob87] F. A. Tobagi, "Modeling and Performance Analysis of Multihop Packet Radio Networks." *Proceedings of The IEEE*, Vol. 75, No. 1, January 1987.
- [TsB86] J. N. Tsitsiklis and D. P. Bertsekas, "Distributed Asynchronous Optimal Routing in Data Networks." *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 4, April 1986.
- [Tsu88] P. F. Tsuchiya, "The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks," *Proceedings of SIGCOMM '88*, August 1988, Stanford, CA, pp. 35-42.
- [Vit95] A. J. Viterbi, "CDMA Principles of Spread Spectrum Communication." Addison-Wesley Publishing Company, Massachusetts, 1995.
- [WBE94] J. E. Wieselthier, C. M. Barnhart and A. Ephridemes, "A Neural Network Approach to Routing Without Interference in Multihop Radio Networks," *IEEE Transactions on Communications*, Vol. 42, No. 1, January 1994.
- [Yat95] R. D. Yates, "A Framework for Uplink Power Control in Cellular Radio Systems," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 7, pp. 1341-1348, September 1995.
- [Yat96] R. D. Yates, "Power Controlled Packet Radio Networks," *The Gallager Intellectual Fest - Celebrating Robert Gallager's 65th Birthday*, May 1996, Cambridge, MA. Available from MIT Laboratory for Information and Decision Systems, Cambridge, MA.