

# Joint MAC Scheduling and Network Coding: From Wired Networks to Realistic Wireless Networks

by

Raheleh Niati

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs  
in partial fulfillment of the requirements for the degree of  
**Doctor of Philosophy in Electrical and Computer Engineering**

Carleton University  
Ottawa, Ontario

©2012

Raheleh Niati

## Abstract

In this work, we explore different aspects and steps of building a joint network coding and scheduling solution for wireless multihop networks. Network coding allows transmission networks to combine data at intermediate network nodes. In contrast to routing, where nodes only relay and replicate data in network nodes, network coding provides an opportunity to improve a performance measure of interest like the throughput or the transmission energy.

The focus of this work is on formulating a joint linear optimization problem for wireless networks. The optimization problem is formulated on a wired graph that represents the wireless graph, allowing us to apply known code-design techniques for wired networks. We also study network codes design methods for the equivalent wired graph which satisfy two constraints: wireless broadcast links have to transmit the same (coded) information, and the transfer of codes from the wired back to the wireless domain has to correctly deal with scheduling sets with unequal timeshares. Furthermore, we compare the performance and run-time complexities of the two most well-known interference models in wireless networks, *physical model* and *protocol model*, for scheduling transmissions. The performance of the protocol model is very close to that of the physical model in throughput maximization problems, while its run-time complexity is significantly lower. Therefore, the protocol model can be used in those problems without loss of the performance. In minimum energy problems however, the run-time complexities of the two model are not very different and the performance of the protocol model is worse than that of the physical model. Therefore, using the physical model is a better choice when studying the minimum energy problem. Finally, we propose a new scheduling heuristic that outperforms the existing heuristics for networks with central schedulers. Our new scheduling method, *capacity-bundling scheduling*, builds scheduling sets based on their link capacities, which results in improved throughput in throughput maximization problems and energy savings in energy minimization problems. Our results show that this improvement is up to about 80% for the throughput, and about 55% for the energy consumption in the scenarios simulated in this work.

## Acknowledgments

I would like to express my deepest gratitude to my supervisors Dr. Amir H. Banihashemi, and Dr. Thomas Kunz for their excellent guidance and advices. This work would not have been possible without their patience and continuous and tireless support.

I would also like to thank my friends and colleagues for their invaluable support. Finally, I would like to dedicate this thesis to my dear husband, Shahab, my parents, Maryam and Javad, and my sisters, Atefeh and Bahareh, for their sacrifices, support, and encouragement during my studies.

# Table of Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b> |
| 1.1      | Research Motivations and Contributions . . . . .   | 2        |
| 1.1.1    | Formulating a Joint Linear Optimization Problem . . . . .  | 2        |
| 1.1.2    | Unequal Scheduling Timeshares and Network Code Design Requirements for Wireless Networks . . . . .                               | 3        |
| 1.1.3    | Performance Comparison of Physical and Protocol Interference Models in Joint Scheduling and Network Coding Optimization Problems | 3        |
| 1.1.4    | Capacity-bundling Scheduling: An Improved MAC Scheme for Joint Scheduling and Network Coding in Wireless Networks . . . . .      | 4        |
| 1.2      | Published and Submitted Work . . . . .   | 4        |
| 1.2.1    | Published Journal Paper . . . . .  | 4        |
| 1.2.2    | Submitted Journal Paper . . . . .  | 5        |
| 1.2.3    | Published Conference Papers . . . . .  | 5        |
| 1.3      | Organization . . . . .   | 5        |
| <b>2</b> | <b>Problem Formulation</b>   | <b>7</b> |
| 2.1      | Network Coding and its Advantages in Wired and Wireless Networks . . . . .   | 7        |
| 2.2      | Notations and Network Model . . . . .  | 10       |
| 2.3      | The Min-cut Max-flow Theorem . . . . .   | 11       |
| 2.4      | Related Work . . . . .   | 13       |
| 2.5      | Research Approach . . . . .  | 16       |

|          |  |           |
|----------|--|-----------|
| <b>3</b> | <b>Joint MAC Scheduling and Network Coding Optimization Problem</b>  | <b>17</b> |
| 3.1      | Wireless Channel Model . . . . .   | 18        |
| 3.2      | Wireless Interference Model and Scheduling . . . . .   | 19        |
| 3.3      | Joint Linear Optimization Problem with Schedule-Specific Flows . . . . .   | 21        |
| 3.4      | Optimization Problem for Minimizing the Energy . . . . .   | 25        |
| 3.5      | Conclusions . . . . .  | 27        |
| <b>4</b> | <b>Unequal Scheduling Timeshares and Requirements of Constructing Network Codes for Wireless Networks</b>                                    | <b>28</b> |
| 4.1      | Review of Wireless Network Code Construction . . . . .   | 29        |
| 4.2      | Wireless-Aware Network Code Construction . . . . .   | 35        |
| 4.3      | Unequal Time Fractions in Wireless Multicast Networks: Maximizing the Throughput . . . . .   | 38        |
| 4.3.1    | Varying Number of Destination Nodes . . . . .  | 39        |
| 4.3.2    | Varying Neighboring Node Densities . . . . .   | 42        |
| 4.4      | Unequal Time Fractions in Wireless Multicast Networks: Minimizing the Energy Per Working Cycle . . . . .                                     | 43        |
| 4.4.1    | Minimizing the Energy for a Set of Fixed Rates . . . . .   | 43        |
| 4.4.2    | Varying Number of Destination Nodes . . . . .  | 46        |
| 4.4.3    | Varying Neighboring Node Densities . . . . .   | 49        |
| 4.5      | Conclusions . . . . .  | 51        |
| <b>5</b> | <b>Performance Comparison of Physical and Protocol Interference Models in Wireless Networks with Joint MAC Scheduling and Network Coding</b> | <b>52</b> |
| 5.1      | Related Work . . . . .   | 53        |
| 5.2      | Channel and Wireless Interference Models . . . . .   | 54        |
| 5.2.1    | Channel Model . . . . .  | 54        |
| 5.2.2    | Interference Models . . . . .  | 55        |
| 5.3      | Scheduling for a Network with Variable-Rate, Variable-Power Nodes . . . . .  | 57        |

|          |  |            |
|----------|--|------------|
| 5.3.1    | Scheduling with Physical Interference Model . . . . .  | 58         |
| 5.3.2    | Scheduling with Protocol Interference Model . . . . .  | 60         |
| 5.4      | Joint Scheduling and Network Coding Optimization Problem . . . . .   | 62         |
| 5.5      | Simulation Results . . . . .   | 65         |
| 5.5.1    | Determining the Optimal Interference Range for the Protocol Model  | 67         |
| 5.5.2    | Throughput Optimization with Protocol and Physical Models . . . .  | 69         |
| 5.5.3    | Energy Optimization with Protocol and Physical Models . . . . .  | 73         |
| 5.6      | Conclusions . . . . .  | 77         |
| <b>6</b> | <b>Capacity-bundling Scheduling: An Improved MAC Scheme for Joint<br/>Scheduling and Network Coding in Wireless Networks</b> | <b>78</b>  |
| 6.1      | Capacity Bundling Scheduling . . . . .   | 79         |
| 6.1.1    | Basic Scheduling for a Network with Variable-Rate, Variable-Power<br>Nodes . . . . .   | 79         |
| 6.1.2    | Capacity-bundling for a Network with Variable-Rate, Variable-Power<br>Nodes . . . . .  | 80         |
| 6.2      | Simulation Results . . . . .   | 83         |
| 6.2.1    | Throughput Maximization with and without Capacity-Bundling . .   | 83         |
| 6.2.2    | Energy Minimization with and without Capacity-Bundling . . . . .   | 88         |
| 6.3      | Conclusions . . . . .  | 91         |
| <b>7</b> | <b>Conclusions and Future Work</b>   | <b>92</b>  |
| 7.1      | Conclusions . . . . .  | 92         |
| 7.2      | Future Work . . . . .  | 93         |
| <b>A</b> | <b>Derivations for Chapter 3</b>   | <b>100</b> |
| A.1      | Determining the Size of the Simulation Field Based on the Average Number<br>of Neighbor Nodes . . . . .                      | 100        |
|          | <b>List of References</b>  | <b>102</b> |

## List of Figures

|      |   |    |
|------|---|----|
| 2.1  | The butterfly network . . . . .   | 9  |
| 2.2  | Wireless network example . . . . .  | 10 |
| 3.1  | A wireless network example . . . . .  | 19 |
| 3.2  | Network realizations for the network of Figure 3.1. . . . .   | 22 |
| 3.3  | Wireless node $i$ broadcasts to nodes $l$ and $m$ during the 3rd timeshare . . .  | 24 |
| 3.4  | Wireless broadcast model of node $i$ . . . . .  | 25 |
| 4.1  | Coding coefficients in a) wired networks, b) wireless networks . . . . .  | 30 |
| 4.2  | a) Equivalent wired graph corresponding to the wireless network graph of<br>Fig. 3.1, b) Capacity achieving coding coefficients for the wired network . . | 33 |
| 4.3  | Network coding solution for the wireless network in Figure 3.1 . . . . .  | 34 |
| 4.4  | Graph $N^{ex}$ of the graph in Figure 4.2-a . . . . .   | 38 |
| 4.5  | Percentage of networks that require unequal timeshares for optimal through-<br>put (the average number of neighboring nodes is $D = 6$ ). . . . .         | 40 |
| 4.6  | Throughput improvement using unequal timeshares (the average number of<br>neighboring nodes is $D = 6$ ). . . . .   | 41 |
| 4.7  | Percentage of networks that require unequal timeshares for optimal through-<br>put ( $n_D = 2$ ). . . . .   | 42 |
| 4.8  | Throughput improvement using unequal timeshares ( $n_D = 2$ ). . . . .  | 43 |
| 4.9  | Minimum energy to deliver a fixed rate ( $n_D = 2$ , $D = 7$ ). . . . .   | 44 |
| 4.10 | Energy saving using unequal timeshares ( $n_D = 2$ , $D = 7$ ). . . . .   | 45 |

|      |  |    |
|------|--|----|
| 4.11 | Percentage of networks that require unequal timeshares to minimize the energy consumption ( $n_D = 2, D = 7$ ). . . . .  | 46 |
| 4.12 | Minimum energy to deliver a fixed rate ( $n_D = \{2, 5, 8\}, D = 7, r = 0.8 \times r_{eq}$ ). . . . .  | 47 |
| 4.13 | Energy saving using unequal timeshares ( $n_D = \{2, 5, 8\}, D = 7, r = 0.8 \times r_{eq}$ ). . . . .  | 48 |
| 4.14 | Percentage of networks that require unequal timeshares to minimize the energy consumption ( $n_D = \{2, 5, 8\}, D = 7, r = 0.8 \times r_{eq}$ ). . . . .           | 48 |
| 4.15 | Minimum energy to deliver a fixed rate ( $n_D = 2, D = \{6, 8, 10\}, r = 0.8 \times r_{eq}$ ). . . . .   | 49 |
| 4.16 | Energy saving using unequal timeshares ( $n_D = 2, D = \{6, 8, 10\}, r = 0.8 \times r_{eq}$ ). . . . .   | 50 |
| 4.17 | Percentage of networks that require unequal timeshares to minimize the energy consumption ( $n_D = 2, D = \{6, 8, 10\}, r = 0.8 \times r_{eq}$ ). . . . .          | 50 |
| 5.1  | Comparing the maximum throughput optimization results for different interference ranges, $N = 50, D = \{4, 6, 8\}$ . . . . .                                       | 68 |
| 5.2  | Comparing the maximum throughput optimization results for different interference ranges, $N = 100, D = \{4, 6, 8\}$ . . . . .                                      | 68 |
| 5.3  | Comparing the effect of changing $\delta$ in maximum throughput optimization problems, $N = 50, D = 6$ . . . . .   | 69 |
| 5.4  | Comparing the maximum throughputs of the different interference models . . . . .   | 71 |
| 5.5  | The average number of removed links from realizations based on the protocol model . . . . .  | 72 |
| 5.6  | Comparing the CPU times when using different interference models . . . . .   | 72 |
| 5.7  | Comparing the minimum energy required to deliver a fixed rate for different interference models , $r = \{0.7, 0.8, 0.9\} \times r_{apl}, D = 6, n_D = 2$ . . . . . | 75 |
| 5.8  | Comparing the minimum energy required to deliver a fixed rate for different interference models , $r = 0.7 \times r_{apl}, D = \{4, 6, 8\}, n_D = 2$ . . . . .     | 76 |
| 5.9  | Comparing the CPU times when using different interference models, $r = 0.7 \times r_{apl}, D = \{4, 6, 8\}, n_D = 2$ . . . . .                                     | 76 |
| 6.1  | Wireless butterfly network with different link capacities . . . . .  | 81 |



|      |  |    |
|------|--|----|
| 6.2  | A set of realizations using basic scheduling . . . . .   | 81 |
| 6.3  | A set of additional realizations based on scheduling links with equal capacities   | 82 |
| 6.4  | Comparing the average throughput of basic scheduling with that of capacity-bundling, $D = \{4, 6, 8\}$ , $n_D = 2$ . . . . .   | 84 |
| 6.5  | The throughput improvement of capacity-bundling over basic scheduling, $D = \{4, 6, 8\}$ , $n_D = 2$ . . . . .   | 85 |
| 6.6  | Average number of active realizations in different scheduling methods, $D = 6$ , $n_D = 2$ . . . . .   | 86 |
| 6.7  | Comparing the percentage of realizations with links that under-utilize their capacity, $D = \{4, 6, 8\}$ , $n_D = 2$ . . . . .   | 87 |
| 6.8  | The average normalized under-utilized capacity for basic and capacity-bundling scheduling methods, $D = \{4, 6, 8\}$ , $n_D = 2$ . . . . .   | 87 |
| 6.9  | Comparing the minimum energy required to deliver a fixed rate using basic and capacity-bundling scheduling methods, $r = \{0.7, 0.8, 0.9\} \times r_B$ , $D = 6$ , $n_D = 2$ . . . . . | 89 |
| 6.10 | Comparing the minimum energy required to deliver a fixed rate using basic and capacity-bundling scheduling methods, $r = 0.7 \times r_B$ , $D = \{4, 6, 8\}$ , $n_D = 2$ .             | 90 |
| 6.11 | The energy saving of capacity-bundling compared to basic scheduling, $r = 0.7 \times r_B$ , $D = \{4, 6, 8\}$ , $n_D = 2$ . . . . .  | 91 |

## List of Acronyms

| Acronym | Explanation                             |
|---------|---|
| AWGN    | Additive White Gaussian Noise           |
| BER     | Bit Error Rate                          |
| CPU     | Central Processing Unit                 |
| DAG     | Directed Acyclic Graph                  |
| PSK     | Phase Shift Keying                      |
| LP      | Linear Program                          |
| MAC     | Media Access Control                    |
| MFSK    | Multiple Frequency Shift Keying         |
| MQAM    | M-ary Quadrature Amplitude Modulation   |
| NP      | Non-deterministic Polynomial-Time       |
| o-d     | origin to destination                   |
| PRN     | Packet Radio Network                    |
| SINR    | Signal to Interference plus Noise Ratio |
| SNR     | Signal to Noise Ratio                   |
| XOR     | eXclusive OR                            |

## List of Symbols

| Symbol             | Explanation  |
|--------------------|--|
| $\mathbf{a}$       | A vector   |
| $\mathbf{A}$       | A matrix   |
| $\mathbf{I}$       | Identity function                                      |
| $\mathcal{U}(x)$   | Unit step function of $x$                              |
| $\mathbf{A}_{ij}$  | Element at row $i$ and column $j$ of $\mathbf{A}$      |
| $G$                | A Graph  |
| $V$                | Vertices of graph $G$                                  |
| $E$                | Edges of graph $G$                                     |
| $(i, j)$           | Graph edge starting at node $i$ and ending at node $j$ |
| $ E $              | Cardinality of the edges in a network graph            |
| $s$                | Source node  |
| $N$                | Total number of nodes in a network                     |
| $\mathcal{M}$      | Total number of realizations                           |
| $\mathcal{M}_{ph}$ | Number of realizations based on the physical model     |
| $\mathcal{M}_{pl}$ | Number of realizations based on the protocol model     |
| $\mathcal{M}_a$    | Number of active realizations                          |
| $M$                | Constellation size of MQAM modulation                  |
| $d_i$              | $i$ th destination node                                |
| $n_D$              | Number of destination nodes in a wireless network      |
| $\mathcal{D}$      | Set of destination nodes in a wireless network         |
| $D$                | Average number of neighboring nodes in a network       |
| $\mathbb{F}_q$     | Finite field of size $q$                               |
| $\oplus$           | XOR operation  |
| $In(i)$            | Input edges to node $i$                                |
| $Out(i)$           | Output edges from node $i$                             |
| $tail(l)$          | Network node that edge $l$ starts from                 |
| $head(l)$          | Network node that edge $l$ ends at                     |
| $\mu(v)$           | Discrete and independent random processes at node $v$  |

|                       |  |
|-----------------------|--|
| $X(v, l)$             | One of the $h$ source processes ( $X_i$ s) that exists at node $v$<br>( $l$ counts the number of available source processes at $v$ ) |
| $Y(e)$                | Edge process belonging to edge $e$   |
| $Z(v, j)$             | One of the $h$ output processes at destination node $v$ ( $v \in D$ )  |
| $\alpha_l(e)$         | Scalar local coding coefficient on edge $e$ (applied to source streams)  |
| $\beta(e, \acute{e})$ | Scalar local coding coefficient between edge $e$ and $\acute{e}$   |
| $\epsilon_j(e)$       | Scalar decoding coefficient at edge $e$  |
| $Z(d, i)$             | $i$ th output symbol at destination node $d$   |
| $\mathbf{C}_d$        | Network codes transfer matrix for destination $d$  |
| $f_j^{d_i}$           | The $j$ -th edge-disjoint path between the source node and destination node $d_i$  |
| $c_{u,v}$             | Capacity of edge $(u, v)$  |
| $f_{u,v}$             | Total flow on edge $(u, v)$  |
| $f_{u,v}(d_i)$        | Fractional flow on edge $(u, v)$ destined to destination $d_i$   |
| $d_{ij}$              | Distance between nodes $i$ and $j$   |
| $R_i^{in}$            | Inner radio range of node $i$  |
| $R_i^{out}$           | Outer radio range of node $i$  |
| $R_i$                 | Radio range of node $i$  |
| $R^{(q)}$             | Transmission range of constellation size $M_q$   |
| $\hat{R}_i$           | Interference range for node $i$  |
| $\tau_m$              | $m$ -th time fraction  |
| $N_i^f$               | The $i$ -th network realizations for a wireless graph  |
| $N^f$                 | A sufficient set of network realizations for a wireless graph  |
| $V_m^f$               | Set of all nodes in realization $N_m^f$  |
| $E_m^f$               | Set of all links in realization $N_m^f$  |
| $T^{(m)}$             | Set of transmitters in realization $N_m^f$   |
| $R^{(m)}$             | Set of receivers in realization $N_m^f$  |
| $P_{i,j}$             | Required power to transmit from node $i$ to $j$  |
| $P_i^{(m)}$           | Power of transmitting from node $i$ during time fraction $\tau_m$  |
| $C_i^{(m)}$           | Broadcast capacity from node $i$ during time fraction $\tau_m$   |

|                              |   |
|------------------------------|---|
| $C_{ij}^{(m)}$               | Transmission capacity of link $(i, j)$ during time fraction $\tau_m$                              |
| $V_m^{f,(T)}$                | Active transmitters during time fraction $\tau_m$   |
| $P(m)$                       | Summation of transmission powers during time fraction $\tau_m$                                    |
| $P_{max}$                    | Maximum transmission power of the wireless nodes  |
| $\alpha_j^{(m,n)}$           | Coding coefficient from the $m$ -th realization to the $n$ -th (for the $j$ -th source stream)    |
| $\beta^{(m,n)}((u, v))$      | Coding coefficient on link $(u, v)$ and from realization $m$ to realization $n$                   |
| $\epsilon_k^{(m,n)}((u, d))$ | $k$ -th decoding coefficient on link $(u, d)$ and from realization $m$ to realization $n$         |
| $X_j^{(m)}(v)$               | $j$ -th source process at node $v$ during the $m$ -th realization                                 |
| $Y^{(n)}(v)$                 | Encoded flow at node $v$ and during realization $N_n^f$   |
| $Z_k^{(n)}(v)$               | $k$ -th Output process at node $v$ during the $n$ -th realization                                 |
| $N^g$                        | Hypothetical wired graph  |
| $N^{ex}$                     | Expanded Graph  |
| $f_{i,j}^{(m)}$              | Total flow on link $(i, j)$ during the $m$ -th realization  |
| $f_{i,j}^{(m)}(d)$           | Fractional flow on link $(i, j)$ , destined to destination $d$ and during the $m$ -th realization |
| $f_{i,j}^{(m),l}(d)$         | $f_{i,j}^{(m)}(d)$ in the $l$ -th iteration   |
| $f_i^{(m)}$                  | Broadcast flow from node $i$  |
| $\tau_i^{int}$               | Integer representation value of time fraction $\tau_i$  |
| $E_{m,i}^b$                  | Broadcast links of node $i$ during time fraction $\tau_i$   |
| $V_{m,i}^b$                  | Set of nodes that receive broadcast messages from node $i$ during time fraction $\tau_m$          |
| $h$                          | Integer throughput  |
| $r$                          | Throughput  |
| $r_{opt}$                    | Optimal value of network flow   |
| $r_{eq}$                     | Value of network flow with equal time fractions   |
| $r_{eqT}$                    | The best network flow among all combinations of equal time fractions                              |
| $\eta_r$                     | Normalized throughput improvement (comparing $r_{opt}$ and $r_{eq}$ )                             |

|                         |   |
|-------------------------|---|
| $\eta_{rT}$             | Normalized throughput improvement (comparing $r_{opt}$ and $r_{eqT}$ )            |
| $\mathcal{E}_{opt}$     | Optimal minimum energy  |
| $\mathcal{E}_{opt}$     | Minimum energy solution among all combinations of equal time fractions            |
| $\eta_{\mathcal{E}}$    | Normalized energy saving (comparing $\mathcal{E}_{eq}$ and $\mathcal{E}_{opt}$ )  |
| $\mathcal{E}_{CB}$      | Minimum energy with capacity-bundling scheduling                                  |
| $\mathcal{E}_B$         | Minimum energy with basic scheduling  |
| $\eta_{\epsilon(CB-B)}$ | Energy saving of capacity-bundling method compared with basic scheduling          |
| $B_{ij}$                | Bandwidth of link $(i, j)$ in Hz  |
| $\gamma_{ij}$           | SINR of the signal transmitted from node $i$ and received by node $j$             |
| $\gamma_{t,ij}$         | Target SINR of the signal transmitted from node $i$ and received by node $j$      |
| $\gamma_q$              | Target SNR of constellation size $M_q$  |
| $p_{b,ij}$              | Target BER for link $(i, j)$  |
| $\sigma^2$              | Power of the AWGN noise   |
| $a_{ij}$                | Power gain of link $(i, j)$   |
| $\alpha$                | Pathloss exponent   |
| $X$                     | Set of active transmitters in a network   |
| $d_0$                   | Reference distance for the transmitting antenna's far field                       |
| $\lambda$               | Wavelength  |
| $\zeta$                 | Average normalized under-utilized capacity  |
| $r_{apl}$               | Maximum achievable throughput with the protocol model                             |
| $r_{CB}$                | Throughput with capacity-bundling scheduling                                      |
| $r_B$                   | Throughput with basic scheduling  |
| $\eta_{r(CB-B)}$        | Throughput improvement of capacity-bundling method compared with basic scheduling |

## Chapter 1

### Introduction

During the past decade, network coding has become the focus of many research groups in communications and computer networks fields. Starting with the pioneering work of Ahlswede *et al.* [1], network coding is currently well established for wired networks. Much of the research in this area is devoted to a) finding the optimal performance of a network under a certain criterion, where the network nodes are capable of performing coding operations, and b) constructing a network coding solution that achieves the optimal performance. Although these two aspects of network coding are studied in detail for wired networks, they can not be directly applied to wireless networks. Unique properties of the wireless networks such as broadcast media and the possibility of having interferences or collisions among concurrent transmissions result in additional complexities in both determining the achievable throughput and designing network codes. One important consequence is that network coding does not necessarily result in an improvement in the performance measure of interest like throughput or energy, if Media Access Control (MAC) and network coding are designed independently. This has been demonstrated in [2] for the throughput maximization problem, where the authors compare independent and joint designs of network coding and MAC. It is therefore important to create a joint optimization framework for the access to the shared media and network coding in wireless networks. The following three steps would be needed in such a framework:

1. Scheduling: Constructing a set of realizations for the wireless network.

2. Optimization: Solving a joint optimization problem based on the realizations to obtain the transmission flows and the timeshares in which realizations should be active and transmitting.
3. Coding: Constructing network codes that can achieve the optimal performance predicted by the solution to the optimization problem.

## 1.1 Research Motivations and Contributions

It is shown that the general problem of scheduling for multi-hop wireless networks is NP-hard, [3]- [5]. As a result many researchers fall back on heuristics to schedule transmission in these networks [9], [10]. In our work, we investigate each of the three mentioned steps towards jointly solving the scheduling and network coding problem in wireless networks.

### 1.1.1 Formulating a Joint Linear Optimization Problem

The joint design of MAC scheduling and network coding has been the subject of much recent research, see, e.g., [6], [7], [8], [9], and [10]. The formulations of [6], and [8] are linear but the link flows that result from solving the optimization problem are the total flows during the whole set of timeshares (one working cycle of the network). In general, such flows can not be used for coding purposes, because we need to know the exact share of each link flows in each timeshare. If a link is a member of more than one realization, then the assignment of flows to different schedules corresponding to those realizations is not specified. In [9] and [10], this issue is resolved by formulating the problem as a nonlinear and mixed-integer problem. Such a problem in general is complex to solve. In our work, we formulate a linear optimization problem that results in schedule-specific flows that can be readily used for coding purposes. Our formulation can be easily applied to different performance criteria of interest. In our work, we performed simulations to maximize the throughput or to minimize the energy in wireless networks.



### **1.1.2 Unequal Scheduling Timeshares and Network Code Design Requirements for Wireless Networks**

According to our study, the problem of having unequal scheduling timeshares is not addressed in designing network codes for wireless networks. Therefore, we first study the importance of enabling the code design approach to deal with unequal timeshares and then present requirements of the code construction algorithms for wireless networks to capture both the broadcast property of these networks and design proper codes when timeshares are unequal. Our simulations of the significance of considering unequal timeshares show that the throughput can be improved by about 35% in maximum flow problems and energy savings are between 13% and 30%, depending on the network size, in minimum energy problems.

### **1.1.3 Performance Comparison of Physical and Protocol Interference Models in Joint Scheduling and Network Coding Optimization Problems**

There are two commonly used interference models in wireless networks. They are the physical model, also known as the SINR (Signal to Interference plus Noise Ratio) model and the protocol model. The physical model is an elaborate yet computationally complex interference model whose application becomes even more complicated when it is used to schedule feasible wireless transmissions (through iterative approaches) in a network. The protocol model, on the other hand, is a simplified interference model that does not consider the wireless interferences at the physical level and hence decreases the computational complexity at the expense of possibly not having an achievable solution in actual wireless settings. In our work, we compare the joint scheduling and network coding throughput/energy optimal solutions of networks that employ a set of embedded M-ary Quadrature Amplitude Modulation (MQAM) signal constellations for transmission and use either of the two models as their interference model. Our simulation results show that the performance of the protocol model and the physical model are very close in throughput maximization problems, while the run-time complexity of the protocol model is significantly lower. Therefore, the

protocol model can be used to study those problems without sacrificing predicted performance gains. In the minimum energy problem however, the run-time complexities of the two models are quite close and the performance of the protocol model is inferior to that of the physical model. Therefore, we conclude that the physical model is still the best choice when studying the energy problem.

#### **1.1.4 Capacity-bundling Scheduling: An Improved MAC Scheme for Joint Scheduling and Network Coding in Wireless Networks**

As we explained before, the scheduling problem in general is an NP-hard problem. Therefore if heuristics are used to schedule wireless transmissions, the outcome of the optimization problems based on a sub-optimal scheduling depends on the selected set of realizations. To the best of our knowledge, there is no prior work on using the link capacities as a criterion for such a selection. The existing heuristics are agnostic of link properties, our proposed heuristic improves on that by considering one important link property, its capacity. In this work, we propose a *capacity-bundling* scheduling heuristic, where links with similar capacities are bundled together in one realization. We later demonstrate that using the proposed scheduling heuristic, results in improved throughput in throughput maximization problems and energy savings in energy minimization problems. Our results show that this improvement is up to about 80% for the throughput, and about 55% for the energy consumption in the scenarios simulated in this work.

## **1.2 Published and Submitted Work**

The following list of published and submitted works summarizes the peer-reviewed papers that appeared as a result of this work.

### **1.2.1 Published Journal Paper**

- R. Niati, A. H. Banihashemi, and T. Kunz, “Throughput and Energy Optimization in Wireless Networks: Joint MAC Scheduling and Network Coding,” *IEEE Transactions*

on *Vehicular Technology*, Vol. 61, No. 3, Mar 2011, pp. 1372-1382.

### 1.2.2 Submitted Journal Paper

- R. Niati, T. Kunz, A. H. Banihashemi, “An Improved Multiple Access Control (MAC) Scheme for Joint MAC Scheduling and Network Coding in Wireless Networks,” Submitted to *IEEE/ACM Transactions on Networking*, Feb. 2012.

### 1.2.3 Published Conference Papers

- R. Niati, T. Kunz, and A. H. Banihashemi, “The effect of transmission capacities on interference-free scheduling in wireless networks”, *Proceedings of the 9th Conference on Communications Networks and Services Research (CNSR 2011)*, Ottawa, Canada, May 2-5, 2011, pp. 257-262.
- R. Niati, A. H. Banihashemi, and T. Kunz, “Scheduling and Network Coding in Wireless Multicast Networks: A Case for Unequal Time Shares,” *Proceedings of IEEE WCNC 2011, Quintana-Roo, Mexico*, March 28-31, 2011, pp. 456-460.
- R. Niati, A. H. Banihashemi, and T. Kunz, “On Code Design in Joint MAC Scheduling and Wireless Network Coding,” *Proceedings of the 25th Biennial Symposium on Communications*, Queens University, Kingston, Canada, May 12-14, 2010, pp. 456-460.

## 1.3 Organization

The rest of this document is organized as follows: Chapter 2 introduces the problem formulation, network model and notations. Chapter 3 presents the linear joint optimization problem for throughput and energy. Chapter 4 investigates the scheduling timeshares and examines the result of a code design that is not capable of dealing with unequal timeshares. We also specify network code design requirements in this chapter. In Chapter 5, we compare the performance and run-time complexities of maximum rate or minimum energy problems

when the interference model is either the physical model or the protocol model. In Chapter 6, we propose “capacity-bundling scheduling” which is an improved scheduling method for joint scheduling and network coding optimization problems. Through simulations we show it outperforms a scheduling method whose scheduling criterion is based on considering interferences only. Finally the document concludes in Chapter 7.

## Chapter 2

# Problem Formulation

### 2.1 Network Coding and its Advantages in Wired and Wireless Networks

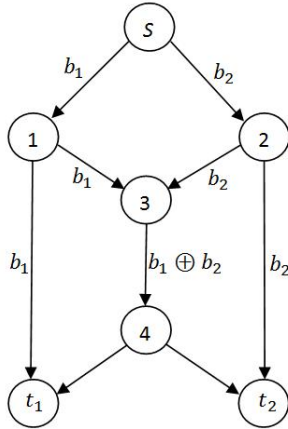
Network Coding started with the seminal paper of Ahlswede *et al.* [1] in 2000. In [1], network coding is defined as “coding at a node in a network”. By coding, they mean an arbitrary mapping from input information streams of a node to its outputs. They look at specific networks consisting of nodes connected by error-free, point-to-point links. The network model of Ahlswede *et al.* is a special case of those ordinarily studied in network information theory, and as a result, essentially all wireline networks can be cast into their model once the physical layer has been abstracted into error-free conduits for carrying bits. From another perspective, network coding can be seen as coding above the physical layer. This is unlike network information theory, which is generally concerned with coding at the physical layer.

In general, network coding is superior to routing. In network coding, routers and switches are replaced by devices called coders. Instead of directing the packets toward their ultimate destination, like the traffic flow of vehicles in streets, the coders transmit combinations of their incoming messages along multiple paths simultaneously. This distribution method can increase the effective capacity of a network by minimizing the number and severity of bottlenecks. When a receiver has enough message combinations, it can

compute the intended message/packet. Even if some packets on some of the routes are lost or corrupted, the original message gets through if there are enough coded packets available at the receivers to retrieve the original message. Network coding can improve throughput, robustness, complexity and security of transmission networks. Here we present an example to show how the throughput is improved using network coding.

The most well-known advantage of using network coding is improving the throughput. It is achieved by communicating more information with fewer packet transmissions. In this way, packet transmissions are used more efficiently. The most famous example of throughput efficiency is presented by Ahlswede *et al.* [1]. Their example, widely known as the “butterfly network”, depicts a multicast session from one source node to two destination nodes. This network is shown in Figure 2.1. Both of the destinations wish to receive all the messages from the source node. The messages are in the form of packets and each packet is a collection of bits. A capacity-achieving solution can be established only if one of the intermediate nodes (i.e., a node that is neither source nor destination) performs a coding operation. Node 3 in Figure 2.1 forms a new packet by taking the binary sum, or XOR, of its two received packets, and outputs the resulting packet. Thus if the contents of the two received packets are vectors  $\mathbf{b}_1$  and  $\mathbf{b}_2$ , then the output packet is  $\mathbf{b}_1 \oplus \mathbf{b}_2$ , formed by the bitwise XOR of  $\mathbf{b}_1$  and  $\mathbf{b}_2$ . The destinations decode by performing decoding operations on the packets that they receive. Destination  $\mathbf{t}_1$  recovers  $\mathbf{b}_2$  by taking the XOR of  $\mathbf{b}_1$  and  $\mathbf{b}_1 \oplus \mathbf{b}_2$ .

In routing, intermediate nodes are only allowed to forward copies of the received packets on their outputs. Therefore, under routing, we could communicate, for example  $\mathbf{b}_1$  and  $\mathbf{b}_2$  to  $\mathbf{t}_1$ , but we would then only be able to communicate one of  $\mathbf{b}_1$  or  $\mathbf{b}_2$  to  $\mathbf{t}_2$ .



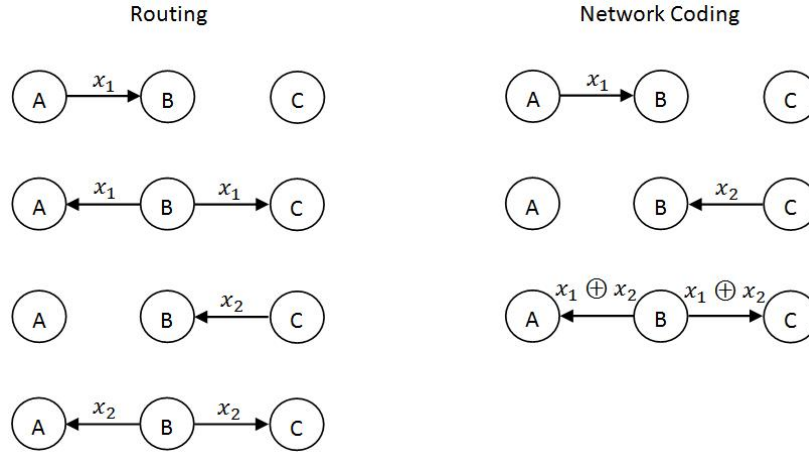
**Figure 2.1:** The butterfly network

The main features that distinguish wireless from wireline networks are shared medium and time variability. By increasing its transmission power, a wireless node will eventually reach sufficient signal-to-noise ratio to be able to transmit to every node in the network, but it will at the same time create significant interference to other nodes. Moreover, channels may vary over time due to fading or node mobility. Finally, resources such as computational power or battery life are often limited in wireless networks [14].

In a wireless environment, network coding can be used to offer benefits in terms of battery life, wireless bandwidth, and delay. We describe this with an example in Figure 2.2. Consider a wireless ad hoc network in which nodes  $A$  and  $C$  would like to exchange binary files  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . Node  $B$  is a relay node. Assuming half-duplex communication, each node can either transmit or receive at any timeslot. Figure 2.2 shows the approach with routing and network coding. The network coding approach takes advantage of the natural capability of wireless channels to broadcast and gives benefits in terms of resource utilization. Node  $C$  receives both  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , bitwise XORs them to create  $\mathbf{x}_1 \oplus \mathbf{x}_2$  and broadcasts the result to both receivers using one transmission. Node  $A$  has  $\mathbf{x}_1$  and thus can decode  $\mathbf{x}_2$ . Node  $C$  has  $\mathbf{x}_2$  and therefore can decode  $\mathbf{x}_1$  [15], [16].

This example shows benefits in terms of energy efficiency (node  $B$  transmits once instead of twice), delay (the transmission is concluded in 3 transmissions instead of 4), wireless

bandwidth (the wireless channel is occupied for a smaller amount of time), and interference (if there are other wireless nodes attempting to communicate in the neighborhood).



**Figure 2.2:** Wireless network example

Most wireless networking protocols operating today use the wireless medium to create point-to-point connections, and do not exploit the broadcasting capabilities of wireless transmissions. Network coding that uses the broadcast nature of the wireless medium is considered to be a great step forward to move from existing wireless systems towards practical information-theoretic based solutions. MIT's COPE project is the first network-coding-based wireless network that shows throughput gains over the routing solution [17]. This approach treats interference at a receiver as harmful to the system performance, and relies on appropriately scheduling the broadcast transmissions to minimize it.

In the next section we present general notations and the network model used in our work and then introduce the min-cut max-flow theorem.

## 2.2 Notations and Network Model

The focus of our studies are wireless multihop networks. We consider multicast scenarios in such networks. A multicast is transmitting common information from a set of one or more source nodes to a fixed set of destination nodes. The wireless nodes in our model have a



single transceiver per node and operate in half-duplex mode. Transmissions are assumed to be error-free and omnidirectional. Furthermore, interferences caused by simultaneous transmissions are modeled with one of the interference models introduced in the next chapters and handled via scheduling.

The general system model for a multihop wireless network is a directed acyclic graph (DAG)  $G := (V, E)$ , where  $V$  is the set of all nodes and  $E \subseteq V \times V$  denotes the set of all the edges. We assume that  $G$  is finite, i.e.,  $|E| < \infty$  and  $|V| < \infty$ . Nodes represent wireless terminals that can transmit and/or receive messages and edges represent links over which the messages can be transmitted. A directed edge from node  $i$  to node  $j$  is denoted by  $e = (i, j)$ . In this case,  $i = \text{tail}(e)$  and  $j = \text{head}(e)$ . Furthermore,  $In(i)$  denotes the set of incoming edges to node  $i$  and  $Out(i)$  is the set of outgoing edges of node  $i$ . Let  $S \subset V$  be the set of independent sources, and  $\mathcal{D} \subset V$  be the set of destinations ( $\mathcal{D} \cap S = \phi$ ). In a wireless multicast scenario, each source  $s_i \in S$  sends some messages to all the destination nodes  $d_j \in \mathcal{D}$ . As long as the  $s_i$ 's have independent information, they can be modeled as a single source node. In this case, a virtual source node is added to the network graph that connects to all the actual source nodes [18].

## 2.3 The Min-cut Max-flow Theorem

The min-cut max-flow theorem characterizes the maximum achievable rate of a network that is represented with a graph based on its minimum cut value. We will later explain that this maximum rate is achievable by network coding. The min-cut max-flow theorem is a special case of the duality theorem in linear programming. It was proven by P. Elias, A. Feinstein, and C.E. Shannon in 1956, and independently by L.R. Ford, Jr., and D.R. Fulkerson in the same year [19].

In this theorem we consider a graph  $G = (V, E)$ , and a unicast connection from a source node  $s \in V$  to a destination node  $d \in V$ . The maximum flow from  $s$  to  $d$  is characterized by the minimum cut value between these two nodes.

**Definition 2.1.** A *cut* between  $s$  and  $d$  is a set of graph edges whose removal disconnects

$s$  from  $d$ . A *min-cut* is a cut with the smallest (minimal) value. The value of the cut is the sum of the capacities of the edges in the cut.  $\square$

For unit capacity edges, the value of a cut is equal to the number of edges in the cut, and it is sometimes referred to as the size of the cut. The min-cut can be considered as the bottleneck for the information transmission between  $s$  and  $d$ . In fact, the min-cut max-flow theorem discussed below states that the maximum information rate from  $s$  to  $d$  is equal to the min-cut value.

**Theorem 2.1.** If the min-cut between  $s$  and  $d$  in a graph  $G = (V, E)$  with unit capacity edges is equal to  $h$ , then the information can be sent from  $s$  to  $d$  at a maximum rate of  $h$ . Equivalently, there exist  $h$  edge-disjoint paths between  $s$  and  $d$  [15].  $\square$

Theorem 2.1 relates the maximum flow of a unicast transmission to the minimum cut value of the graph representing the communication network. Paper [20] presents another theorem called the “main theorem of network coding” that proves there exists a network coding scheme over a large enough finite field that delivers the rate from the max-flow, min-cut theorem.

The main theorem of network coding can be proved for the multicast case by extending the preceding results for the unicast case and the following theorem and corollary are a result of that for the multicast case.

**Theorem 2.2.** Consider an acyclic delay-free multicast problem with  $h$  source processes originating at a source node  $s$  and demanded by a set  $\mathcal{D}$  of destination nodes. There exists a solution if and only if for each destination node  $d \in \mathcal{D}$  there exists a flow of rate  $h$  between  $s$  and  $d$  [21].  $\square$

The following corollary is a direct result of Theorem 3.2.

**Corollary 2.1.** The maximum multicast rate is the minimum, over all destination nodes, of the minimum cut between the source node and each destination node.  $\square$

## 2.4 Related Work

In [3], [4] and [5], it is shown that the general problem of interference-free scheduling for multi-hop wireless networks is NP-hard. The reason is that the problem of interference-free scheduling of a wireless network is equivalent to finding maximal independent sets of a graph that represents the wireless network. In [2] it is shown that from a practical point of view, network coding does not necessarily result in an improvement of throughput if network coding and MAC scheduling are designed independently. The result is compared to a joint design of network coding and scheduling. References [6] and [7] were the first to propose a cross-layer approach to minimize energy for multiple multicast scenarios in ad hoc networks. They used artificial nodes to model omnidirectional transmissions in wireless networks. In [6], [7], the energy minimization problem is formulated based on *elementary capacity graphs* (ECG). Transmissions in each ECG do not interfere with each other and timesharing takes place among different multicast sessions. However, the timesharing in these works is not defined for link flows, so although the optimization problem proposed in [6], [7] results in optimal values of flows and time shares for multiple multicast scenarios, the assignment of flows to time shares is not clear. It means the optimization results can not be directly used to schedule the flows. Also, the work of [8] maximizes the throughput for single multicast scenarios and similar to [6], the assignment of the flows to the timeshares is not specified. Our formulation aims at specifying the membership of flows to timeshares. This assignment is important in the code design. In [11], scheduling the realizations is based on a conflict graph approach. Then the constraints of the flow optimization problem are built based on the non-interfering transmissions from the conflict graph, but the optimization problem to maximize the throughput does not consider time sharing variables for each transmission and the time shares are implicitly assumed to be equal. Finally, in [9] and [10], the joint design of MAC scheduling and network coding is formulated as a non-linear optimization problem. The formulation however fails to deliver correct code design if the scheduling timeshares are not all equal to each other. In general, most of the past studies only focus on one or part of the three steps mentioned in Chapter 1 and the solutions either do not cover a proper

code design that is practical for wireless networks, or the designed code does not satisfy the broadcast property of the wireless networks. In our work we focus on formulating a linear optimization problem that can be applied to different objective functions as well as specifying a code design approach that can model the broadcast property of the wireless transmissions as well as properly handle the unequal scheduling timeshares.

Regarding the scheduling problem, the previous works are scheduling based on building the *elementary capacity graphs* in [6], [7], building the scheduling sets based on the conflict graph approach [11], or building the realizations based on the heuristic presented in [9], [10]. Since the last two references introduce a complete solution for joint scheduling and network coding, we used their heuristic as a starting point before formulating our linear joint optimization problem. The mentioned heuristic builds realizations based on the following approach: it starts with assigning a random node (excluding the source) of the wireless graph as a receiver and finds a transmitter (if possible) with minimum cost (power) to reach that node. Receivers and transmitters are added to a realization until we can add no more transmitters to the realization without violating the interference-free condition. The same process will start over for the next realization by choosing another random node as receiver. The stopping criterion for this process is when each node (except the source node) is assigned to be a transmitter and receiver at least once in the total set of realizations. The heuristic implicitly chooses each wireless link at most once. Later we modify this heuristic to add realizations with equal capacity to the existing set and our results show that it outperforms the one proposed by [9], [10].

About designing network codes, there are many code design schemes available for wired networks. Authors of [22] present low-complexity network coding schemes in their work. Both deterministic polynomial time and randomized coding algorithms are discussed in [22]. [21] and [23] are two examples of well-known code construction algorithms that are highly complex. The algorithms presented in [22] are less complex and are hierarchical. Random network coding presented in [24] is also one of the probabilistic approaches of distributed nature that can accommodate the broadcasting constraints and unequal timeshares. Yet

since the coding scheme is random, there is a failure probability associated with the designed code. Furthermore, each packet has to carry the coding coefficients that relate the coded packet to the source packets. This increases the overhead transmission through the network. For the scheme of [22], however, although the information about such coding coefficients are required at the destinations, they do not need to be transmitted through the network and can be sent directly to the destination nodes by the central control unit. Our focus is on the deterministic and centralized code design of [22] and we modify it to include the broadcast property of wireless transmissions.

Regarding the interference models, there are a number of previous works that compared and evaluated different interference models [25]- [29]. In [28], it is shown that if we use the protocol model only, we may determine throughputs that are not achievable in a real setting due to the neglected interferences in that model. It is also shown that if we modify the interference range of the protocol model appropriately, we can get close to the results from the physical model in a routing solution of multi-hop wireless networks. The authors of [25], [26] introduce three methods of adjusting the interference range of the protocol model considering the SINR. Their adjustments are applied to wireless mesh networks and include: all node adjustment (ANA) method, all interference node adjustment (AINA) method, and nearest interference node adjustment (NINA) method. The adjustments in the interference range of these models are done based on the calculated set of SINRs for all the nodes and trying to satisfy a threshold for them. The work of [27] compares the physical interference model with two other simplified models (one of them is the protocol model) for wireless mesh networks and concludes that the performance of the simplified models are not very close to the exact model and care should be taken before accepting the results of the simplified models. In this work, we compare the results of a joint network coding and scheduling optimization problem for throughput and energy when the interference model is either the physical or the protocol model. We also calculate the achievable result of the protocol model based on the interferences in the environment. Our work applies to wireless multihop networks that use MQAM modulations and therefore have variable transmit power and link

capacities. In comparison with our work, [25], [26] only consider fixed transmission power which contradicts with adjusting the interference ranges with their introduced methods. Furthermore, our work is the first to compare the optimization results for network coding solutions and we also compare the run-time complexities of the mentioned methods.

## 2.5 Research Approach

During this whole thesis we assume that the wireless networks are static, and we have complete knowledge about the nodes and the sources and the destinations of each network. Using a centralized approach, we study the impact of different factors such as changing the interference model or modifying the scheduling heuristic on both the throughput and the minimum energy problems. The goal is to make the wireless models realistic by going above and beyond the fixed rate, fixed transmission power levels commonly used in other works. Furthermore, we want to emphasize that our work does not include the distributed implementation and performance measurement of the proposed solutions. As such, the results presented in the thesis can be considered as “benchmarks”, showing the impact of performing certain changes in formulation, scheduling heuristic, or interference models. For the same reason, the actual network codes being used will not matter as much, as long as the codes adhere to the criteria defined in Chapter 4, so either well-designed codes or random linear network codes could be used.

In the next chapter, we introduce the joint linear scheduling and network coding optimization problem as well as the wireless broadcast model to correctly model wireless transmissions.

## Chapter 3

# Joint MAC Scheduling and Network Coding Optimization Problem

The contributions of this chapter are briefly explained in Section 1.1.1. In general, a joint scheduling and network coding optimization should provide not only the optimal values of the transmission flows, but also the timeshares allocated to each realization and their corresponding link flows. It is also desirable to have a formulation that can accommodate different objective functions, including throughput and energy. In this work, we address all the above issues based on a linear program formulation of the joint optimization problem.

Our work builds mainly on the framework first introduced in [9] and [10]. Within that framework, we formulate the joint optimization problem in a linear form that yields link flows that can be scheduled over scheduling timeshares. We also extend the wireless broadcast model of [6], [7] to tailor it to our optimization problems. This model is used in both solving the optimization problem and designing a network code that preserves the broadcast properties of the wireless network. The network model is previously presented in Section 2.2. In Section 3.1 we introduce our wireless channel model. Section 3.2 discusses the scheduling problem in wireless networks and its complexity. In Section 3.3, we formulate a linear optimization problem to jointly optimize the link flows and scheduling timeshares in a graph corresponding to a wireless network with independent sources. The chapter concludes in Section 3.5.

### 3.1 Wireless Channel Model

For the wireless channel model, to keep things relatively simple for the time being, we consider channels with fixed, unit capacity data rates (1 symbol/timeslot) and fixed transmission ranges with probabilistic propagation [30] and fixed transmission powers. Each node  $i$  has an inner radio range ( $R_i^{in}$ ) and an outer radio range ( $R_i^{out}$ ). If a node is inside the inner range of a transmitting node, it can receive messages. If a node is outside the outer range of a transmitting node, it can not receive any message. Nodes that are between the inner and the outer range may receive messages with a probability that is uniformly decreasing with the distance from the transmitting node. More formally, if we have two nodes  $i$  and  $j$  placed at a distance  $d_{ij}$  from each other and want to determine the existence of a connection from  $i$  to  $j$ , (i.e., if link  $(i, j)$  belongs to the set of network edges  $E$ ), we use the following conditional probabilities based on inner and outer ranges of node  $i$ :

$$P((i, j) \in E | d_{ij} \leq R_i^{in}) = 1, \quad (3.1)$$

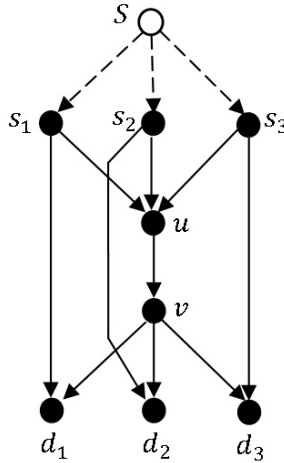
$$P((i, j) \in E | d_{ij} \geq R_i^{out}) = 0, \quad (3.2)$$

$$P((i, j) \in E | R_i^{in} \leq d_{ij} \leq R_i^{out}) = \frac{R_i^{out} - d_{ij}}{R_i^{out} - R_i^{in}}. \quad (3.3)$$

In later chapters, we will refine this channel model to accommodate power-controlled multi-rate transmitters as commonly used in modern communication systems. Throughout this chapter, we apply our formulations on an example network. This network is presented below.

**Example 3.1.** We consider a wireless network with 3 sources and 3 destinations. The connectivity graph associated with this network is shown in Figure 3.1. Given some values of the inner and outer radio ranges for different nodes in the network, this graph is a realization of the probabilistic model described above. We consider a multicast scenario from the three sources to the set of three destinations. The virtual source node  $S$  is also shown in the figure. □





**Figure 3.1:** A wireless network example

In the next section, we introduce the wireless interference model used in this chapter as well as explore the scheduling problem and its complexity.

### 3.2 Wireless Interference Model and Scheduling

The purpose of scheduling wireless transmissions is to share the media among transmitters. Starting from a DAG that represents a wireless network, a scheduling scheme is responsible for assigning the wireless transmissions within a number of subsets so that the members of each subset can be active at the same time without violating each other's transmission. In [5], it is shown that the problem of establishing a throughput optimal scheduling in a radio network is NP-hard. The author defines the capacity region of a packet radio network (PRN) as the set of all *origin-to-destination* (o-d) message rates that are achievable via any arbitrary protocol. Then he shows that the problem of determining whether a given point belongs to the capacity region of a PRN is NP-hard. I.e., there exists no polynomial time algorithm for determining the feasibility of a set of o-d rates unless existing well-known combinatorial problems have polynomial time algorithms [31].

The scheduling problem, however, is proved to be solvable in polynomial time if *secondary* conflicts are acceptable and only *primary* conflicts are avoided [32]. A primary

conflict happens when a node transmits and receives at the same time, or receives more than one transmission destined to it at the same time [33]. A secondary conflict happens when a receiver tuned to a particular transmitter is also within the transmission range of another transmitter whose signal is intended for other receivers. The secondary conflicts are assumed to be permitted when the communication is based on spread spectrum techniques [34]. Therefore, small changes in the problem definition result in huge difference in the complexity. In our work, we define scheduling to be the assignment of transmissions to a number of node sets that avoids both primary and secondary conflicts. In [35], it is shown that finding the optimal link scheduling for multicast scenarios is NP-hard.

In this chapter, we use the *protocol interference model* to model interferences and therefore determine secondary conflicts. In this model, each node  $i$  has a fixed *interference range* around it ( $\hat{R}_i$ ) (potentially larger than the communication ranges introduced in Section 3.1). Transmissions from node  $j$  to  $i$  are successful only if node  $i$  is not inside the interference range of any other actively transmitting node  $k$  [36]. Since scheduling transmissions while avoiding both primary and secondary conflicts is NP-hard, we use the heuristic method of [9], [10]. Note that applying the probabilistic model explained in Section 3.1 to a group of wireless nodes results in a DAG which is the starting point of building the realizations using a heuristic based on the protocol interference model.

Following [9], [10], we call the  $m$ th scheduling set a *network realization* and denote it by  $N_m^f = (V_m^f, E_m^f)$ , where  $V_m^f$  and  $E_m^f$  are the node and the link sets of the realization, respectively. The set of all realizations is denoted by  $N^f = \{N_1^f, \dots, N_{\mathcal{M}}^f\}$ , where  $\mathcal{M}$  is the total number of realizations. A timeshare  $\tau_m$  is assigned to each realization  $N_m^f$ . It shows the duration of time that realization  $N_m^f$  is active during one working cycle of the network. A working cycle of the network is the duration of time it takes until each of the  $\mathcal{M}$  realizations are activated according to their timeshare. The working cycle is repeated periodically with each of the  $\mathcal{M}$  realizations taking turn to be activated in the specified

order. The indicator function  $\mathbf{I}_{E_m^f} : E \rightarrow \{0, 1\}$  is defined by

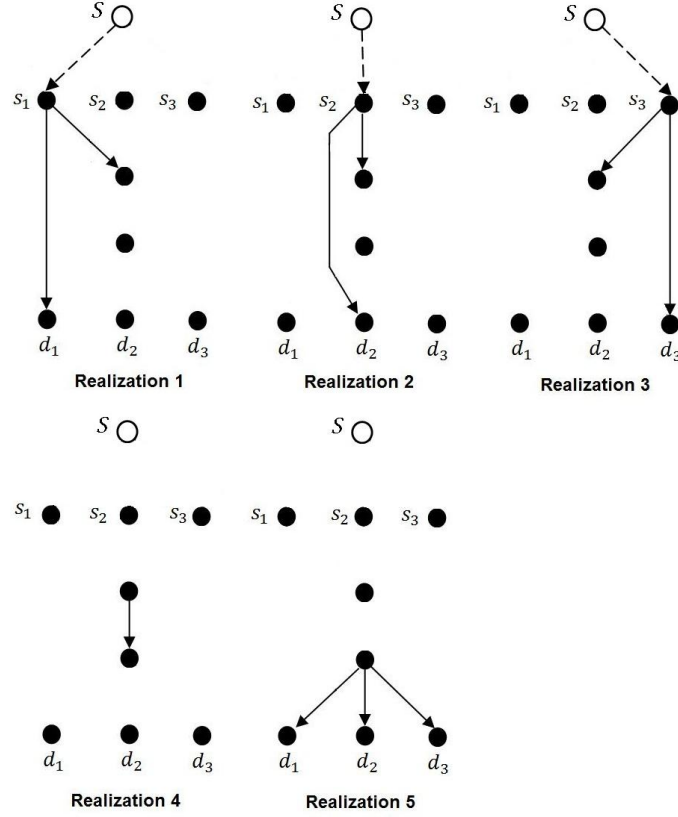
$$\mathbf{I}_{E_m^f}((i, j)) = \begin{cases} 1 & (i, j) \in E_m^f \\ 0 & (i, j) \notin E_m^f \end{cases}$$

This function shows the membership of link  $(i, j)$  in realization  $N_m^f$ . In this paper, we use the heuristic approach of [9], [10] to generate a sufficient set of realizations  $N^f$ . This set will be used as the starting point of our joint scheduling and network coding optimization.

**Example 3.2.** For the network graph of Figure 3.1, we can build the set of realizations according to the above mentioned heuristic approach. This heuristic schedules the transmissions into realizations such that there exist at most one active transmitter closer than its interference range for every receiving node. For example, no more than one of the three source nodes of Figure 3.1 can be active at the same time because they cause interference on node  $u$ . Figure 3.2 shows one possible set of network realizations.  $\square$

### 3.3 Joint Linear Optimization Problem with Schedule-Specific Flows

The characteristics of wireless networks in general require time-varying network flows. Wireless nodes should be scheduled in different timeshares to deal with the interferences by other transmitters. As a result, network coding coefficients of wireless nodes are a function of their scheduling timeshares. In the following, we formulate the joint optimization of network flows and scheduling timeshares as a linear optimization problem. We first construct a hypothetical wired network  $N^g = (V^g, E^g)$ , based on the superposition of the set of  $\mathcal{M}$  network realizations  $N^f$ , where  $V^g = \cup_{m=1}^{\mathcal{M}} V_m^f$  and  $E^g = \cup_{m=1}^{\mathcal{M}} E_m^f$ . The link capacity of an edge  $e = (i, j) \in E^g$  is equal to  $c_{i,j} = \sum_{m=1}^{\mathcal{M}} \tau_m \mathbf{I}_{E_m^f}((i, j))$ . In a single multicast scenario with a source node  $s$  and  $\mathcal{D}$  destination nodes, we define  $f_{i,j}^{(m)}(d)$  as the portion of the flow on link  $(i, j) \in E^g$  destined to node  $d \in \mathcal{D}$  and  $f_{i,j}^{(m)}$  as the total flow on that link, both



**Figure 3.2:** Network realizations for the network of Figure 3.1.

during the activation of the  $m$ th realization with timeshare  $\tau_m$ . Then, we have the following constraints based on the wired graph representation:

$$0 \leq f_{i,j}^{(m)}(d) \leq f_{i,j}^{(m)}, \forall m \in \{1, \dots, \mathcal{M}\}, \forall (i, j) \in E^g, \forall d \in \mathcal{D}, \quad (3.4)$$

capacity constraint:

$$0 \leq f_{i,j}^{(m)} \leq \tau_m \mathbf{I}_{E_m^f}((i, j)), \forall m \in \{1, \dots, \mathcal{M}\}, \forall (i, j) \in E^g, \quad (3.5)$$

flow conservation constraint:

$$\sum_{m=1}^{\mathcal{M}} \left( \sum_{j:(i,j) \in E_m^f} f_{i,j}^{(m)}(d) - \sum_{j:(j,i) \in E_m^f} f_{j,i}^{(m)}(d) \right) = \sigma_i, \forall i \in V^g, \forall d \in \mathcal{D}, \quad (3.6)$$

where  $\sigma_i = r$  if  $i = s$ ,  $-r$  if  $i = d$ , or 0, otherwise. Assuming a normalized working cycle for the network operation, we also have

$$\sum_{m=1}^{\mathcal{M}} \tau_m = 1, \quad (3.7)$$

$$0 \leq \tau_m \leq 1. \quad (3.8)$$

Therefore, an optimization problem to maximize the multicast throughput,  $r$ , of a wireless network is formulated as:

$$\max r \quad (3.9)$$

s.t. Constraints (3.4)-(3.8).

This optimization problem is in fact, optimizing the throughput for a wired network with unknown link capacities. Note that here we only have a constraint on link capacities. The optimization solution is the optimal value of  $r$  and the optimal values of  $f_{i,j}^{(m)}(d)$ ,  $f_{i,j}^{(m)}$ , and  $\tau_m$ , which result in the optimal throughput. The formulation that we introduced here has specific flow variables for different scheduling timeshares. This allows us to know how the link flows should be assigned during different timeshares. However, before we can apply the formulated optimization problem to a wireless network, we need to incorporate the broadcast property of the wireless node in the optimization problem.

The wireless broadcast model introduced here is an extension of the model in [6]. Assume that node  $i$  of a wireless network broadcasts over a set of links  $E_{m,i}^b$  during the timeshare  $\tau_m$ . We indicate the set of nodes that receive the broadcast messages by  $V_{m,i}^b$ . Since node  $i$  broadcasts during  $\tau_m$ , the message combinations delivered to  $V_{m,i}^b$  should be the same, and the rate of transmission from node  $i$  should be upperbounded by  $\tau_m$ .

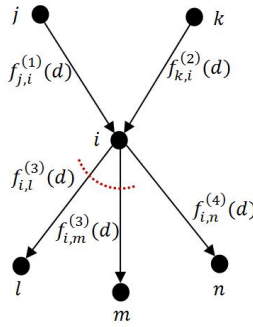
Node  $i$  is a broadcast node of the interference free set  $N_m^f$  if the cardinality of the set  $\{j | (i, j) \in E_m^f\}$  is larger than 1. For a broadcast node  $i$ , we add a node  $\hat{i}$  and the following set of links to our graph:  $(i, \hat{i})$  and  $\{(i, v) | v \in V_{m,i}^b\}$ . We refer to node  $\hat{i}$  and the added links as being *virtual*. This modification is performed on broadcast edges of every realization  $N_m^f$ . We set the capacity of those links equal to  $\tau_m$ . The expanded realizations are denoted by  $N_m^{ex} = (V_m^{ex}, E_m^{ex})$  and the superposition of all the expanded realizations is called an *expanded graph*,  $N^{ex} = (V^{ex}, E^{ex})$ . Here,  $V^{ex} = \cup_{m=1}^{\mathcal{M}} V_m^{ex}$  and  $E^{ex} = \cup_{m=1}^{\mathcal{M}} E_m^{ex}$ . We also

define an indicator function,  $\mathbf{I}_{E_m^{ex}}((i, j))$ , for  $N_m^{ex}$  according to the following relations. For each link  $(i, j) \in E_m^{ex}$ ,

$$\mathbf{I}_{E_m^{ex}}((i, j)) = \begin{cases} 1 & \text{if } (i, j) \in E_m^f, \\ & \text{or } (i, j) \notin E_m^f \ \& \ \exists k | (i, k) \in E_m^f \ \& \ (j, k) \in E^{ex}, \\ & \text{or } (i, j) \notin E_m^f \ \& \ \exists k | (k, j) \in E_m^f \ \& \ (k, i) \in E^{ex}. \\ 0 & \text{otherwise.} \end{cases}$$

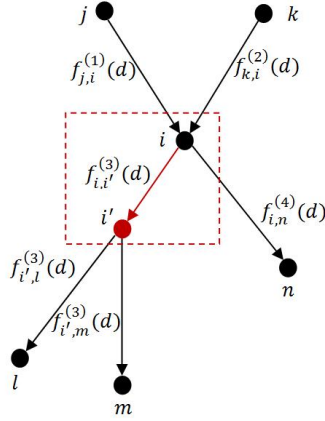
Since the expanded graph preserves the broadcast property of wireless transmissions, we formulate the optimization problem of Section 3.3 for  $N^{ex}$  instead of  $N^g$ . As a result we can use the same formulation without introducing new constraints to preserve the broadcast property. Therefore, the steps towards formulating an optimization problem for a wireless network graph  $G$  is as follows: build the set of realizations  $N^f$ , expand the realizations for broadcast transmissions to obtain  $N^{ex}$  and then formulate the optimization problem of Section 3.3 for  $N^{ex}$ . Graph  $N^g$ , in this case, can be thought of as a condensed version of  $N^{ex}$ , where the virtual node and edges are retracted back to the original broadcast node and its outgoing edges. Either  $N^g$  or  $N^{ex}$  can be used to design network codes.

**Example 3.3.** Figure 3.3 shows a broadcast node  $i$  and its set of broadcast edges and Figure 3.4 shows its corresponding broadcast model. □



**Figure 3.3:** Wireless node  $i$  broadcasts to nodes  $l$  and  $m$  during the 3rd timeshare

**Lemma 3.1.** The optimal value of a flow on a virtual edge (in  $N^{ex}$ ) is the same as the



**Figure 3.4:** Wireless broadcast model of node  $i$

optimal value of the broadcast flow on the corresponding edges in  $N^g$ .  $\square$

**Proof.** The proof is not provided here as it is similar to that of Theorem 3.1 in [6]. Lemma 1 however, generalizes the result of Theorem 1 of [6] to the case where each wireless node may have up to a maximum of  $\mathcal{M}$  virtual edges.  $\square$

### 3.4 Optimization Problem for Minimizing the Energy

The same general approach can be used to formulate an optimization problem for minimum energy, which is also of importance in wireless networks, where devices are frequently battery-operated. Small modifications are required however. As described in Section 3.1, our network model of choice has nodes with fixed transmission powers. Transmission power from node  $i$  is denoted by  $P_i$ . Nodes can be active during different time fractions. The summation of all the transmission powers for transmitting nodes that belong to the  $m$ th realization ( $V_m^{f,T}$ ) is:  $P(m) = \sum_{i \in V_m^{f,T}} P_i$ . Therefore the energy consumption during time fraction  $\tau_m$  is equal to:  $\tau_m P(m)$  and the objective function of the optimization problem of Section 3.3 changes to

$$\min \sum_{m=1}^M \tau_m P(m). \quad (3.10)$$

The above objective function results in the minimum energy for a normalized working cycle of a multicast network. Note that the multicast rate,  $r$ , has to be fixed in (3.6). As we explained in Section 3.3, the optimization solution is the optimal value of the objective function and the optimal values of  $f_{i,j}^{(m)}(d)$ ,  $f_{i,j}^{(m)}$ , and  $\tau_m$ . Now if the optimal value of the total flow on all links  $(i, j)$  in realization  $N_m^f$  that originate from node  $i \in V_m^{f,T}$  is zero, then we need to exclude node  $i$  from the summation  $P(m) = \sum_{i \in V_m^{f,T}} P_i$ . This lowers the energy consumed by the realization during  $\tau_m$ . To incorporate this in the optimization formulation,  $P(m)$  should be replaced by  $\sum_{i \in V_m^{f,T}} P_i \mathcal{U}(\sum_{(i,j) \in E_m^f} f_{i,j}^{(m)})$ , where  $\mathcal{U}(x)$  is the unit step function defined by  $\mathcal{U}(x) = 1$ , if  $x > 0$ , and  $= 0$ , if  $x \leq 0$ . This will make the objective function non-linear. To maintain the linearity of the optimization problem, we resort to an iterative approach, where in each iteration  $l$ , the total transmit power in realization  $m$ ,  $P^l(m)$ , is calculated based on the active links (those with  $f_{i,j}^{(m)} > 0$ ) in iteration  $l - 1$ . In the first iteration, it is assumed that all the links in  $E_m^f$  are active. If we show the total and fractional flows on link  $(i, j) \in E_m^f$  in the  $l$ -th iteration with  $f_{i,j}^{(m),l}$  and  $f_{i,j}^{(m),l}(d)$  respectively, the iterative version of the energy minimization problem is as follows:

$$\begin{aligned} \min \quad & \sum_{m=1}^{\mathcal{M}} \tau_m P^l(m) & (3.11) \\ \text{subject to} \quad & \\ & 0 \leq f_{i,j}^{(m),l}(d) \leq f_{i,j}^{(m),l}, \forall m \in \{1, \dots, \mathcal{M}\}, \forall (i, j) \in E^g, \forall d \in \mathcal{D}, \\ & 0 \leq f_{i,j}^{(m),l} \leq \tau_m \mathbf{1}_{E_m^f}((i, j)), \forall m \in \{1, \dots, \mathcal{M}\}, \forall (i, j) \in E^g, \\ & \sum_{m=1}^{\mathcal{M}} \left( \sum_{j:(i,j) \in E_m^f} f_{i,j}^{(m),l}(d) - \sum_{j:(j,i) \in E_m^f} f_{j,i}^{(m),l}(d) \right) = \sigma_i, \forall i \in V^g, \forall d \in \mathcal{D}, \\ & \sum_{m=1}^{\mathcal{M}} \tau_m = 1, 0 \leq \tau_m \leq 1, \end{aligned}$$

where  $\sigma_i = r$  if  $i = s$ ,  $-r$  if  $i = d$ , or 0, otherwise. Also we have,



$$P^1(m) = \sum_{i:\exists(i,j)\in E_m^f} P_i,$$

$$P^l(m) = \sum_{i:\exists(i,j)\in E_m^f, f_{i,j}^{(m),l-1} > 0} P_i.$$

In each iteration, the optimization problem (3.11) is solved and the iterations will continue until convergence, i.e., the solution does not change in two successive iterations.

### 3.5 Conclusions

In this chapter we examined the problem of joint scheduling and network coding in wireless networks. Designing a network coding and scheduling solution for wireless networks involves three consecutive steps: scheduling, optimization, and coding. Here, we formulated a linear optimization problem that results in schedule-specific link flows that are important for coding purposes. Our formulation can be easily applied to different objective functions such as flow and energy.

## Chapter 4

# Unequal Scheduling Timeshares and Requirements of Constructing Network Codes for Wireless Networks

We briefly presented the contributions of this chapter in Section 1.1.2. When we determine the maximum throughput or the minimum energy for an equivalent wired graph, we have the sufficient information to design the network codes from the optimization problem. However, for the approach that we follow, these codes are being designed in the context of a wired network and need to be transferred back to the wireless graph. This transfer has to be proper to deal with unequal scheduling timeshares and broadcast links. Unequal timeshares result in parallel edges in the wired graph and broadcast links are required to carry identical coding information. In this chapter we first show through a motivating example in Section 4.1 that if scheduling timeshares are not all equal to each other then additional measures should be advised in designing network codes in order to achieve the same performance predicted by the optimization problem. Section 4.2 is devoted to code construction algorithms that preserve the broadcast property in wireless networks. We investigate the statistics of unequal scheduling timeshares and the effect of considering them in designing network codes in the next two sections. Section 4.3 provides simulation results for randomly generated networks to maximize the network throughput and Section 4.4 presents the results for minimizing the energy, given a (feasible) transmission rate. In particular, our

statistics show a throughput improvement of about 35% in maximum flow problems, and energy savings between 13% and 30% depending on the network size, in minimum energy problems. The chapter concludes in Section 4.5.

## 4.1 Review of Wireless Network Code Construction

In [23], Koetter & Medard proposed an algebraic approach to construct linear network codes that achieve the capacity of the network. Their formulation is used for wired networks because the network coefficients they introduced are all link-based. It is assumed that there are a total of  $\mu(v)$  discrete and independent random processes at each node  $v$  of the network. These random processes are going to be delivered and replicated in the network to be received by a group of nodes. In this way, for any link of the network,  $e=(v,u)$  such that  $u = head(e)$  &  $v = tail(e)$  there exists a random process, called  $Y(e)$  that satisfies

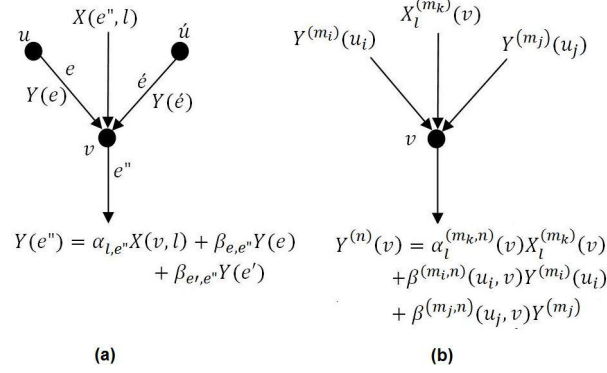
$$Y(e) = \sum_{l=1}^{\mu(v)} \alpha_{e,l} X(v,l) + \sum_{\acute{e}:head(\acute{e})=tail(e)} \beta_{\acute{e},e} Y(\acute{e}) \quad (4.1)$$

Coding coefficients  $\alpha_{e,l}$  and  $\beta_{\acute{e},e}$  are elements of a finite field  $\mathbb{F}_{2^m}$  and  $X(v,l)$  is one of the total  $\mu(v)$  source streams received at  $tail(e)$ . The following relation shows the decoding process at any receiver node  $v$

$$Z(v,j) = \sum_{\acute{e}:head(\acute{e})=v} \epsilon_{\acute{e},j} Y(\acute{e}) \quad (4.2)$$

$\epsilon_{\acute{e},j}$ 's are decoding coefficients and members of  $\mathbb{F}_{2^m}$  and  $max(j)$  is the total number of independent random processes that are going to be decoded at node  $v$ . Figure 4.1-a shows the relationship between  $X$ 's and  $Y$ 's. If we want to employ the same strategy to design wireless network codes, omnidirectional transmission of nodes has to be incorporated into the coding coefficients. In other words, network coding in wireless networks is called to be a node-based design rather than link-based.

Relation 4.3 shows the linear combination of messages at a wireless node:



**Figure 4.1:** Coding coefficients in a) wired networks, b) wireless networks

$$Y^{(n)}(v) = \sum_m \left( \sum_{u: (u, v) \in E_m^f} \beta^{(m, n)}((u, v)) Y^{(m)}(u) + \sum_{j=1}^{\mu(v)} \alpha_j^{(m, n)}(v) X_j^{(m)}(v) \right) \quad (4.3)$$

$Y^{(n)}(v)$  is the encoded flow at node  $v$  and during realization  $N_n^f$ . It uses the coding coefficients  $\beta^{(m, n)}((u, v))$  and  $\alpha_j^{(m, n)}$  that weigh the incoming streams from node  $v$  and the  $j^{\text{th}}$  source stream at node  $u$  respectively. Also, the  $k^{\text{th}}$  output flow is decoded at node  $v \in V$  in network realization  $N_n^f$  as

$$Z_k^{(n)}(v) = \sum_m \sum_{u: (u, v) \in E_m^f} \epsilon_k^{(m, n)}((u, v)) Y^{(m)}(u) \quad (4.4)$$

Figure 4.1-b shows the relation between  $X$ 's and  $Y$ 's. By comparing Figure 4.1-a & 4.1-b, the node-based versus link-based design of network codes for wireless and wired networks becomes quite apparent.

According to *Lemma 1* in [10], the wireless network realizations ( $N^f$ ) and the corresponding hypothetical wired network graph ( $N^g$ ) and also  $N^{ex}$  based on the results in Chapter 3 have the same cut values and therefore have the same maximum flows. Therefore a capacity-achieving coding solution for  $N^f$  achieves the capacity in  $N^g$  as well, of course, after the capacity-achieving code is transferred from  $N^f$  to  $N^g$ .

The claim is correct only if all of the coding characteristics of the wired network are

preserved in the wireless network. The transformation introduced by [10] preserves the omnidirectionality of wireless nodes and assigns the wired coding coefficients to the realizations that occur in different time fractions.

It is instructive to look at the rules of transforming a network coding solution from the wired network ( $N^g$ ) to the wireless one ( $N^f$ ). Relations (4.5-4.7) describe the mappings from  $\alpha_j(\cdot)$ ,  $\beta(\cdot, \cdot)$ , and  $\epsilon_k(\cdot)$  to  $\alpha_j^{(m,n)}(\cdot)$ ,  $\beta_j^{(m,n)}(\cdot)$ , and  $\epsilon_k^{(m,n)}(\cdot)$ . A more detailed description is presented in [10]. Although these transformations are intended to maintain the wireless properties in the resulting wireless coding coefficients, they fail to deliver all properties of the link-based coding coefficients. We will describe this with an example below. The importance and necessity of properly transferring the wired coding solution to the wireless coding solution arises because otherwise there can be no guarantee that the resulting wireless coding solution will achieve the same predicted optimal result as the coding solution for the wired network.

$$\alpha_j^{(m,n)}(\text{tail}(e)) = 1, \text{ if } \alpha_j(e) = 1, e \in E_n^f \quad (4.5)$$

$$\beta_j^{(m,n)}(\acute{e}) = 1, \text{ if } \beta(e, \acute{e}) = 1, \acute{e} \in E_m^f, e \in E_n^f \text{ and } m \neq n \quad (4.6)$$

$$\epsilon_k^{(m,n)}(e) = 1, \text{ if } \epsilon_k(e) = 1, e \in E_m^f, \text{ and } n = \max \left\{ p : \epsilon_k(e) = 1, e \in E_p^f \right\} \quad (4.7)$$

Continuing with the example network from Chapter 3, we next demonstrate that a capacity-achieving linear network code for  $N^g$  does not necessarily achieve the capacity of  $N^f$  if the transformation is performed using only the rules in [10]. In the transformation from the wireless to the wired graph with  $\tau_m$  capacities, if all of the  $\tau_m$ 's are equal, then the links of the wired graph are exactly the same as those of the wireless graph. In this particular case, by choosing a sufficiently large unit of time, all of the link capacities will be equal to 1 and we can build network codes using relations (4.3) & (4.4). However, in case of non-equal time fractions, some modifications are required to maintain all properties of the link-based transmission of  $N^g$  in the wireless graph.

**Example 4.1.** If we solve the optimization problem of Section 3.3 for the example graph in Figure 3.1, using the realizations of Figure 3.2 we have the following results for

the flow maximization problem:

$$\begin{aligned}\tau_1 = \tau_2 = \tau_3 &= 1/7, \tau_4 = \tau_5 = 2/7, \\ r &= 3/7 = 0.42 \text{ symbols/timeslot.}\end{aligned}$$

Achieving the maximum rate (here  $r = 0.42$  symbols/timeslot) requires using realizations 4 and 5 twice as often as the other three realizations. A working period of the wireless network thus includes one timeshare allocated to each of the realizations  $N_1^f$ ,  $N_2^f$ , and  $N_3^f$  and two timeshares to realizations  $N_4^f$  and  $N_5^f$ .  $\square$

A prerequisite for the network code design is that each edge  $e$  of the network must have an integer capacity [37], [22]. By choosing a sufficiently large unit of time, any network can be approximated arbitrarily accurately by a network with edges having integer capacities [37] and any fractional throughput  $r$  can be represented with an integer value  $h$ . After representing all edge capacities by integers, capacities larger than 1 are modeled by parallel edges between two nodes. Now if we build a wired graph based on the capacities (time fractions) that we determined above, Figure 4.2-a shows the resulting graph. A capacity-achieving network coding solution for the wired graph is shown in Figure 4.2-b. For this specific wired graph, there is no capacity-achieving coding scheme such that the coding coefficients for the two outgoing edges of  $u$  are the same. To avoid ambiguity we do not show all of the network coding coefficients in both networks. Wireless coefficients are derived using Relations 4.5-4.7. Now if we focus on two of the edges in  $N^g$  defined by:  $e_7 = e_8 = (u, v)$  we see that based on Relation 4.1, random processes leaving node  $u$  in the wired network are:

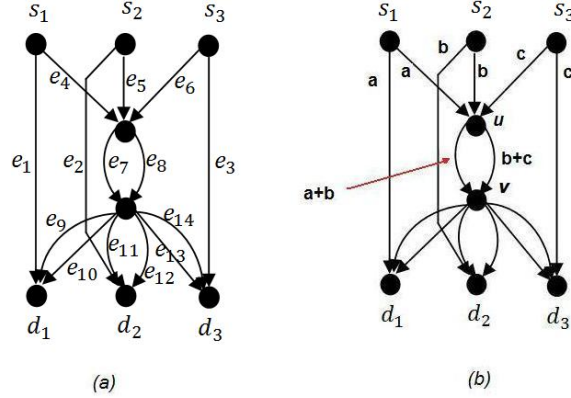
$$\begin{aligned}Y(e_7) &= \beta(e_4, e_7)Y(e_4) + \beta(e_5, e_7)Y(e_5) \\ Y(e_8) &= \beta(e_5, e_8)Y(e_5) + \beta(e_6, e_8)Y(e_6)\end{aligned}$$

According to Relation 4.3, the stream out of node  $u$  in the wireless graph will be the following

$$Y^{(4)}(u) = \beta^{(1,4)}(e_7)Y^{(1)}(s_1) + \beta^{(2,4)}(e_7)Y^{(2)}(s_2) \\ + \beta^{(2,4)}(e_8)Y^{(2)}(s_2) + \beta^{(3,4)}(e_8)Y^{(3)}(s_3)$$

Because wireless coding is a node-based scheme rather than link-based, it does not differentiate between  $e_7$  and  $e_8$ . When the coding coefficients of the wired graph are transferred back to the wireless one, it represents a broadcast model which does not preserve the properties of the wired coding scheme. Knowing that  $\beta^{(2,4)}(e_7) = \beta^{(2,4)}(e_8) = 1$  and the coefficients are in  $F_2$ , the stream out of node  $u$  will be:

$$Y^{(4)}(u) = \beta^{(1,4)}(e_7)Y^{(1)}(s_1) + \beta^{(3,4)}(e_8)Y^{(3)}(s_3)$$



**Figure 4.2:** a) Equivalent wired graph corresponding to the wireless network graph of Fig. 3.1, b) Capacity achieving coding coefficients for the wired network

Accordingly, nodes  $d_1$  and  $d_3$  can receive information from the first and the third sources during 5 time fractions and node  $d_2$  is only able to receive information from the second source in the same time duration, because it is unable to decode the rest of the messages. Consequently, the receiving rate at nodes  $d_1$  and  $d_3$  is  $\frac{2}{5} = 0.4$  packets/timeslot and the receiving rate at node  $d_2$  is  $\frac{1}{5} = 0.2$  packets/timeslot. None of these rates are the maximum achievable rate of the wireless graph. In order for  $N^g$  and  $N^f$  to have the same rate, nodes in  $N^f$  should perform the same encoding and decoding operations as determined for the corresponding wired network.

The approach presented in [10] works for wireless graphs in which all of the time fractions are equal. Otherwise, we have to preserve the following condition while wired coding coefficients are transferred to wireless ones:

- If there are two (or more) edges like  $e_i$  and  $e_j$  in the wired graph such that  $\text{tail}(e_i) = \text{tail}(e_j) = u$ , and they are part of the same network realization, but their coding coefficients are different, they belong to repetitions of a network realization. Therefore, messages transmitted from node  $u$  during the repetitions of a realization should be specified separately.

In our example graph, we can specify messages on edges  $e_7$  and  $e_8$  of the wired graph in the following two wireless linear combinations transmitted from node  $u$  during the two repetitions of the fourth realization:

$$Y_1^{(4)}(u) = \beta^{(1,4)}(e_7)Y^{(1)}(s_1) + \beta^{(2,4)}(e_7)Y^{(2)}(s_2)$$

$$Y_2^{(4)}(u) = \beta^{(2,4)}(e_8)Y^{(2)}(s_2) + \beta^{(3,4)}(e_8)Y^{(3)}(s_3)$$

Figure 4.3 shows a network coding solution with scheduling based on these  $Y_1^{(4)}(u)$  and  $Y_2^{(4)}(u)$  that allows the network to transmit different linear combinations when a realization is repeated. In this table  $s_1 \vec{b}_1 u$  means  $s_1$  transmits  $b_1$  to  $u$ . The total flow of this solution is 0.42 packets/timeslot and is equal to the optimal rate.

|          |                      |                      |                      |                      |
|----------|----------------------|----------------------|----------------------|----------------------|
| Timeslot | 1                    | 2                    | 3                    | 4                    |
| Schedule | $s_1 \vec{b}_1 u$    | $s_2 \vec{b}_2 u$    | $s_3 \vec{b}_3 u$    | $u b_1 \oplus b_2 v$ |
| Timeslot | 5                    | 6                    | 7                    | -                    |
| Schedule | $v(b_1 \oplus b_2)D$ | $u(b_2 \oplus b_3)v$ | $u(b_2 \oplus b_3)D$ | -                    |

**Figure 4.3:** Network coding solution for the wireless network in Figure 3.1

So far, we examined the effect of having non-equal time fractions when we solve a linear programming problem to optimize a criterion of interest in the wireless networks through an example. Theorem 4.1 generalizes the (potential) impact of such non-equal time fractions on network codes to achieve the equivalent performance in both networks.



**Theorem 4.1.** The equivalent wired graph of a wireless graph will have parallel edges with different coding combinations, if and only if solving an LP optimization problem (of interest) for the set of conflict-free realizations of the wireless graph results in non-equal time fractions.

**Proof.** (forward direction) Suppose  $\tau_i$  is the time fraction of nodes in realization  $i$  and  $\tau_i = m\tau$ , where  $m$  is a constant and  $\tau$  is the smallest time fraction of all of the realizations in the network. Now assume for every node in realization  $i$ , there are less than  $m$  outgoing messages with different coding coefficients. Suppose that the maximum number of different coding coefficients out of all nodes in realization  $i$  is  $\acute{m}$  ( $\acute{m} < m$ ). Therefore, all of the nodes in realization  $i$  can transfer their messages during  $\acute{m}\tau$  duration of time. It means  $m\tau$  is not part of the optimal solution.

(backward direction) Suppose there are  $m$  different encoding combinations to be delivered by one wireless node (let's say  $u$ ) in realization  $i$ . Also assume that there exists at least one node  $v$  that is interested in all of the  $m$  different combinations from node  $u$ . These coded streams must belong to realization  $i$ , because all of them are transmitted over link  $(u, v)$ . Therefore, realization  $i$  must happen  $m$  times during one schedule period.  $\square$

## 4.2 Wireless-Aware Network Code Construction

Now that we presented an example that shows how the performance can be affected if the wireless code is not designed properly, we explain wireless-aware code construction here. In this section, our focus is on linear network codes over the finite field  $\mathbb{F}_{2^m}$ , where each element of the field can be represented by a binary vector of length  $m$ . There are a number of coding schemes for wired networks that can be tailored to the wireless networks discussed in this paper. The important issue is to make sure that the coding scheme accommodates the constraints imposed by broadcasting. An attractive low-complexity network coding scheme for wired networks is presented in [22]. Both deterministic polynomial time and randomized coding algorithms are discussed in [22]. Compared to the well-known code constructions of [21], [23], these algorithms are less complex. Random network coding presented in [24]

is a probabilistic approach of a distributed nature that can accommodate the broadcasting constraints and unequal timeshares. Yet since the coding scheme is random, there is a failure probability associated with the designed code. Furthermore, each packet has to carry the coding coefficients that relate the coded packet to the source packets. This increases the overhead transmission through the network. For the scheme of [22], however, although the information about such coding coefficients are required at the destinations, they do not need to be transmitted through the network and can be sent directly to the destination nodes by the central control unit. In the following, we focus on the deterministic and centralized code design of [22] and explain how the design is modified for wireless networks.

In Section 4.1, we explained that we represent the fractional multicast rate  $r$  with a corresponding integer value  $h$ , by scaling up the normalized working cycle in time, resulting in edges with integer capacities. To design a network code that supports a multicast rate of  $h$  from a virtual source  $S$  to all the destinations, the algorithms of [22] first solve a flow problem to find  $h$  edge-disjoint paths between the source and each of the destinations  $d \in \mathcal{D}$ . For each destination  $d$ , then, an  $h \times h$  matrix  $\mathbf{C}_d$  is formed whose columns store the global encoding vectors of  $h$  edges, each belonging to one of the edge-disjoint paths between  $S$  and  $d$ . The matrix  $\mathbf{C}_d$  is initiated by the identity matrix and is updated as the local coding coefficients of edges are designed by traversing the paths from  $S$  to  $d$ . In this process, the coding coefficients of an edge are determined only when the coding coefficients of all its upstream edges are already assigned. The goal is to design the coding coefficients one edge at a time such that all the  $\mathbf{C}_d$  matrices stay full rank until all the edges are processed. This means that the decoding matrices in all  $\mathcal{D}$  destinations will be full-rank, which in turn implies successful decoding of information at rate  $h$  in all destinations.

To adopt the construction of [22] for wireless networks, we can solve the flow problem for either  $N^{ex}$  or  $N^g$  to find the  $h$  edge-disjoint paths between  $S$  and each of the destinations. This is a simple task based on the outcomes of the joint optimization problem. The algorithm of [22] can then be applied by keeping track of which edges are the broadcast edges in the same network realization. The following lemma proves that such edges carry

data destined for different destinations. Yet, due to the broadcast property, such edges must carry the same data and thus must have the same local and global coding vectors. Care should then be taken in assigning the coding vectors of such edges, making sure that the global coding matrices of all destinations which are to use this data for decoding remain full-rank.

**Lemma 4.1.** The destination nodes reachable from different broadcast edges (through the edge-disjoint paths) that originate from a common node and belong to the same realization are different.  $\square$

**Proof (by contradiction):** If two broadcast edges are part of two edge-disjoint paths going to the same destination, they must have linearly independent global coding vectors. This contradicts the requirement that the coding vectors for the two edges are equal due to the broadcast property.  $\square$

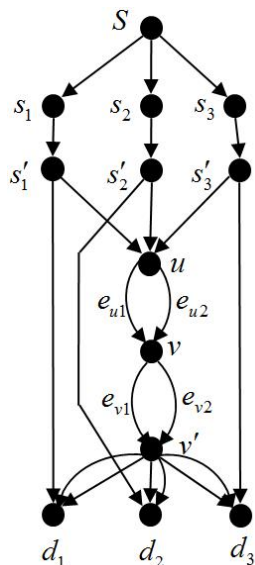
The following example explains the details of designing a network coding solution for the network of Figure 4.2-a.

**Example 4.2.** Consider the wireless network of Figure 4.2-a and the problem of network code design to achieve the maximum throughput of  $\frac{3}{7}$  symbols/timeslot. The graph  $N^{ex}$  corresponding to the optimal throughput is given in Figure 4.4, where the virtual source node  $S$  is added and is connected to the three actual source nodes. Corresponding to the optimal throughput, we have  $h = 3$ , and as the first step of the code design, we need to identify the three edge-disjoint paths between  $S$  and each of the destinations ( $f^{d_i s}$ ). This is an easy task, e.g., for  $d_1$ , we have the following edge-disjoint paths:

$$\begin{aligned} f_1^{d_1} &= \{(S, s_1), (s_1, \acute{s}_1), (\acute{s}_1, d_1)\}, \\ f_2^{d_1} &= \{(S, s_2), (s_2, \acute{s}_2), (\acute{s}_2, u), e_{u1}, e_{v1}, (\acute{v}, d_1)_1\}, \\ f_3^{d_1} &= \{(S, s_3), (s_3, \acute{s}_3), (\acute{s}_3, u), e_{u2}, e_{v2}, (\acute{v}, d_1)_2\}, \end{aligned}$$

where the subscript  $i$  on  $(\acute{v}, d_1)_i$  distinguishes between the two parallel edges between  $\acute{v}$  and  $d_1$ . The network code can now be designed by going through the edges of  $N^{ex}$  one at a time starting from the edges connecting  $S$  to  $s_1$ ,  $s_2$ , and  $s_3$  and moving downstream towards

$d_1$ ,  $d_2$ , and  $d_3$ . At each step, the local coding coefficients for an edge are selected such that the three  $3 \times 3$  global coding matrices associated with the three destinations remain full-rank. In this process, the coding coefficients for a virtual edge  $(s, \acute{s})$  are just copied to the downstream virtual edges  $(\acute{s}, i)$ ,  $i \in V_{m,s}^b$ . A coding solution obtained by this technique is presented in Figure 4.2-b, where the graph  $N^{ex}$  is retracted back to  $N^g$  and the virtual node  $S$  and its edges are also removed. In this figure, the two parallel edges between  $v$  and every  $d_i$ ,  $i = 1, 2, 3$ , carry  $a + b$  and  $b + c$ , respectively.  $\square$



**Figure 4.4:** Graph  $N^{ex}$  of the graph in Figure 4.2-a

### 4.3 Unequal Time Fractions in Wireless Multicast Networks: Maximizing the Throughput

In Example 4.1, we presented an example whose realizations had different timeshares. We also showed in Section 4.1 that a code design solution that enforces equal timeshares would result in a lower throughput. Dealing with unequal timeshares in the code design increases the complexity as described in [38]. Here, using the linear program described in Section 3.3, we explore in depth whether the occurrence of unequal timeshares is rare or common

in randomly generated networks, and how much performance improvement can be expected when unequal timeshares are adequately considered. Our simulations are based on two different objective functions: maximizing the multicast throughput, or minimizing the energy consumption during one working cycle of the network. To simulate a wireless multicast network, we consider random placement of  $N$  uniformly distributed nodes in a two dimensional (2D) rectangular field. Based on the channel model described in Section 3.1, we establish the connections among the nodes. We then choose a random node as a source node and also pick  $n_D$  random nodes as destinations. We decide about the directivity of the edges (and therefore built a DAG) according to all the existing paths from the source to all the destinations. Using the heuristic approach of [9] based on the protocol interference model, we determine a set of interference-free link sets for the network. We then solve the optimization problem described in Section 3.3. As a result, we obtain the optimal values of time fractions and link flows and the optimal value of the cost function. In the following sections, we present our simulation results for different scenarios concerning the maximization of throughput. Simulation results related to the energy minimization are presented in Section 4.4.

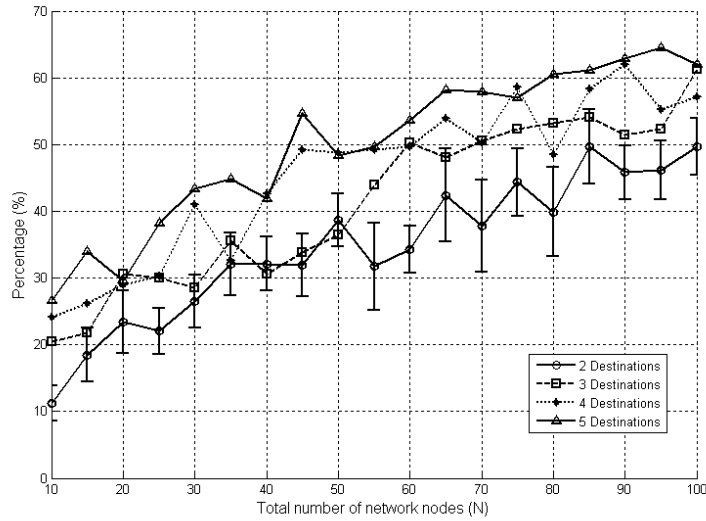
### 4.3.1 Varying Number of Destination Nodes

In the first scenario, we solve the throughput optimization problem for random networks with different number of nodes and  $n_D = \{2, 3, 4, 5\}$  destinations. We assume the average density of neighboring nodes ( $D$ ) in each network is fixed and is equal to 6. The size of the rectangular field changes for different values of  $N$  to keep the density fixed and the ratio of the two sides of the field is  $\frac{1}{3}$ . Appendix A shows how to achieve that for the nodes with probabilistic propagation. All results presented here are the averages over 200 simulated networks for each network size. The same 200 networks are used for different  $n_D$  values to make the comparisons fair. Figure 4.5 shows the percentage of all simulated networks in which achieving the optimal throughput ( $r_{opt}$ ) requires unequal scheduling timeshares. For those networks whose optimal values of timeshares are unequal, we also compute the

throughput that the network can achieve if the timeshares are forced to be equal ( $r_{eq}$ ). We compute  $r_{eq}$  by assigning equal values to all the non-zero timeshares that result from the optimization problem in (3.9). Then we solve a linear program with fixed and equal timeshares to maximize the throughput. Figure 4.6 shows the average of the normalized throughput improvement defined by

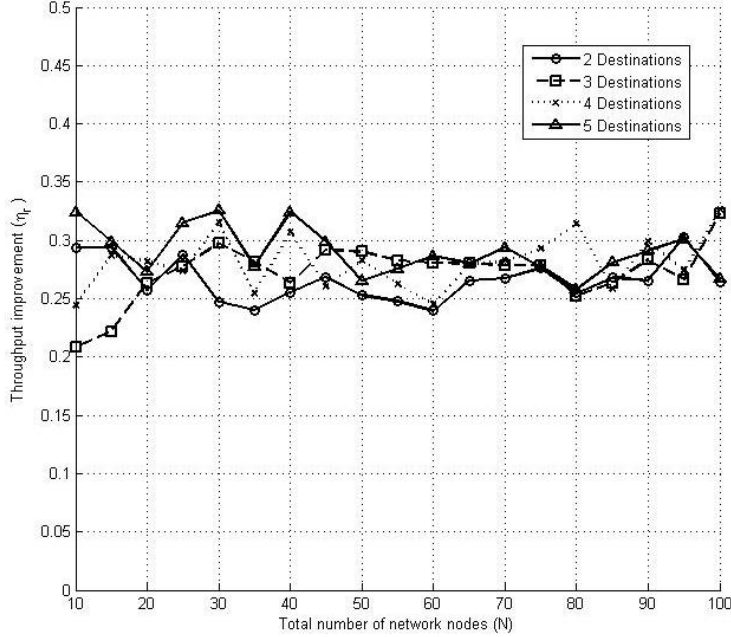
$$\eta_r = \frac{r_{opt} - r_{eq}}{r_{opt}}. \quad (4.8)$$

These figures show the significant performance improvements that can be achieved by correctly dealing with unequal timeshares when optimizing the network throughput.



**Figure 4.5:** Percentage of networks that require unequal timeshares for optimal throughput (the average number of neighboring nodes is  $D = 6$ ).

Figure 4.5 shows that as the network size grows, more networks tend to require unequal timeshares for optimal throughput. The number of variables and constraints in the optimization problem of Section 3.3 grows linearly with the network size. We associate the higher chances of having unequal timeshares with the more complex optimization problems in terms of the number of variables and constraints. The lower percentage of the networks with unequal timeshares for small  $N$  may explain the reason why unequal timeshares have



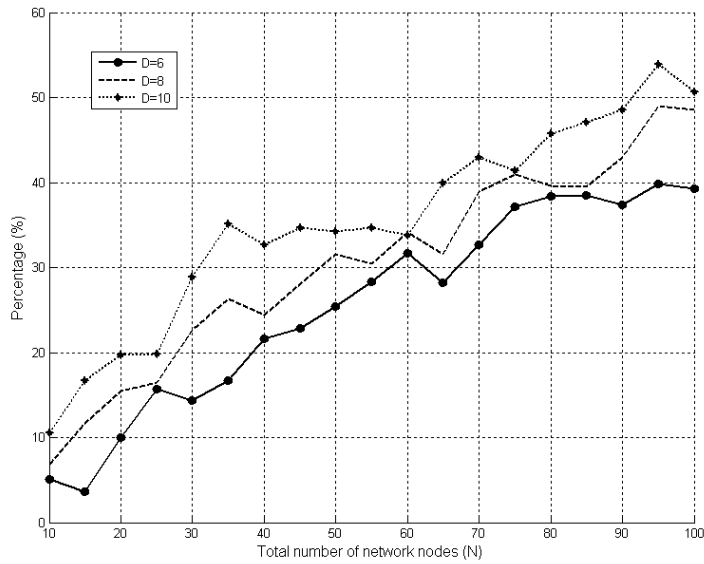
**Figure 4.6:** Throughput improvement using unequal timeshares (the average number of neighboring nodes is  $D = 6$ ).

not been considered in designing network codes before. We also examined the 95% confidence intervals for this graph to justify the noise on the curves. Figure 4.5 shows these intervals for the results with 2 destination nodes. As seen on the figure, the confidence intervals for the consecutive data items overlap, and therefore there is statistical evidence to believe that the rapid changes in the figure are random and not statistically significant.

Furthermore, Figure 4.5 shows that the percentage of networks with unequal timeshares changes slightly with  $n_D$ . This trend can be explained as follows: for a given network, the number of relations in (3.4) and (3.6) is a linear function of  $n_D$ , so the joint optimization problem for networks with larger  $n_D$  is more complex and there is a larger probability of having unequal timeshares in the optimal answer. Figure 4.6 shows that  $\eta_r$  does not change significantly with the number of destinations or the network size, and is in the range of 25% to 33%. We also calculated 95% confidence intervals for the networks with different number of destinations. These confidence intervals were relatively small, i.e., about  $\pm 5\%$  of the reported results.

### 4.3.2 Varying Neighboring Node Densities

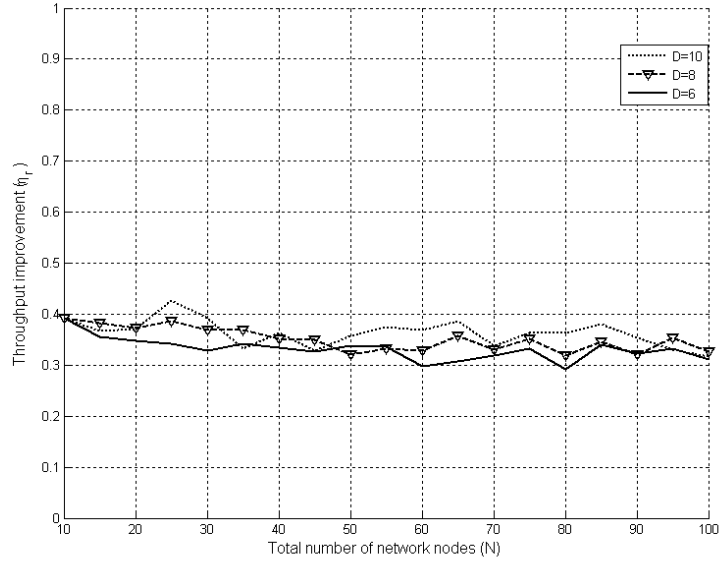
We also explore the impact of network density on the frequency of having unequal time fractions in multicast networks. Here, we consider different neighboring node densities ( $D$ ) and a fixed number of destination nodes  $n_D = 2$ . The densities in our simulations are 6, 8, and 10. The percentage of networks requiring unequal timeshares for optimal throughput is shown in Figure 4.7. The figure shows that the percentage decreases slightly with an increase in  $D$ . We also considered the 95% confidence intervals for this figure. The differences are statistically significant for most of the network sizes. Some of the confidence intervals overlap for consecutive density sizes, but they are non-overlapping if we compare the results of  $D = 6$  and  $D = 10$ .



**Figure 4.7:** Percentage of networks that require unequal timeshares for optimal throughput ( $n_D = 2$ ).

Figure 4.8 shows that  $\eta_r$  does not depend much on  $D$  and stays in the same range for all network sizes. (Again, the 95% confidence intervals are relatively small and show that the average improvement in throughput remains in between about 30% and 40%.)





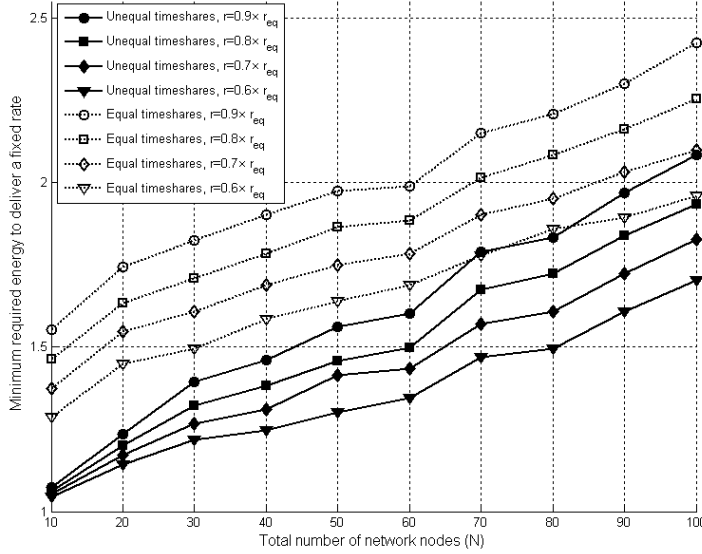
**Figure 4.8:** Throughput improvement using unequal timeshares ( $n_D = 2$ ).

## 4.4 Unequal Time Fractions in Wireless Multicast Networks: Minimizing the Energy Per Working Cycle

### 4.4.1 Minimizing the Energy for a Set of Fixed Rates

When exploring the impact of unequal timeshares on the problem of minimizing the required energy to achieve a given throughput rate  $r$ , we consider multicast networks with  $n_D = 2$  destinations. The average density of neighboring nodes ( $D$ ) in each network is fixed and is equal to 7. For each multicast network, we first solve the flow optimization problem with equal time fractions ( $r_{eq}$ ) to determine the maximum achievable rate for the network with equal timeshares. For each network, we then fix the rate at different fractions of  $r_{eq}$  and compare the network solutions for minimizing the energy with both equal timeshares and unequal timeshares. (The reason that we use  $r_{eq}$  instead of the maximum flow with unequal timeshares,  $r_{opt}$ , is to ensure that the chosen (fixed) rate has a feasible solution for minimizing the energy with equal timeshares as well.) The data rates that we use for the

energy minimization problem are  $\{0.6, 0.7, 0.8, 0.9\} \times r_{eq}$ . Minimum energy with unequal timeshares is represented by  $\mathcal{E}_{opt}$ . For the case where timeshares are forced to be equal, the minimum energy is represented by  $\mathcal{E}_{eq}$ . Figure 4.9 shows the minimum required energy to deliver a fixed rate in normalized working cycles of networks for different rates. The network size varies from  $N = 10$  to  $N = 100$  and the number of simulated networks for each network size is 200. This figure shows that supporting unequal scheduling timeshares drastically lowers the minimum energy for smaller networks and achieves a fairly constant saving in energy for larger networks.



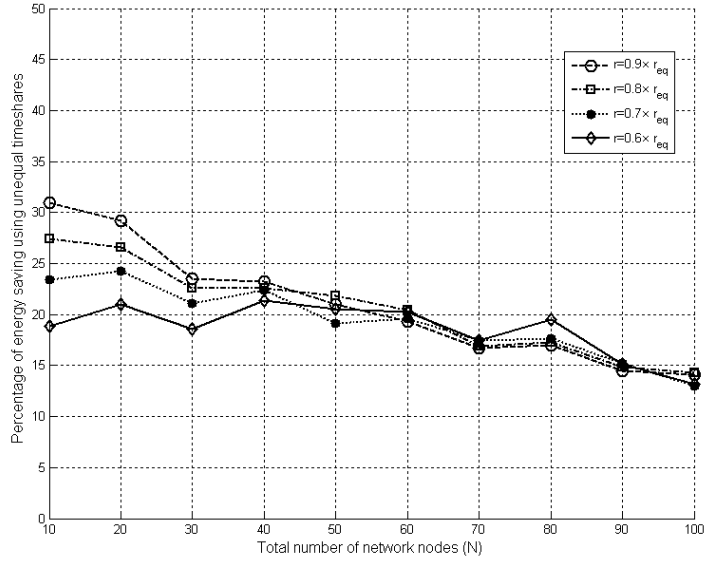
**Figure 4.9:** Minimum energy to deliver a fixed rate ( $n_D = 2$ ,  $D = 7$ ).

We define  $\eta_{\mathcal{E}}$  as the normalized energy saving that results from using unequal timeshares:

$$\eta_{\mathcal{E}} = \frac{\mathcal{E}_{eq} - \mathcal{E}_{opt}}{\mathcal{E}_{eq}}. \quad (4.9)$$

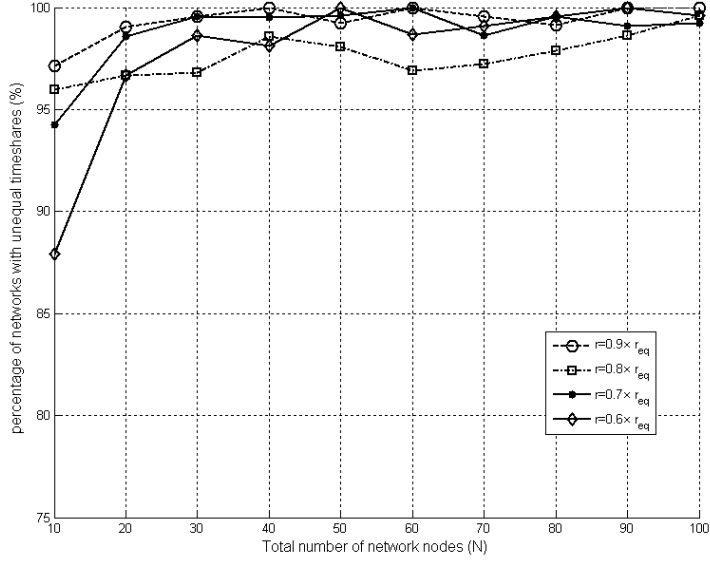
Figure 4.10 shows the value of  $\eta_{\mathcal{E}}$  versus the network size for different rates. The figure shows that the saving in small networks can be up to 31% while the saving for larger networks is at least 13%. We also checked the percentage of all the simulated networks where the minimum energy solution involves unequal timeshares. Figure 4.11 shows that,

starting with network sizes of  $N = 20$ , almost all the simulated networks required unequal timeshares to minimize the energy.



**Figure 4.10:** Energy saving using unequal timeshares ( $n_D = 2$ ,  $D = 7$ ).

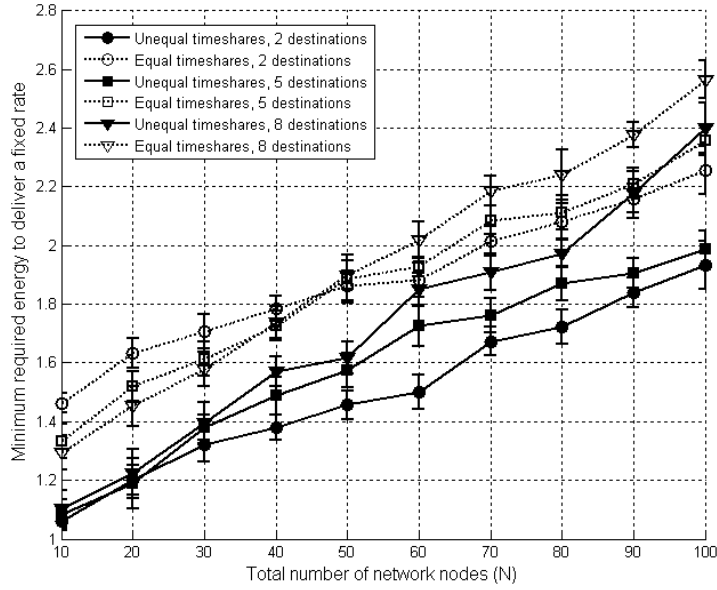
We also performed some simulations to examine the energy consumption of a throughput optimal solution and also the energy consumption of the solutions to the energy minimization problem that achieve portions of the optimal throughput ( $\{0.95, 0.9, .85, .8\} \times r_{opt}$ ). These results follow similar trends as the curves presented in Figure 4.9 for unequal timeshares, and are thus omitted to prevent redundancy.



**Figure 4.11:** Percentage of networks that require unequal timeshares to minimize the energy consumption ( $n_D = 2$ ,  $D = 7$ ).

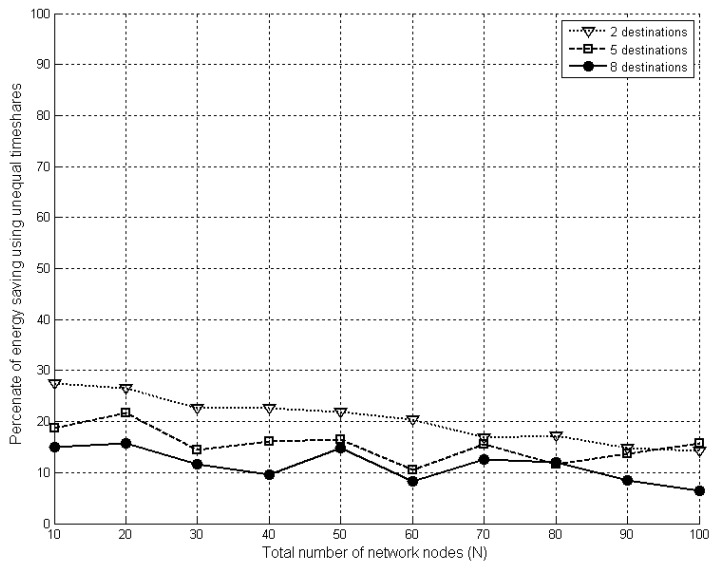
#### 4.4.2 Varying Number of Destination Nodes

In this section, we present simulation results on networks with different number of destination nodes,  $n_D = \{2, 5, 8\}$ . The density of the neighboring nodes is equal to 7. We fix the rate at  $0.8 \times r_{eq}$  and then minimize the energy using the optimization problem of Section 3.4. Figure 4.12 shows the minimum required energy to deliver a fixed rate for different network sizes and different number of destination nodes. The figure shows that while the minimum energy using unequal timeshares is well below the energy when we enforce equal timesharing, it increases with the number of destination nodes. The 95% confidence interval for these plots shows the energy difference between unequal timeshares and equal ones are statistically significant. The confidence intervals are shown in the figure.

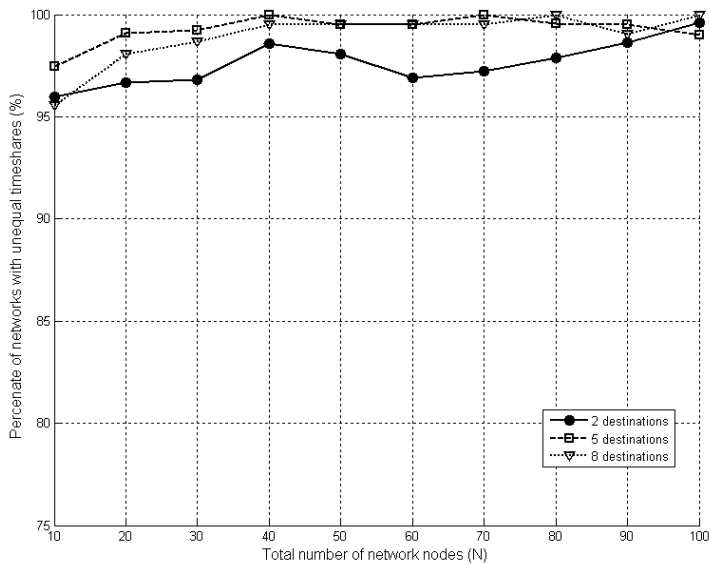


**Figure 4.12:** Minimum energy to deliver a fixed rate ( $n_D = \{2, 5, 8\}$ ,  $D = 7$ ,  $r = 0.8 \times r_{eq}$ ).

Figure 4.13 shows  $\eta_{\mathcal{E}}$  versus the network size for different number of destination nodes. This figure shows that the amount of saving in energy decreases with the increase in the number of destinations for smaller networks, and is rather unaffected by the change in the number of destinations for larger networks. Also, Figure 4.14 shows that almost all the simulated networks require unequal timeshares to minimize the energy. This is particularly the case for larger networks with larger number of destinations.



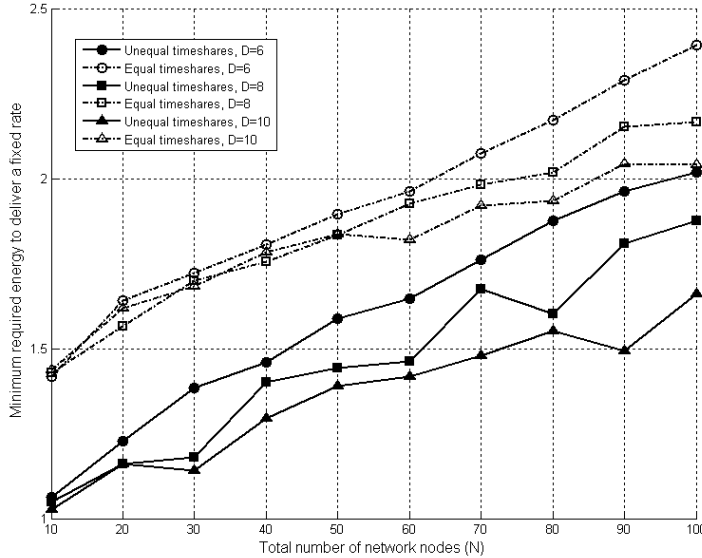
**Figure 4.13:** Energy saving using unequal timeshares ( $n_D = \{2, 5, 8\}$ ,  $D = 7$ ,  $r = 0.8 \times r_{eq}$ ).



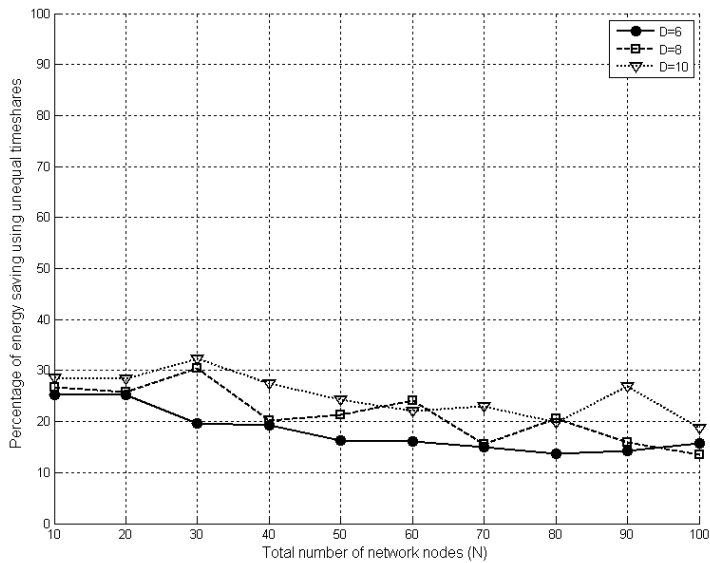
**Figure 4.14:** Percentage of networks that require unequal timeshares to minimize the energy consumption ( $n_D = \{2, 5, 8\}$ ,  $D = 7$ ,  $r = 0.8 \times r_{eq}$ ).

### 4.4.3 Varying Neighboring Node Densities

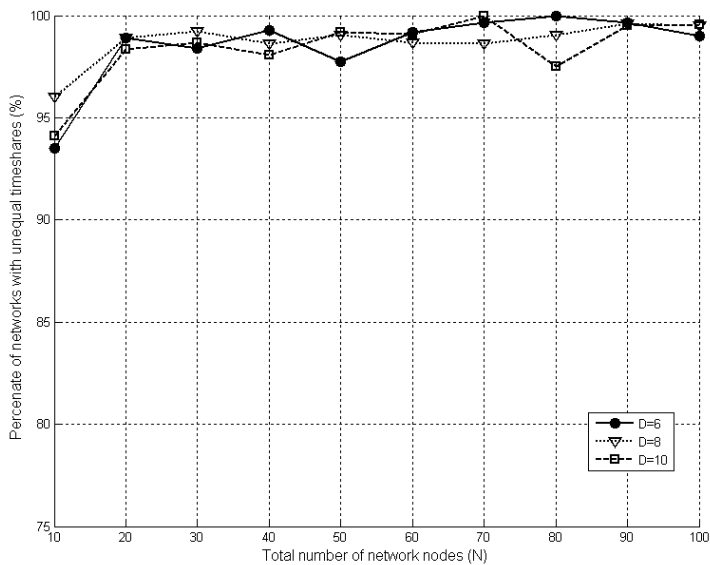
Next, we focus on minimizing the energy for different node densities. In our simulations,  $n_D = 2$  and  $D = \{6, 8, 10\}$ . We fix the rate at  $0.8 \times r_{eq}$  and then minimize the energy using the optimization problem of Section 3.4. Figure 4.15 presents the minimum required energy to deliver a fixed rate for different densities. This figure shows that, as expected, the minimum energy decreases with an increase in the node density. Also, expectedly the results of unequal timesharing is always better than the ones with equal timeshares. We calculated the 95% confidence interval for these plots which show the energy difference between the two groups of equal and unequal timeshares are statistically significant. Figure 4.16 shows  $\eta_{\mathcal{E}}$  versus the network size for different node densities. The energy saving is slightly higher for larger node densities, but remains in the same range for most network sizes. Figure 4.17 shows that almost all the simulated networks, particularly the larger ones, require unequal timeshares to minimize the energy.



**Figure 4.15:** Minimum energy to deliver a fixed rate ( $n_D = 2$ ,  $D = \{6, 8, 10\}$ ,  $r = 0.8 \times r_{eq}$ ).



**Figure 4.16:** Energy saving using unequal timeshares ( $n_D = 2$ ,  $D = \{6, 8, 10\}$ ,  $r = 0.8 \times r_{eq}$ ).



**Figure 4.17:** Percentage of networks that require unequal timeshares to minimize the energy consumption ( $n_D = 2$ ,  $D = \{6, 8, 10\}$ ,  $r = 0.8 \times r_{eq}$ ).



## 4.5 Conclusions

Having formulated the optimization problem on the equivalent wired graph  $G^{ex}$ , the output provides sufficient information to design network codes that achieve the optimization result in a wired network. To apply this solution to the wireless network, the codes have to be transformed from their link-based nature in wired networks to a node-based representation. As shown by an example, such a transfer has to be done carefully to ensure that the predicted performance can indeed be achieved. So we presented modifications to the code construction to successfully deal with unequal timeshares while supporting wireless broadcast as well. Simulation results confirm that correctly incorporating unequal timeshares can improve the multicast throughput by 35% for maximum flow problems and will result in energy savings between 13% to 30% in minimum energy problems.

## Chapter 5

# Performance Comparison of Physical and Protocol Interference Models in Wireless Networks with Joint MAC Scheduling and Network Coding

So far, we studied joint scheduling and network coding in networks where transmitters have fixed transmit power levels, links have unit capacity, and under a specific, simple interference model called the protocol interference model. In real networks, transmitters may have variable transmission powers and their capacities are based on different modulations that they use. So the simple protocol model might not be the best choice for different settings. In this chapter, we focus on the impact of two different interference models when modeling wireless networks. The physical model, also known as the SINR (Signal to Interference and Noise Ratio) model and the protocol model are studied in the context of solving a joint multiple access control (MAC) scheduling and network coding problem in wireless multihop networks. Both throughput and energy optimization problems are considered. In previous chapters, we used the protocol model for the simple scenario of networks with unit capacity links and fixed transmission powers to study unequal timeshares in optimal solutions. In this chapter, we compare the joint scheduling and network coding throughput/energy optimal solutions of networks that employ a set of embedded M-ary Quadrature Amplitude Modulation (MQAM) signal constellations for transmission. Our simulation results show that the performance of the protocol model and the physical model are very close in

throughput maximization problems, while the run-time complexity of the protocol model is significantly lower. Therefore, the protocol model can be used in those problems without losing the performance. In minimum energy problems however, the complexities of the two model are quite close and the performance of the protocol model is not quite comparable to that of the physical model. Therefore, we conclude that the physical model is still the best choice for the energy problem.

## 5.1 Related Work

The physical and the protocol models are the two most popular interference models in wireless networks. Interference at a receiver is caused by transmissions from unintended transmitters. It affects the reception rate at the receiver from the intended transmitter. A transmission is successful according to the physical model if the SINR at the receiving node is higher than a specified threshold. In this model, interferences from other active transmitting nodes are considered as being undesirable and their power is added to that of noise. The protocol model however, decides based only on the distance between a potential pair of transmitter and receiver and also the distance between the already active transmitters and a potential receiver [39]. Briefly, if the distance between a potential transmitter-receiver pair is less than the transmission range of the transmitter and the receiver is beyond the interference range of other active transmitters, we assume successful transmission between the considered pair of nodes. The transmission range of a transmitter depends on its transmission power and the acceptable SNR (signal to noise ratio) threshold at the receiving nodes. The interference range is a parameter unique to the protocol model which shows the distance that transmissions interfere with each other. We define these terms in more detail in Section 5.2. Using either of these interference models for wireless networks has its own advantages and disadvantages. The physical model is an exact model of the interferences in a wireless setting but applying it in a network is computationally complex. The protocol model is an approximate and straightforward model of less complexity but might result in solutions that are not feasible. As a result, there are a number of previous works that discuss

what model is best suited for different wireless networks [25], [26], [27], and how we can correctly apply the protocol model to the wireless networks [28]. We apply our comparison of the interference models on a network coding solution for throughput maximization or energy minimization as explained in Chapter 3.

We explained in Section 2.4 that a number of previous works compared and evaluated different interference models [25]- [29] and we explained the contributions of each work. In this work, we compare the results of a joint network coding and scheduling optimization problem for throughput and energy when the interference model is either the physical or the protocol model. We also calculate the achievable result of the protocol model based on the interferences in the environment. Our work applies to wireless multihop networks that use MQAM modulations and therefore have variable transmit power and link capacities. In comparison with our work, [25], [26] only consider fixed transmission power which contradicts with adjusting the interference ranges with their introduced methods. Furthermore, our work is the first to compare the optimization results for network coding solutions and we also compare the run-time complexities of the mentioned methods.

## 5.2 Channel and Wireless Interference Models

We use the same network model that we introduced in Section 2.2. In the following, the channel and the two interference models are explained in more detail.

### 5.2.1 Channel Model

We assume that the communication channel is lossless and the transceivers of wireless networks employ a set of  $Q$  embedded MQAM signal constellations, with sizes  $M_1 < \dots < M_Q$ . The spectral efficiency of the transmissions over a link  $(i, j)$  of a graph  $G$  is equal to [10]:

$$\frac{c_{ij}}{B_{ij}} = \log_2 M, \quad (5.1)$$

where  $c_{ij}$  and  $B_{ij}$  are the transmission capacity (in bits/second) and the bandwidth of the channel (in Hz), respectively, and  $M$  is the size of the constellation. We further assume that the channel experiences AWGN (Additive White Gaussian Noise) as well. The chosen modulation can be easily replaced with other modulations that have a range of constellation sizes like PSK or MFSK. Each constellation size of the mentioned modulation schemes requires a SINR level to satisfy a BER threshold. Furthermore, we do not use channel coding in our work and using the channel coding would not affect our results. It only changes the choice of SINR thresholds to satisfy a certain BER.

### 5.2.2 Interference Models

According to the physical model, a transmission from node  $i$  to  $j$  is successful if and only if the SINR at node  $j$  is higher than a specified threshold. The SINR is defined as follows, in which the interference from other transmitting nodes are treated as noise.

$$\gamma_{ij} = \frac{a_{ij}P_i}{\sigma^2 + \sum_{k \in X, k \neq i} a_{kj}P_k}. \quad (5.2)$$

In (5.2),  $\sigma^2$  is the power of the AWGN at node  $j$ ,  $P_i$  is the transmit power at node  $i$ ,  $a_{ij}$  is the power gain (less than one) of the link  $(i, j)$ , and  $X$  is the set of active transmitters in the network. The link gain  $a_{ij}$  represents the attenuation due to the path loss. The general formula for path loss for transmitting from node  $i$  to  $j$  is equal to [40]:

$$a_{ij} = K \left( \frac{d_0}{d_{ij}} \right)^\alpha, \quad (5.3)$$

where  $K = \left( \frac{\lambda}{4\pi d_0} \right)^2$ ,  $d_0$  is a reference distance for the transmitting antenna's far field,  $\alpha$  is the pathloss exponent and varies for different environments,  $d_{ij}$  is the distance between nodes  $i$  and  $j$ , and  $\lambda$  is the wavelength of the transmitted signal. For free space propagation, with no obstacles,  $\alpha = 2$ , the value we will use in the remainder of this thesis. The information received at each node  $j$  over a link  $(i, j)$  is acceptable only if it satisfies a target bit error

rate  $p_{b,ij}$  for MQAM. Corresponding to  $p_{b,ij}$  and the constellation size  $M$ , node  $j$  has a target SINR,  $\gamma_{t,ij}$ , approximated by [41]

$$\gamma_{t,ij} = -\frac{2}{3}(M-1)\ln(5p_{b,ij}), \quad (5.4)$$

with the assumption that the receiver has an ideal coherent detector.

The protocol model on the other hand, is a simplified interference model. A successful transmission under the protocol model happens when a receiving node ( $j$ ), falls inside the transmission range of an intended transmitter ( $i$ ), and is outside the interference range of other active transmitters [39]. The transmission range of node  $i$  (i.e.  $R_i$ ), which is a radio range that transmissions of a certain capacity from node  $i$  can reach, is set based on an SNR threshold and can be appropriately set based on the thresholds in the physical model so that the two models become comparable [28]. The interference range of node  $i$  (i.e.  $\hat{R}_i$ ) however is unique to the protocol model and  $\hat{R}_i \geq R_i$ . If a receiving node  $j$  falls in the interference range of an active transmitter  $i$ , the receiver will not be able to receive another transmitter's signal because of the interference caused by  $i$ . The transmission and the interference ranges of the protocol model are fixed parameters. However for a system with a set of MQAM constellation sizes,  $M \in \{M_1, \dots, M_Q\}$ , there is one transmission range per constellation size. In order for the protocol and the physical model to be comparable, we determine the transmission range of each constellation size (defined as  $R^{(q)}$ ) for free space using the following relation:

$$\gamma_q = \left( \frac{\lambda}{\sigma 4\pi R^{(q)}} \right)^2 P_{max}, \quad (5.5)$$

where  $\gamma_q$  is the target SNR of constellation size  $M_q$  and is given by (5.4), and  $P_{max}$  is the maximum transmit power of the nodes. The maximum transmit power and the target BER are assumed to be fixed for all the nodes. To set the interference range, we have to note that the interference is caused by the transmitted signal and is independent of the transmission capacity. Therefore, the interference range is independent of the constellation size. We set

the interference range of node  $i$  as  $\hat{R}_i = (1 + \Delta)R^{(1)}$ ,  $\Delta \geq 0$ . In summary, for a system with the mentioned set of constellation sizes, we have a set of transmission ranges  $\{R^{(1)}, \dots, R^{(Q)}\}$ , and a fixed interference range. As we explained before, the interference range is unique to the protocol model. The closer the interference range is to the transmission range, the less sensitive is the model to possible interferences from other transmitters. Consequently, in throughput maximization problems, there is a higher chance of determining a maximum throughput that is higher than that of the physical model. However that solution might not necessarily be an achievable solution because of all the neglected but existing interferences. In this case, the chosen interference range is overly optimistic about the interferences. On the other hand, if the interference range is a lot larger than the transmission range, it may suppress concurrent transmissions that cause small interference. As a result of being too conservative in this case, we may end up having very low throughput for maximum throughput optimization problems. Overall, different values for the interference range result in different solutions, an issue we will explore in more detail later.

### 5.3 Scheduling for a Network with Variable-Rate, Variable-Power Nodes

As discussed previously, the scheduling problem in wireless networks is NP-hard, and therefore transmissions are scheduled using heuristic approaches. As a result, depending on how the links are scheduled to transmit, the result of the optimization problem (maximizing the throughput or minimizing the energy) would differ. Here we use the basic idea of scheduling transmissions presented in [9], [10] and adapt it to networks with variable-power, variable-rate nodes. We consider that the transceivers in a wireless network are enabled to use one of the  $Q$  constellations with sizes  $M_1, \dots, M_Q$ , resulting in links with different transmission capacities. We first explain the common steps and assumptions for building a set of realizations when the interference model is either the physical model or the protocol model and then continue with explaining the steps unique to each model. We assume the network

nodes are placed in known locations, and that we know the distances between the nodes, and that each node  $i$  has a maximum transmission power  $P_{max}$  ( $P_i \leq P_{max}$ ). We assume that the power level for transmission from each node in each realization is determined and fixed by the scheduling heuristic. We first define a graph for the set of wireless nodes and then apply the scheduling heuristic to that graph. We define the set of potential links,  $(i, j)$ 's, as feasible if the SNR at node  $j$  satisfies

$$\gamma_{ij} = \frac{a_{ij}P_{max}}{\sigma^2} \geq \gamma_1, \quad (5.6)$$

or equivalently,  $d_{ij} \leq \frac{\lambda}{4\pi\sigma} \sqrt{\frac{P_{max}}{\gamma_1}}$ . Since  $M_1 < M_2 < \dots < M_Q$ , based on (5.4), for a fixed target BER,  $\gamma_1 < \gamma_2 < \dots < \gamma_Q$ . Therefore, to find feasible links, it is enough to consider  $\gamma_1$  only. Potential links of a wireless network build a collection of directed links for every transmitter-receiver pair  $(i, j)$ . In order to define a multicast scenario over these potential links, we pick one or more random nodes as source nodes  $s_i$  and also pick  $n_D$  nodes randomly as destinations. Then we consider all the directed cycle-free paths from the source node(s) to the destination nodes and define the superposition of those links as a directed graph. Now we can apply scheduling heuristics to the directed graph and group transmissions into realizations.

### 5.3.1 Scheduling with Physical Interference Model

As explained in Section 5.2.2, a transmission from node  $i$  to  $j$  is successful under the physical interference model if  $\gamma_{ij} \geq \gamma_{t,ij}$ . Since transmission powers can be adjusted, we require a criteria to determine transmission powers when constructing realizations. Minimizing the transmission powers has been suggested in [6], [9]:

$$\begin{aligned} \min \sum_{i \in X} P_i \\ \text{subject to} \end{aligned}$$



$$\left\{ \begin{array}{ll} \gamma_q \leq \gamma_{ij} < \gamma_{q+1}, & \text{if } q < Q, i, j \in V \\ \gamma_{ij} \geq \gamma_Q, & \text{if } q = Q, i, j \in V \end{array} \right. ,$$

$$P_i \leq P_{max}, i \in V, \tag{5.7}$$

where  $X$  is the set of active transmitters and the constellation sizes, the target BER and hence the SINR thresholds and the link capacities are fixed and the variables are transmission powers. The constraints of the optimization problem are considered as follows: if the spectral efficiency of a link is  $\log_2 M_q$  and  $q < Q$ , then its SINR (which is related to the transmitter node's transmit power) should satisfy the constraint:  $\gamma_q \leq \gamma_{ij} < \gamma_{q+1}$  (for  $q = Q$ , we have  $\gamma_{ij} \geq \gamma_Q$ ). The solution of the optimization represents a potential realization. To maximize the number of active transmitters, we incrementally build realizations, starting from a single transmitter-receiver pair. For this link, we randomly assign a transmitter constellation size (i.e. a spectral efficiency) and find the optimal transmission power according to the optimization problem in (5.7). Only the constellation size of the first node is picked randomly as there is no other interfering node to force a specific constellation size to the first transmitter. If we pick the smallest constellation size, this would be a potentially poor choice for both maximizing the throughput and minimizing the energy optimization problem, it forces the link to settle for the lowest capacity, while it could probably transmit at a higher level. If we start from the maximum constellation size, since there is no other assigned transmitter in the realization, the link may end up having a very high capacity, but later nodes are forced to have very low capacities due to the interferences. To add the second transmitter-receiver pair, we pick a new link randomly and solve the optimization problem of (5.7) for the two links. The SINR threshold for the first link (and its transmission rate) remains the same. For the SINR threshold (or equivalently the transmission capacity) of the second link, we start with the highest SINR (/transmission capacity) and check to see if the optimization problem of (5.7) has a solution. If it does not, the two links can not be active at the same time while their SINR constraints are satisfied. Therefore, we decrease the SINR threshold to a lower level until either the optimization problem of (5.7) has a

solution or we run out of choices in terms of SINR thresholds and therefore conclude that the newly considered link can not be added to the realization. Additional links are added to a realization until no further link can be added without violating the SINR constraints of the already selected links. For every randomly picked transmitter-receiver pair, we also check all other broadcast links from the chosen transmitter to its next hop nodes with the same constellation size and add them to the realization if the optimization problem in (5.7) remains feasible.

Note that the optimal transmission power levels of the scheduled links may change when we add a new link to the set of already chosen links due to the additional interference. Once a realization can not be extended further, we build the next realization with a new randomly selected link. The process of building new realizations continues until every feasible link is part of at least one realization.

### 5.3.2 Scheduling with Protocol Interference Model

We explained in Section 5.2.2 that for a system with MQAM, there is a transmission range  $R^{(q)}$  associated with each constellation size  $M_q$  and an interference range  $\acute{R}_i$  for every active transmitter  $i$ . Therefore, a transmission from node  $i$  to  $j$  with capacity  $\log_2 M_q$  under the protocol model is successful only if  $d_{ij} \leq R^{(q)}$  and for all other active transmitters in the same realization (i.e.  $k \in X$ )  $d_{kj} > \acute{R}_k$ . Building realizations based on the protocol model is less complex than that of the physical model because in this case, there is no need to go through an iterative optimization problem. The add/drop decision is merely based on the mentioned transmission and interference ranges.

To build a realization, we randomly pick the first transmitter-receiver pair  $(i, j)$  and assign a random MQAM constellation size (i.e. spectral efficiency) to it. Then we check if  $d_{ij}$  lies within the transmission ranges corresponding to the chosen constellation size. If not, we step down to a constellation with the largest size that is smaller than the current constellation. The reason is that its corresponding transmission range is larger than the previous one and it may include  $d_{ij}$ . We repeat this step (if necessary) until the transmission

range of the chosen constellation size covers  $d_{ij}$ . As discussed in Section 5.4, we construct our DAG with potential links that satisfy the SNR threshold of  $\gamma_1$ , therefore this step always has a solution. The next step is fixing the transmission power. In order for link  $(i, j)$  to have a transmission capacity equal to  $\log_2 M_q$ , node  $i$ 's power needs to be within the following range:

$$\frac{\gamma_q \sigma^2}{a_{ij}} \leq P_i \leq P_{max}. \quad (5.8)$$

We do not set  $P_i$  equal to the lowerbound because, as discussed in [28], the optimal solution with the protocol model may be unachievable and we need to find the achievable result by performing a “feasibility test” to find the real supported rate/energy by the protocol model in Section 5.4. In the next section, we will explain why setting the powers at their lowerbound is not suitable when we check the achievable result with the protocol model. For now, we assume that we set  $P_i = \frac{\gamma_q \sigma^2}{a_{ij}} (1 + \delta)$ , for some  $\delta > 0$  and check to see if the resulting  $P_i$  remains less than or equal to the maximum power. If the resulting  $P_i$  is higher than  $P_{max}$ , then we need to step down to a lower capacity level and try to schedule the link with the lower capacity. If we are already considering the smallest constellation size and the resulting  $P_i$  is higher than  $P_{max}$ , then we drop the link from the realization. For every randomly picked transmitter-receiver pair, the following two conditions should be satisfied in order for the link to be added to a realization. If we can not find any constellation size that satisfy these conditions we drop this particular link.

- Condition 1: A new transmitter should not interfere with already activated receivers
- Condition 2: The receiver should not be inside the interference range of other transmitters.

We also check all other broadcast links from the chosen transmitter to its next hop nodes with the same constellation size and add them to the realization if they satisfy Condition 2 and Condition 3 (presented below):

- Condition 3: The constellation size is the largest such that its transmission range still covers the distance between transmitter and receiver

In this way, additional links are added until no further link can be added. Once a realization can not be extended any further, we build the next realization, starting with a new randomly selected link. The process continues until every feasible link is part of at least one realization.

## 5.4 Joint Scheduling and Network Coding Optimization Problem

Scheduling transmissions assuming the interference model is either the physical model or the protocol model results in different sets of realizations for each model. We assume there are a total of  $\mathcal{M}_{ph}$  realizations built based on the physical model and a total of  $\mathcal{M}_{pl}$  realizations based on the protocol model. We represent the  $m$ -th realization of each model with  $N_m^{ph} = (V_m^{ph}, E_m^{ph})$  and  $N_m^{pl} = (V_m^{pl}, E_m^{pl})$ , where  $V$  and  $E$  represent the node and the link sets of each realization, respectively. The capacity of each link at each realization is determined by the spectral efficiency of the active constellation size at the transmitter of the link. The set of all the realizations for the physical and the protocol models are shown by  $N^{ph} = \{N_1^{ph}, \dots, N_{\mathcal{M}_{ph}}^{ph}\}$  and  $N^{pl} = \{N_1^{pl}, \dots, N_{\mathcal{M}_{pl}}^{pl}\}$ , respectively. We then formulate the joint scheduling and network coding optimization problem to either maximize the throughput or minimize the energy for each of these sets of realizations based on the formulation presented in Section 3.3. According to [28], the result of the optimization problem based on the protocol model may be unachievable due to the ignored interferences. Although the optimization problem based on the protocol model results in a maximum throughput or minimum energy, depending on the objective function, that result might not be realizable when we activate the transmissions with their transmission powers. The transmission powers are fixed when building the realizations. To compare the actual performance of the physical and the protocol model, we need to find the actual feasible solution by the protocol model through a “feasibility test” [28]. In order to do that, we calculate all the neglected interferences at the receiving nodes of every activated link in every realization and make

sure the SINR at the receivers are still larger than the  $\gamma_q$  of the constellation size assigned to that transmission. Note that we previously set transmission powers at every transmitter node at  $P_i = \frac{\gamma_q \sigma^2}{a_{ij}} (1 + \delta)$ , which accounts for SNR only. If the SINR at a receiving node is less than its associated  $\gamma_q$ , we need to lower the transmission capacity (stepping down to  $\gamma_{q-1}$  threshold), while keeping the same power level and re-check the SINR condition. In this way, we adjust the link capacities of each realization based on the actual and existing interferences in the environment. If in a realization there is a receiving node  $j$  that can not satisfy the SINR condition with any of the  $\gamma_q$  thresholds considered for its transmitter (node  $i$ ), then we need to remove node  $j$  and link  $(i, j)$  from the considered realization. If node  $i$  is not transmitting to any other node during the considered realization, then we need to remove  $i$  from the realization (deactivate it). In this case, we need to re-calculate all the SINR values in the realization because removing a transmitter decreases the interference level. This re-calculation is done to re-check the capacity levels that are adjusted for the feasible solution.

This discussion then also explains our choice of initial  $P_i$ 's. We set them equal to the power level sufficient to satisfy the SNR ratio multiplied by  $(1 + \delta)$ . If we fix them to be the exact lower bound in (5.8), then assuming that almost all of the realizations have more than one transmitter node, when we calculate the interferences for the SINR values of the achievable protocol model, we need to change almost all of the transmission capacities because the power levels are marginal. This implies that the slightest amount of interference causes the SINR values at receiving nodes to fall below their  $\gamma_q$  threshold. To prevent this, we fix the powers at values higher than their corresponding lower bound.

In using the joint optimization problem of Section 3.3, we should note that since we use MQAM signal constellations, the link capacities are discrete values and are equal to  $\log_2 M_q$  (in bits/second) for unit bandwidth, assuming that transmitter  $i$  transmits with constellation size  $M_q$  in realization  $N_m^f$ . Also note that we formulate the optimization problem for the expanded graph to preserve the broadcast property of the wireless transmissions.

We modified the optimization problem of Section 3.4 for energy minimization as follows.

First we fix the multicast throughput based on a feasible throughput  $r$  that results from solving the throughput maximization problem. To formulate an energy minimization problem, we assume that node  $i \in V_m^f$  transmits with power  $P_i^{(m)}$ . These transmission powers are fixed when we schedule the wireless transmissions using a scheduling method, which is described in Section 5.3. The energy used by this node to transmit in the realization  $N_m^f$  depends on the duration of time that this node is transmitting with power  $P_i^{(m)}$ . The duration of time depends on both the time fraction that node  $i$  can transmit,  $\tau_m$ , and is also proportional to the ratio of the broadcast flow to the broadcast capacity of the particular node and is thus equal to  $\tau_m P_i^{(m)} \frac{f_i^{(m)}}{C_i^{(m)}}$ , where  $f_i^{(m)}$  and  $C_i^{(m)}$  are the broadcast flow and the broadcast capacity of node  $i$ , respectively. The flow  $f_i^{(m)}$  is equal to flow  $f_{(i,i)}^{(m)}$  in the expanded graph for broadcast nodes and equal to  $f_{ij}^{(m)}$  for nodes that only transmit on one link. The capacity  $C_i^{(m)}$  is determined in Section 5.3 by the scheduling heuristic. If the transmitting node is only active on one link  $(i, j)$  during realization  $N_m^f$  then  $C_i^{(m)} = C_{ij}^{(m)}$ . As we explained in Section 5.3, we pick the same constellation size for broadcast links of a transmitter. Overall, for all the realizations, the objective function for minimizing the energy becomes  $\sum_{m=1}^{\mathcal{M}} \sum_{i \in V_m^{f,(T)}} \tau_m P_i^{(m)} \frac{f_i^{(m)}}{C_i^{(m)}}$ , where  $V_m^{f,(T)}$  denotes the set of active transmitters in realization  $m$ . Note that the objective function also handles situations where the optimal value of the total flow that originates from a certain node  $i$  during the  $m$ -th realization is zero. However, the objective function now has become non-linear. To maintain the linearity of the objective function, we use the iterative approach of Section 3.4. The iterative version of the energy minimization problem for a given multicast rate  $r$  becomes:

$$\begin{aligned}
\min \quad & \sum_{m=1}^{\mathcal{M}} \tau_m P^l(m) & (5.9) \\
\text{subject to} \quad & \\
& \sum_{m=1}^{\mathcal{M}} \tau_m = 1, \\
& 0 \leq \tau_m \leq 1,
\end{aligned}$$

$$\begin{aligned}
0 &\leq f_{i,j}^{(m),l}(d) \leq f_{i,j}^{(m),l}, \forall m \in \{1, \dots, \mathcal{M}\}, \forall (i, j) \in E^g, \forall d \in \mathcal{D}, \\
0 &\leq f_{i,j}^{(m),l} \leq \tau_m c_{ij}^{(m)} \mathbf{I}_{E_m^f}((i, j)), \forall m \in \{1, \dots, \mathcal{M}\}, \forall (i, j) \in E^g, \\
\sum_{m=1}^{\mathcal{M}} \left( \sum_{j:(i,j) \in E_m^f} f_{i,j}^{(m),l}(d) - \sum_{j:(j,i) \in E_m^f} f_{j,i}^{(m),l}(d) \right) &= \sigma_i, \forall i \in V^g, \forall d \in \mathcal{D},
\end{aligned}$$

where  $\sigma_i = r$  if  $i = s$ ,  $-r$  if  $i = d$ , or 0, otherwise. We also have,

$$\begin{aligned}
P^1(m) &= \sum_{i:(i,j) \in E_m^f} P_i^{(m)}, \\
P^l(m) &= \sum_{i:(i,j) \in E_m^f} P_i^{(m)} \frac{f_i^{(m),l-1}}{C_i^{(m),l-1}},
\end{aligned}$$

as part of the optimization problem. The variables  $f_i^{(m),l-1}$  and  $C_i^{(m),l-1}$  denote  $f_i^{(m)}$  and  $C_i^{(m)}$  in iteration  $l - 1$  respectively. In each iteration, the optimization problem (5.9) is solved and the iterations will continue until convergence, i.e., until the solution does not change in two successive iterations. As we explained in Section 4.2, network coding schemes can be designed to achieve the maximum throughput or the minimum energy solution of the above optimization problems. For more details, the reader is referred to [21], [22], and [23].

## 5.5 Simulation Results

As discussed, the interference range in the protocol model is a parameter introduced for this model only and there is no similar parameter in the physical model so that we can find appropriate values of the interference range based on that. The optimal interference range is the one that results in the maximum achievable throughput. Before comparing the overall performance of the physical model and the protocol model, we need to decide which interference range is more suitable for the protocol model. To decide about the interference range, we perform simulations for maximum throughput optimization problem with various interference ranges and compare their results. The rest of our simulations are based on the fixed interference range that we pick as the best choice. Later in this section, we compare the results of maximizing the throughput when the interference model is either the

physical or the protocol model. We also determine the achievable result with the protocol model. To compare the computational complexity of the physical model and the protocol model (with achievable solution), we also consider the CPU times of building(/modifying) the realizations based on either model. We also extend our simulations to the energy minimization problem and compare the performances and the CPU times of the two models for that problem. The absolute times that we present are dependent on the specifications (CPU speed, and amount of RAM) of the computer that is used for simulations, but there is no reason to believe that the ratio of the CPU times would change if a different computer is used for simulations.

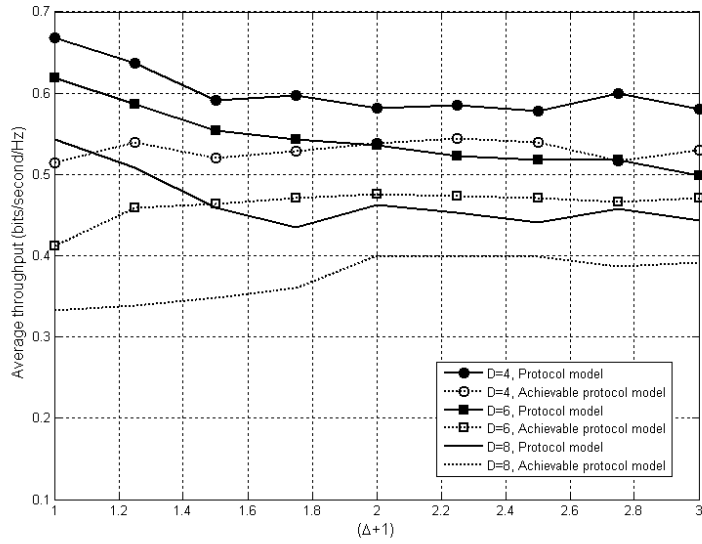
In order to simulate a wireless multicast network, we consider  $N$  nodes are randomly placed in a two-dimensional rectangular field of size  $a\lambda \times b\lambda$ , where  $\lambda$  is the communication wavelength and  $b = 3a$ . We also consider the three values 4, 6 and 8 for the average density of the neighboring nodes ( $D$ ) in our simulations. All the simulations are performed for 100 random networks. The constellation sizes are assumed to be  $M = \{2, 4, 16, 64\}$ , corresponding to BPSK, 4QAM, 16QAM, and 64QAM modulations. The spectral efficiencies associated with these constellation sizes are 1, 2, 4, and 6, respectively. Assuming a normalized bandwidth, these values are equal to link capacities as well (in bits/second). Each node  $i$  has a maximum transmission power  $P_{max}$  ( $P_i \leq P_{max}$ ). In our simulations we assumed  $P_{max} = 500mW$ ,  $\delta = 0.1$ ,  $\sigma^2 = 1mW$ , and BER= $10^{-4}$ . The PC that is used in these simulations has an Intel Core i7-2600 processor (3.40GHz) and 16 GB of RAM. As explained in Section 5.3, we build a set of directed links from wireless nodes using the set of potential links. We then pick a random node as a multicast source node  $s$  and also pick  $n_D$  random nodes as multicast destinations. We consider all the paths from the source node to each destination node and define a directed graph based on the superposition of those paths. The number of destinations in our simulations is 2. Now starting from this DAG, we once build a set of realizations assuming the underlying interference model is the physical model according to the procedure in Section 5.3.1, and also build another set of realizations assuming the interference model is the protocol model (based on Section 5.3.2).



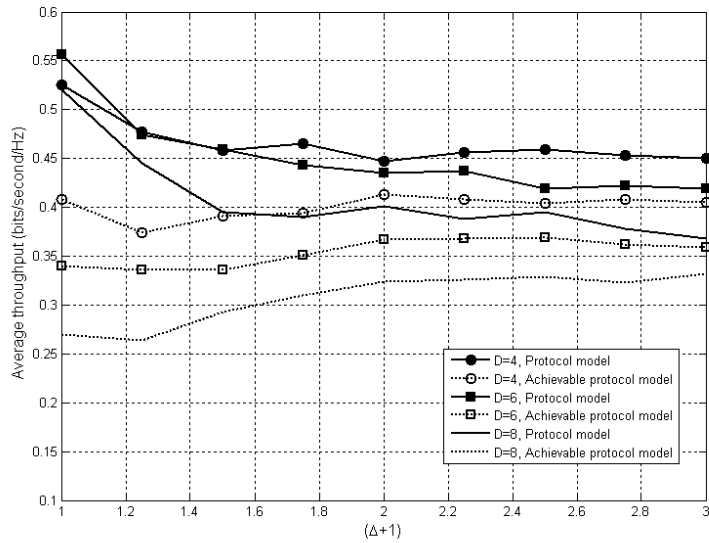
Then the optimal solution (rate/energy) of each of the two sets are determined according to the optimization problem discussed in Section 5.4. Furthermore, the solution for the protocol model is modified to incorporate the interferences for an achievable solution.

### 5.5.1 Determining the Optimal Interference Range for the Protocol Model

Different values of the interference range can affect the optimal solution by the protocol model. If the interference range is set too small, more transmitters are encouraged to transmit at the same time while they increase the interferences at the receivers and result in lowering the performance of the achievable solution. On the other hand, if the interference range is very large, it prevents some of the nodes from transmitting at the same time, while their receivers could potentially tolerate the accumulated interference of those nodes. In this case also the performance is affected. Therefore, finding an interference range that results in the best performance is a balancing act. Here, we first examine the result of varying the interference range for a network with  $N = 50$  and  $N = 100$  nodes and different densities in flow maximization problems when the interference model is the protocol model. We also re-calculate the link capacities to take interferences into account for an achievable solution. We then repeat the optimization problem for the set of realizations with links of modified capacities to get the “achievable protocol model” solution. For every transmitter node in our simulations, we consider  $\acute{R}_i = (1 + \Delta) R^{(1)}$  and set  $\Delta$  to  $\{0, 0.25, 0.5, \dots, 2\}$ . Figure 5.4 shows the maximum throughput results for the protocol model and the achievable result with the protocol model when  $N = 50$  and  $D = \{4, 6, 8\}$ . The results show that when  $\Delta$  is in the range of  $[0.75, 1.25]$  the achievable maximum rate is at its maximum. Figure 5.2 shows the same result for  $N = 100$ . All of the figures suggest setting the interference range as twice the value of  $R^{(1)}$  is a good choice for the achievable protocol model because the achievable throughput with protocol model is highest around this value. We keep this assumption for the rest of our simulations.



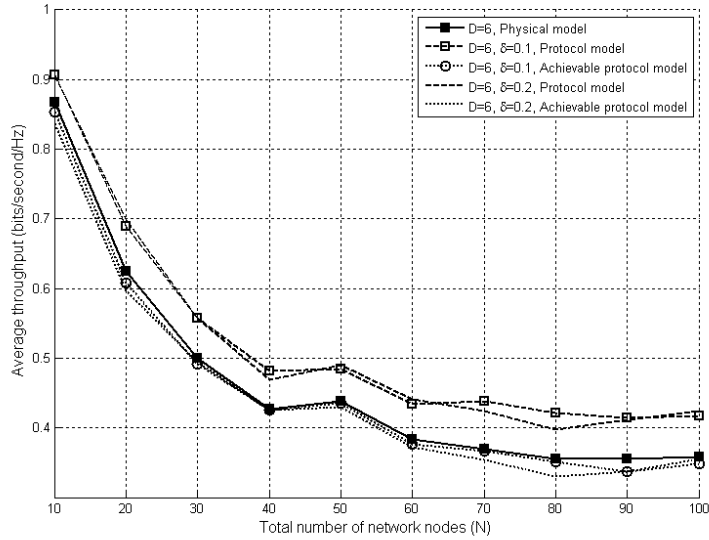
**Figure 5.1:** Comparing the maximum throughput optimization results for different interference ranges,  $N = 50$ ,  $D = \{4, 6, 8\}$ .



**Figure 5.2:** Comparing the maximum throughput optimization results for different interference ranges,  $N = 100$ ,  $D = \{4, 6, 8\}$ .

In the above simulations we fixed the value of  $\delta = 0.1$ . We also explored how sensitive

the result of the achievable protocol model is to this parameter. We compared the result of maximizing the throughput for  $N = 50$  and  $D = 6$  for  $\delta = 0.1$  and for  $\delta = 0.2$ . Higher  $\delta$  values fix the nodes' transmission powers at higher levels. The result of this comparison is shown in Figure 5.3, suggesting that the results are not very sensitive to  $\delta$ . Furthermore, we calculated the 95% confidence intervals for the protocol model and the achievable protocol model results of different  $\delta$ s and they are overlapping, which confirms that the differences are not statistically significant. In the rest of our simulations we assume  $\delta = 0.1$ .



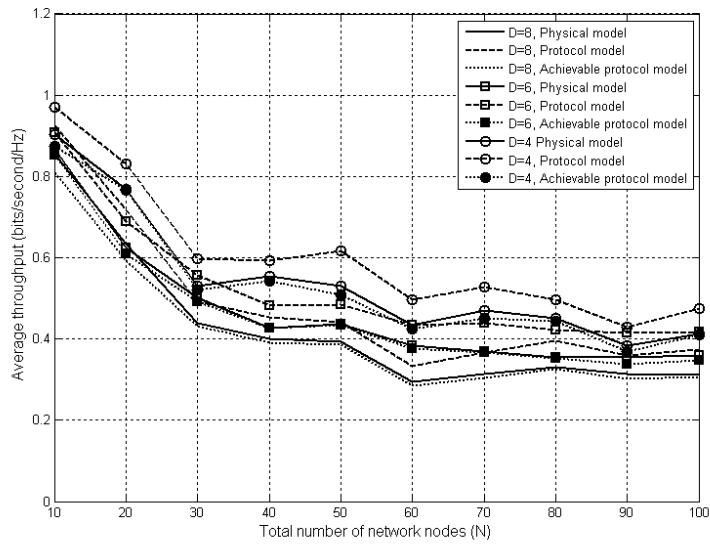
**Figure 5.3:** Comparing the effect of changing  $\delta$  in maximum throughput optimization problems,  $N = 50$ ,  $D = 6$ .

### 5.5.2 Throughput Optimization with Protocol and Physical Models

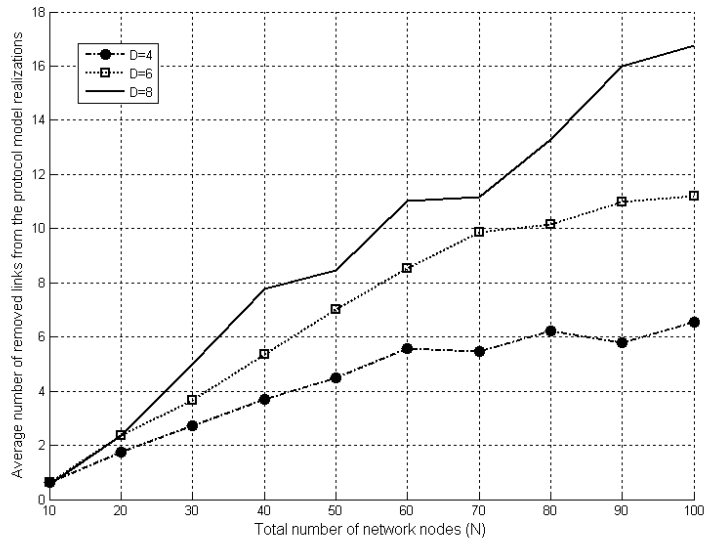
Now that we fix the interference range of the protocol model as twice  $R^{(1)}$ , we compare the performance and run-time complexities of the protocol model and the physical model. We also consider the achievable results with the protocol model and compare it with that of the physical model. We change the network size from  $N = 10$  to  $N = 100$  when the density of the network  $D$  ranges from 4 via 6 to 8. The results of our simulations are average values of the optimization results of 100 randomly generated networks. Figure 5.4 shows the average of

the maximum throughput optimization results for the physical model, the protocol model, and achievable results with the protocol model. The figure shows that the achievable result follows the physical model result very closely for all network sizes and for different densities. Also the average throughput decreases with an increase in the network size and the density. For larger networks, this is due to the fact that the source and destinations are further apart, consequently the path between the source and the destinations are longer and interferences lower the throughput. For denser networks, again, a higher level of interference is the reason for the drop in the throughput. Figure 5.5 shows the average number of the removed links from the protocol model realizations when we adjust the link capacities for the achievable solution. The figure shows that the number of removed links is larger in denser networks, which is due to the higher interferences in those networks. The figure also shows that the number of removed links is higher for larger networks, as there are typically more realizations in larger networks and therefore there are more instances of calculating the SINR that result in eliminating links. Figure 5.6 shows the average CPU times of building realizations when the interference model is the protocol or the physical model and modifying the realizations for the achievable protocol model plus the time it takes to solve the optimization problems to maximize the flow based on each of the mentioned models. Overall, the denser the network, the higher is the required CPU time. The reason is that it takes more time to build realizations in denser networks due to more interference from the adjacent nodes and also having more feasible links in those networks. That is because in denser networks, there are more unsuccessful attempts to add links to realizations in both interference models. Also due to the higher interferences, there are statistically more realizations in denser networks to ensure each link is a member of at least one realizations, which also affects the time it takes to solve the optimization problem for those networks. Second, the overall CPU times of obtaining an optimal answer when the interference model is the physical model is much larger (by up to about 30 times) than the achievable protocol model. The reason is that it takes longer to generate a sufficient set of realizations when using the physical model. For every realization, the scheduling method for the physical model has to solve an optimization

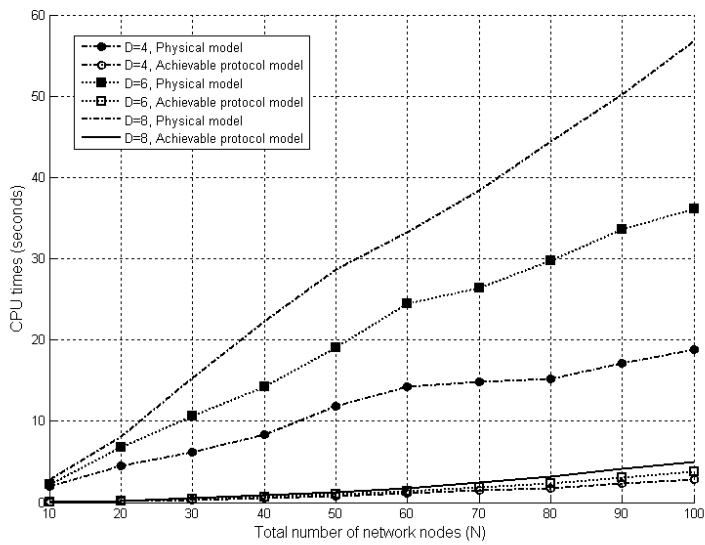
problem to determine if every randomly selected link can or cannot be part of the considered realization. Also, the slope of increasing the CPU time with the increase in the size of the network for the achievable protocol model is much less than that of the physical model. The reason is that although the CPU times are a function of the network size but the protocol model is faster and a lot less complex than the physical model. According to Figure 5.4, the optimal answers are very close, therefore we can conclude that using the achievable protocol model is more efficient while still providing quite accurate results.



**Figure 5.4:** Comparing the maximum throughputs of the different interference models



**Figure 5.5:** The average number of removed links from realizations based on the protocol model



**Figure 5.6:** Comparing the CPU times when using different interference models

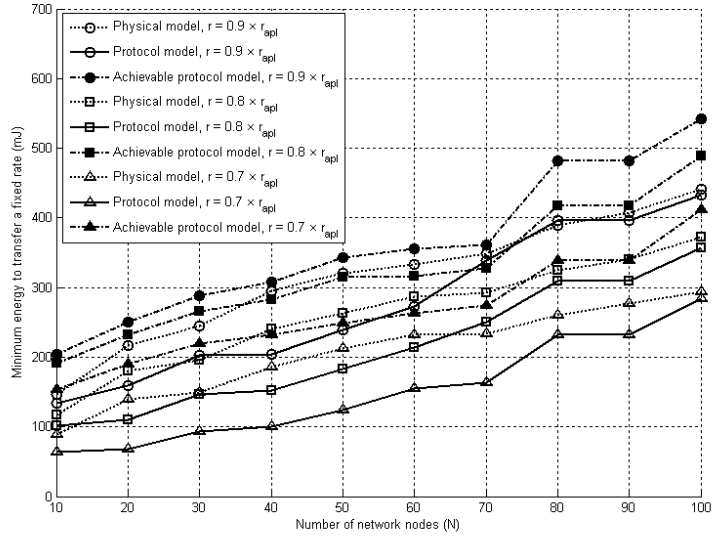
### 5.5.3 Energy Optimization with Protocol and Physical Models

The energy minimization problems are always considered for fixed (/predetermined) multicast rates. Therefore, we first consider a fixed multicast rate for the wireless network and then minimize the energy that is required to deliver the fixed rate through the network. Since we have to fix the rate before minimizing the energy, we are required to determine a rate that is achievable with both interference models. For each multicast network that we simulate, we first find the maximum achievable throughput with the protocol model (shown with  $r_{apl}$ ), which on average is slightly lower than the rate when we use the physical model. We then fix the rate to each of the  $\{0.7, 0.8, 0.9\} \times r_{apl}$  and minimize the transmission energy once assuming that the interference model is the physical model, and another time assuming that it is the protocol model. Again since the protocol model alone does not consider interferences, we need to re-adjust/modify the link capacities of the realizations from the physical model and solve the energy minimization problem again for an achievable solution. Our simulation results for networks with different densities and two destination nodes ( $n_D = 2$ ) are as follows: Figure 5.7 compares the minimum energy required to deliver rates  $r = \{0.7, 0.8, 0.9\} \times r_{apl}$ , when  $D = 6$  for different interference models. As shown in the figure, the minimum energy with the protocol interference model is the lowest energy compared to the other two results and the achievable protocol model has the highest minimum energy. The minimum energy with the protocol model is lower because it neglects interferences and thus it does not need to increase a transmitter's power to compensate for the interference. Also the achievable protocol model results in the highest minimum energy. When we re-adjust the transmission capacities of transmitters to include the effect of interference in the protocol model, the transmission capacities either stay the same or decrease to a level that their corresponding  $\gamma$  can satisfy the SINR ratio at receiving nodes of each transmitter. Therefore, supporting the same throughput with links that have lower capacities results in an increase in overall energy, because according to the objective function that we consider for minimizing the energy (Relatoin 5.9), the required transmission power is inversely proportional to the capacities. The difference between the results of the

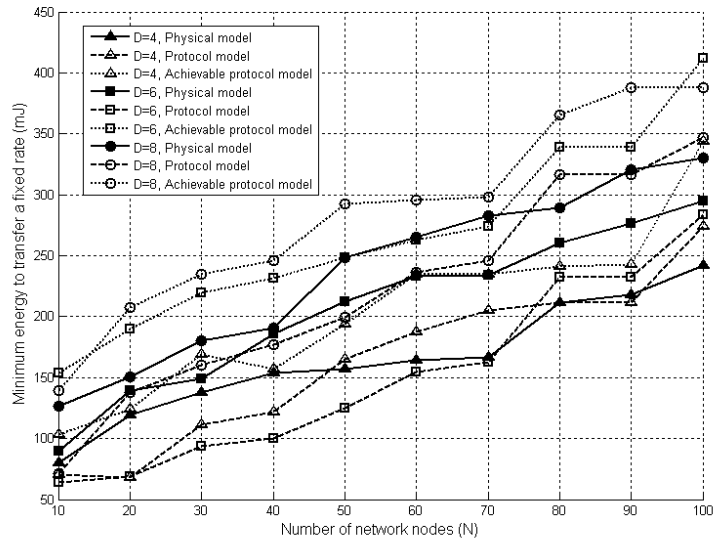
physical model and the achievable protocol model are statistically significant for all of the network sizes and therefore it seems the achievable protocol model is not the best choice for minimizing the transmission energy because it does not follow the result of the physical model very closely. Figure 5.8 shows the result of the minimum energy optimization problem for different models and different densities. This figure shows that the minimum energy required to deliver a fixed rate increases with density for all of the interference models, which is due to the additional interferences of the neighbors in dense networks that require higher transmission powers compared to less dense ones. It also shows that the protocol model results in the lowest minimum energy which of course is not necessarily achievable. The achievable protocol model results in higher minimum energy levels that is due to introducing the neglected interferences in the solution. As a result, the capacity levels of the links tend to become lower, which might affect the link rates and the overall throughput. Therefore, a higher minimum energy is required to deliver that rate. The minimum required energy when scheduling under the physical model is lower than the minimum required energy when scheduling under the achievable protocol model for all network sizes and that difference becomes statistically significant for almost all networks beyond 40 nodes (for clarity, we did not include the confidence intervals in the plot). As a result, it seems the achievable protocol model might not be a good fit for minimum energy problems. Figure 5.9 compares the average CPU times of building realizations when the interference model is either the protocol or the physical model and modifying the realizations for the achievable protocol model. It also considers the time it take to solve the flow maximization problems based on the physical and the achievable protocol model as well as the time to solve the energy minimization problem. The figure shows that the CPU times of the solutions using the physical model is 3 to 4 times that of the achievable protocol model. Also the curves seems to be exponential with respect to the number of nodes in the networks, which is different from the linear curves of CPU times of Figure 5.6. The most time-consuming procedure in both curves for physical and protocol model (that causes their run-time to be a lot closer than the comparison in Figure 5.6) is the iterative energy minimization optimization. The other



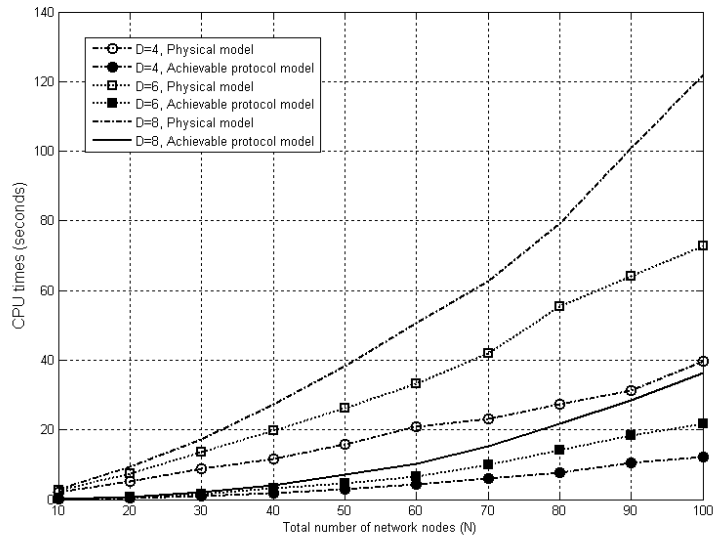
properties such as higher CPU times for denser networks, are similar to Figure 5.6 and are justified similarly. Overall, for energy minimization problems, simplifying the interference model from the physical to the protocol model in hope of having a less complex problem does not seem to be very promising as the performance of the achievable protocol model is not very close to that of the physical model and its run-time complexity is also not very low.



**Figure 5.7:** Comparing the minimum energy required to deliver a fixed rate for different interference models,  $r = \{0.7, 0.8, 0.9\} \times r_{apl}$ ,  $D = 6$ ,  $n_D = 2$ .



**Figure 5.8:** Comparing the minimum energy required to deliver a fixed rate for different interference models,  $r = 0.7 \times r_{apl}$ ,  $D = \{4, 6, 8\}$ ,  $n_D = 2$ .



**Figure 5.9:** Comparing the CPU times when using different interference models,  $r = 0.7 \times r_{apl}$ ,  $D = \{4, 6, 8\}$ ,  $n_D = 2$ .

## 5.6 Conclusions

In this chapter, we compared the performance and run-time complexity of two well-known interference models in wireless networks. We applied our comparison to networks that use transceivers with MQAM modulation. We compared how scheduling based on the physical or the protocol interference models affects the result of maximizing the throughput or minimizing the energy in multicast networks. Also, since the protocol model does not consider interferences in scheduling transmissions, we used a feasibility check to adjust the transmission capacities of the transmitters in the protocol model so that they take interferences into account. Then we examined the achievable result for the protocol model after these adjustments, which we called the feasible protocol model. Furthermore, we performed simulations in order to fix the interference range of the protocol model, which is a parameter unique to the protocol model. Our results on maximizing the throughput shows the result with the achievable protocol model is very close to the result with the physical model, while its run-time complexity is significantly lower than that of the physical model. It indicates that the achievable protocol model can be a good replacement for the much more complex physical model when solving flow maximization problems. The energy minimization results, however, show that the minimum energy based on the achievable protocol model is not very close to that of the physical model. Its run-time complexity is lower than that of the physical model, but less so than in the case of studying maximum throughput problems. So based on our results, the physical model is a better choice when studying energy minimization problems because the achievable protocol model does not result in a very accurate solution while its complexity is also quite high.

## Chapter 6

# Capacity-bundling Scheduling: An Improved MAC Scheme for Joint Scheduling and Network Coding in Wireless Networks

This chapter introduces a new scheduling heuristic for multiple access in wireless communication networks. This new heuristic is based on activating links with similar capacity simultaneously and we apply them to our joint MAC scheduling and network coding optimization problem. To formulate the problem, a physical layer interference model capable of handling variable-power variable-rate scenarios is considered. Our simulations show that the new scheduling heuristic, referred to as *capacity-bundling*, improves significantly over the existing scheme which do not impose any constraint on the capacity of the links that are simultaneously activated. We perform our simulations for multicast scenarios of networks that employ a set of embedded M-ary Quadrature Amplitude Modulation (MQAM) signal constellations for transmission. The improvement resulting from the new scheduling scheme is up to about 80% for the throughput, and about 55% for the energy consumption in the scenarios simulated in our work. The network model in this chapter is the same as the model presented in Section 3.1. The channel model is adopted from Section 5.2.1 and the interference model is the physical model (presented in Section 5.2.2). The joint optimization problems for energy and throughput are solved based on Section 5.4.

## 6.1 Capacity Bundling Scheduling

The joint optimization problems discussed in the previous section are formulated based on a given set of realizations for the wireless network. As discussed previously, finding the optimal solution to this scheduling problem is NP hard, therefore researchers fall back on heuristics. There are works that attempt to formulate the capacity region of the networks, but the optimal scheduling for such networks are only computable under certain considerations such as having single-hop transmission [42]. We however are more interested in finding practical scheduling methods for general networks. It is well-known that depending on the selected set of realizations, the solution of the optimization problem would differ. In this work we propose an improved heuristic to determine realizations that result in better network coding performance (higher throughput or lower energy consumption). In the following, we briefly mention basic scheduling, used in the existing literature, for the construction of realizations. We then provide an example to motivate the capacity-bundling scheduling heuristic proposed here. This is followed by the detailed description and evaluation of the proposed scheduling heuristic.

### 6.1.1 Basic Scheduling for a Network with Variable-Rate, Variable-Power Nodes

Based on [9], realizations are packed while avoiding interferences (using the protocol model) and until each node in the network is assigned to be transmitter and receiver at least once (except the source and destination nodes). The mentioned scheduling has been formulated based on the protocol model and thus assumes nodes with fixed transmit power and links with fixed capacities. Here we however use a modified version of it adapted to the physical interference model. This is in fact the same scheduling heuristic that we used in Section 5.3.1 and we refer to it as “Basic Scheduling”. Preliminary assumptions about the network and the modulation are also similar to the ones explained in Section 5.3.

### 6.1.2 Capacity-bundling for a Network with Variable-Rate, Variable-Power Nodes

The link capacities in our model are a function of both the transmission powers and the constellation sizes. Furthermore, both the total capacity of a link  $C_{ij}$  and the effective capacity of a link during a specific timeshare,  $C_{ij}^{(m)}$ , depend on the scheduling timeshares and link capacities,  $c_{ij}^{(m)}$ . Now assume that the link capacities of a network are equal to either  $c_1$  or  $c_2$  and  $c_1 < c_2$ . In the ideal case, if we can schedule links with the same capacity together, then according to the capacity constraint of our joint optimization problem and the fact that the summation of all the timeshares is equal to a working cycle, we expect that the time share associated with realizations that have lower capacity links increase and the timeshares of realizations containing higher capacity links decrease, while collectively offering a higher common rate  $r$  to all destinations. As the working cycle is normalized, bundling the large (small) capacities in the same realization causes their respective timeshares to have small (large) values. On the other hand, if a link with very large capacity is scheduled with one which has a small capacity, the resulting realization may be assigned a relatively large time share. The flow on the higher capacity link may not be large enough during the realization to fully utilize the link capacity (link flows are upperbounded by  $\tau_m c_{ij}^{(m)} \mathbf{1}_{E_m^f((i,j))}$ ).

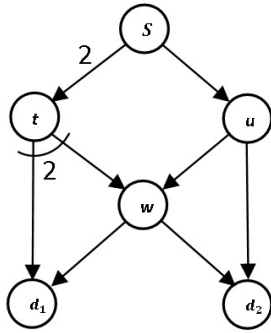
Next we present an example that shows how capacity-bundling helps to improve the throughput.

**Example 5.1.** Consider the wireless network shown in Figure 6.1. This network is the wireless version of the butterfly network presented in [1]. The network has one multicast source node  $S$ , and two multicast destination nodes  $d_1$  and  $d_2$ . The capacity of links  $(S, t)$ ,  $(t, w)$ , and  $(t, d_1)$  is equal to 2 packets per timeslot, while the other links have unit capacities. Figure 6.2 shows a set of realizations from basic scheduling presented in [9], [10]. Maximizing the throughput for this set results in:  $\tau_1 = \tau_2 = \frac{1}{4}$ , and  $\tau_3 = \frac{1}{2}$ , and the maximal rate using basic scheduling is  $r_B = \frac{3}{4}$ . Now if we add new realizations based on capacity-bundling, shown in Figure 6.3, then solving the optimization problem for the set with 7 realizations

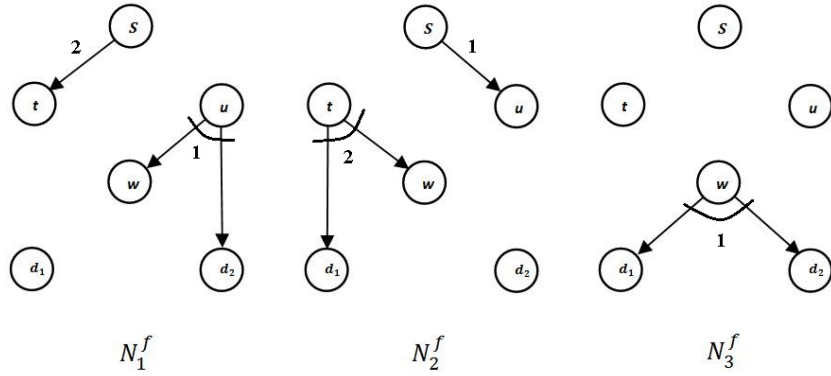
results in

$$\tau = \{0.2810, 0.1398, 0.0730, 0, 0.0602, 0.3270, 0.1190\},$$

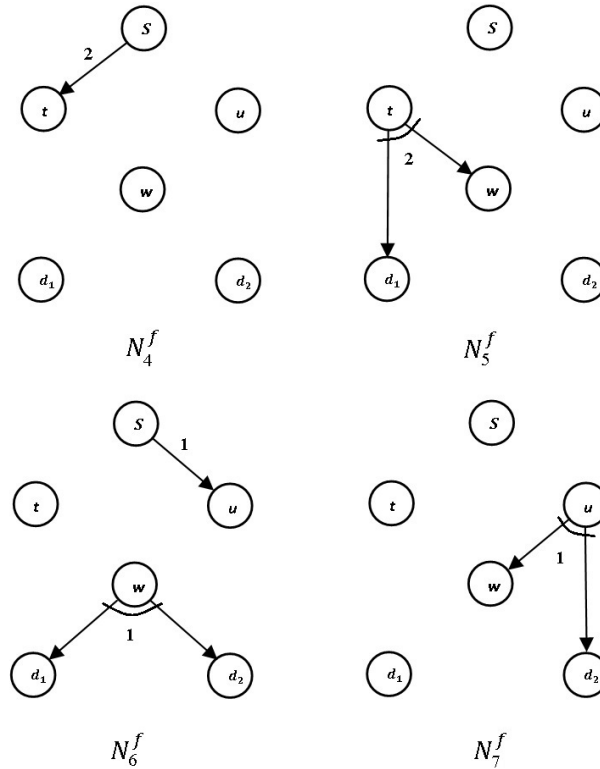
and the maximal rate with capacity-bundling is  $r_{CB} = \frac{4}{5}$ , which is higher than  $r_B$ . This second set of realizations is derived using a heuristic that bundles links with equal capacity in the same realization, explained below. Optimizing over these additional realizations results in MAC schedules which support higher throughputs. As the results reported later in this paper show, this improvement increases with network size and can be quite substantial.  $\square$



**Figure 6.1:** Wireless butterfly network with different link capacities



**Figure 6.2:** A set of realizations using basic scheduling



**Figure 6.3:** A set of additional realizations based on scheduling links with equal capacities

Therefore, we propose to derive additional realizations by scheduling links with equal capacities in the same realization. Extending the basic scheduling heuristic discussed above, to build realizations based on bundling similar capacity links, we start with links of the highest capacity, and only pack together, in the same realization, links with the same capacity that do not violate the SINR constraint of the already activated links. This process continues until we cannot add another link without violating the SINR constraints of already established links. We repeatedly generate realizations at this capacity level until all possible links have been scheduled at least once. Then we continue with the next lower capacity level in our system. All the preliminary steps (towards building a directed graph) along with the assumptions regarding the node placement and BER, are the same for the two schemes. The only difference between the additional realizations based on bundling links of equal capacities and those of basic scheduling is in the optimization problem that determines



the realizations. For realizations that only schedule links of equal capacity, all the SINR thresholds are the same. To form the final set of realizations, we add these additional realizations (such as the ones shown in Figure 6.3) to the ones derived from the basic scheduling heuristic (such as the ones shown in Figure 6.2).

In the next part, we compare the results of maximizing the throughput and also minimizing the energy for a fixed rate using the two different scheduling heuristics: basic scheduling and our proposed capacity-bundling scheduling.

## 6.2 Simulation Results

### 6.2.1 Throughput Maximization with and without Capacity-Bundling

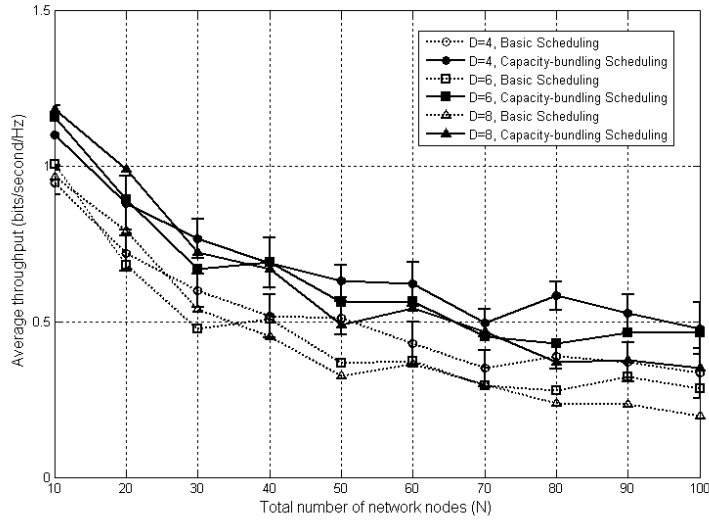
The assumptions for the networks simulated in this chapter are the same as the one explained in Section 5.5. After building a DAG for every network, we use the basic scheduling described in Section 6.1.1 and determine a set of realizations for this network. We then add new realizations based on bundling links with equal capacities, as discussed in Section 6.1, to the set of realizations from basic scheduling and solve the throughput maximization problem of Section 6.1 to obtain the maximum throughput for the realizations from basic scheduling as well as the capacity-bundling realizations. The results are shown in Figures 6.4 and 6.5. Keeping the network density constant at 6 and varying the network size, one can observe from Figure 6.4 that the throughput for capacity-bundling is always higher than that of basic scheduling. The average throughput decreases with an increase in network size since in larger networks, the source and destinations are further apart and more links along the paths from the source to the destinations might act as bottlenecks for the information flow. We also calculated the 95% confidence intervals for these results and the difference between the two scheduling heuristics are statistically significant. As an example the confidence intervals for  $D = 4$  are shown in Figure 6.4.

We define the throughput improvement of the capacity-bundling, ( $CB$ ), compared with

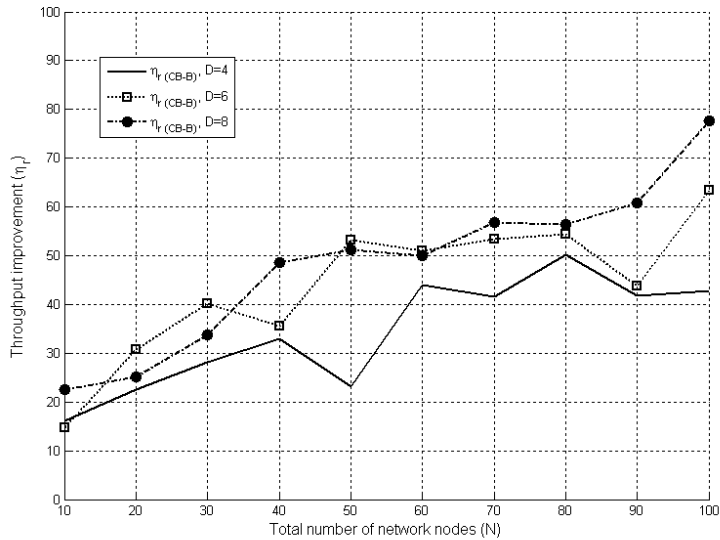
basic scheduling, ( $B$ ), as follows:

$$\eta_{r(CB-B)} = \frac{r_{CB} - r_B}{r_B}. \quad (6.1)$$

Figure 6.5 shows  $\eta_{r(CB-B)}$ . The maximum throughput improvement for different network densities is between about 50% to about 80% for the simulated networks. The improvement generally increases with network size, which shows using realizations from capacity-bundling is more advantageous in larger networks.



**Figure 6.4:** Comparing the average throughput of basic scheduling with that of capacity-bundling,  $D = \{4, 6, 8\}$ ,  $n_D = 2$



**Figure 6.5:** The throughput improvement of capacity-bundling over basic scheduling,  $D = \{4, 6, 8\}$ ,  $n_D = 2$

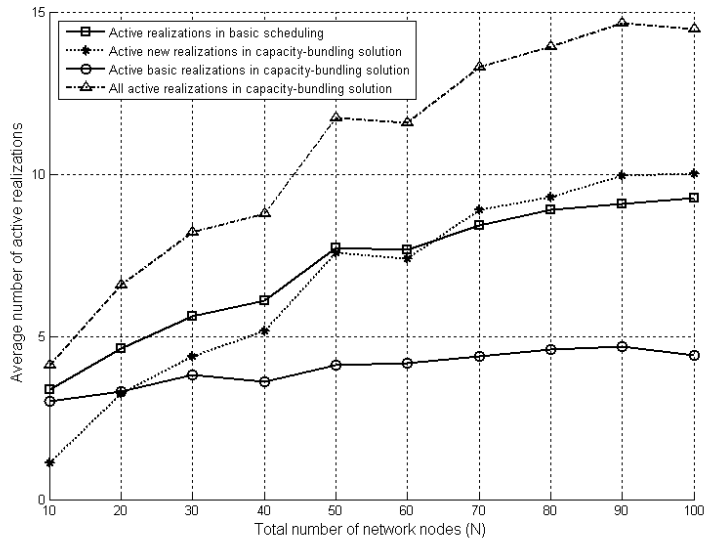
Figure 6.4 also shows that the maximum throughput of both basic scheduling and capacity-bundling decreases slightly with an increase in  $D$  which is due to the increased level of interference in denser networks. On the other hand, Figure 6.5 shows that the throughput improvement slightly increases with the density, which shows realizations from capacity-bundling are more advantageous in denser networks. For example, for a network density of 8 and larger networks, the improvement nearly reaches 80%.

We also investigated the number of active realizations (that have non-zero time fractions) for maximum throughput solutions for basic scheduling and capacity-bundling solutions. Figure 6.6, shows the average number of active realizations for  $D = 6$ . This figure shows that the number of active realizations in a capacity-bundling set is larger than that of basic scheduling. A large fraction of active realizations in the capacity-bundling set are those additional realizations that bundle links of equal capacity. Furthermore, we also compared the percentage of realizations that have links whose flows are less than their effective capacity in basic scheduling and capacity bundling scheduling. Figure 6.7 shows that this percentage is higher in basic scheduling. Capacity bundling results in network realizations that use the

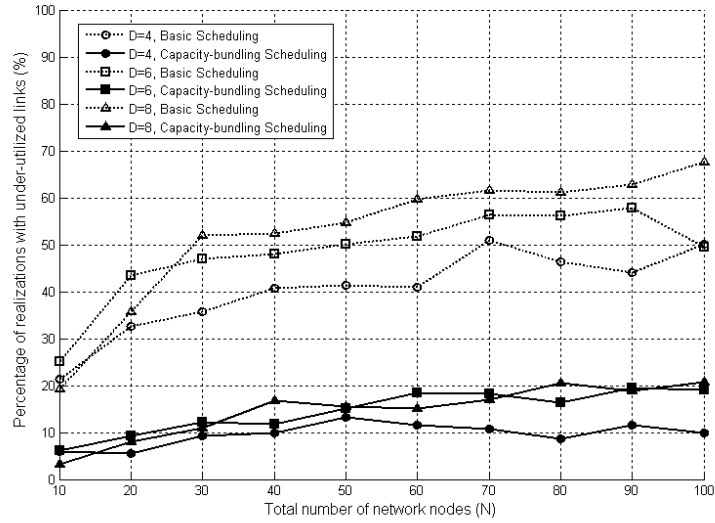
available link capacities more efficiently, ultimately resulting in higher overall throughputs compared to basic scheduling as depicted in Figure 6.4. Consider the following parameter:

$$\zeta = \frac{\sum_{m=1}^{\mathcal{M}} \sum_{i \in V_m^f} (C_i^{(m)} - f_i^{(m)})}{\mathcal{M}_a \sum_{m=1}^{\mathcal{M}} \sum_{i \in V_m^f} C_i^{(m)}}, \quad (6.2)$$

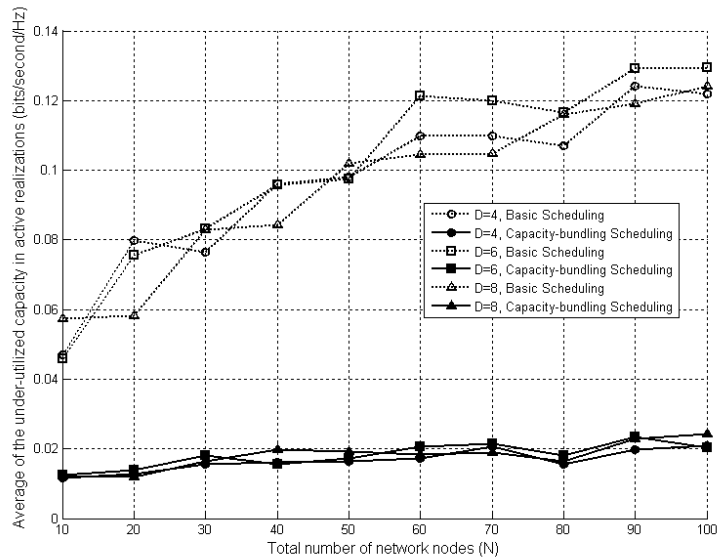
where  $\mathcal{M}_a$  represents the number of active realizations. Parameter  $\zeta$  is a normalized measure of how under-utilized the links in each scheduling scheme are. In Figure 6.8, we have plotted the average value of  $\zeta$  (averaged over all simulated networks per realization) for both basic and capacity-bundling schedules. This figure shows that the under-utilized capacity in basic scheduling is about 4 to 7 times higher than that of the capacity bundling scheduling and that  $\zeta$  increases with network size for the basic scheduling method.



**Figure 6.6:** Average number of active realizations in different scheduling methods,  $D = 6$ ,  $n_D = 2$



**Figure 6.7:** Comparing the percentage of realizations with links that under-utilize their capacity,  $D = \{4, 6, 8\}$ ,  $n_D = 2$ .



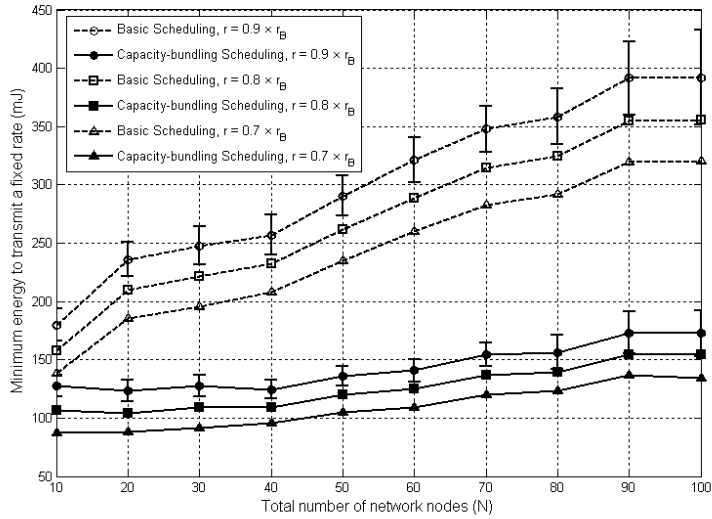
**Figure 6.8:** The average normalized under-utilized capacity for basic and capacity-bundling scheduling methods,  $D = \{4, 6, 8\}$ ,  $n_D = 2$ .

### 6.2.2 Energy Minimization with and without Capacity-Bundling

In minimum energy problems, we first consider a fixed multicast rate for the wireless network and then minimize the energy that is required to deliver the fixed rate through the network. Since we have to fix the rate before minimizing the energy, we are required to determine a rate that is achievable. For each multicast network that we simulate, we first solve the throughput maximization problem and use basic scheduling to schedule transmissions (the result is  $r_B$ , which on average is lower than  $r_{CB}$ ). We then fix the rate to  $\{0.7, 0.8, 0.9\} \times r_B$  and minimize the transmission energy under both scheduling heuristics for these three rates.

#### Varying the Network Size

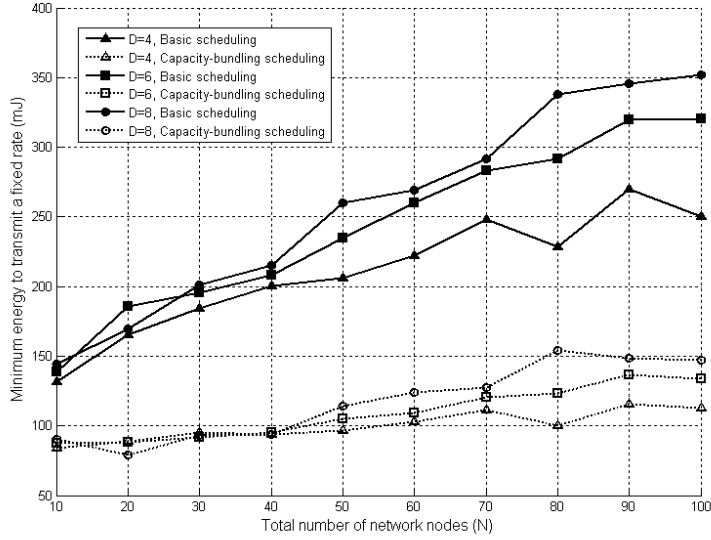
We explore minimizing energy for different network sizes with and without capacity-bundling. Minimum required energy with basic scheduling and capacity-bundling are denoted by  $\mathcal{E}_B$ , and  $\mathcal{E}_{CB}$ , respectively. Figure 6.9 compares the minimum required energy using the two scheduling heuristics for different network sizes. Each group of three plots show the minimum energy results for the three mentioned fixed rates. For each group, from top to bottom, the plots belong to  $r = 0.9 \times r_B$ ,  $r = 0.8 \times r_B$ , and  $r = 0.7 \times r_B$ , respectively. The results show that an energy-optimal solution derived with basic scheduling consumes significantly more energy compared to the solution for the capacity-bundling scheduling. We also calculated the 95% confidence intervals of the results and the reported differences between the two scheduling schemes are statistically significant. As an example, the confidence intervals for  $r = 0.9 \times r_B$  are shown in Figure 6.9. Also the energy in both cases increase with the number of network nodes, because as the network grows, the paths between the source and destination nodes become longer.



**Figure 6.9:** Comparing the minimum energy required to deliver a fixed rate using basic and capacity-bundling scheduling methods,  $r = \{0.7, 0.8, 0.9\} \times r_B$ ,  $D = 6$ ,  $n_D = 2$ .

### Varying Neighboring Nodes Densities

Next, we focus on minimizing the energy for different node densities. In our simulations,  $n_D = 2$  and  $D = \{4, 6, 8\}$ . We fix the rate at  $0.7 \times r_B$  and then minimize the energy as explained at the beginning of this section. Our results in Figure 6.10 show that the minimum required energy to deliver a fixed rate increases with  $D$ , which may be due to the additional interferences of the neighbors in dense networks that requires higher transmission powers compared to less dense ones.



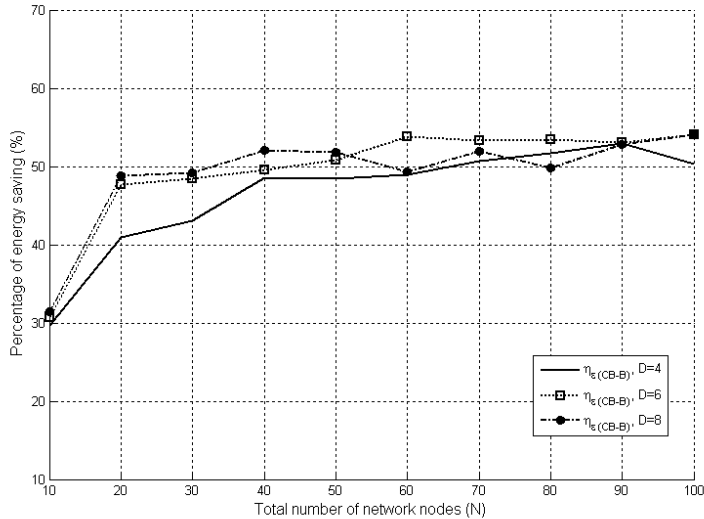
**Figure 6.10:** Comparing the minimum energy required to deliver a fixed rate using basic and capacity-bundling scheduling methods,  $r = 0.7 \times r_B$ ,  $D = \{4, 6, 8\}$ ,  $n_D = 2$ .

We also define  $\eta_{\epsilon(CB-B)}$  as the percentage of energy saving that results from switching the scheduling heuristic from basic scheduling to capacity-bundling.

$$\eta_{\epsilon(CB-B)} = \frac{\mathcal{E}_B - \mathcal{E}_{CB}}{\mathcal{E}_B}. \quad (6.3)$$

Our simulation results in Figure 6.11 show that  $\eta_{\epsilon(CB-B)}$  increases slightly with network size and we can save up to about 55% in energy if we switch from basic scheduling to capacity-bundling scheduling. We save energy when deriving realizations based on our proposed heuristic because, as shown in Figure 6.6, there are more active realizations in a capacity-bundling solution compared to basic scheduling. As a result, the transmissions are spread among a larger number of realizations which decreases the interferences among transmissions and hence lowers the required transmission power compared to the required power when the realizations are based on basic scheduling.





**Figure 6.11:** The energy saving of capacity-bundling compared to basic scheduling,  $r = 0.7 \times r_B$ ,  $D = \{4, 6, 8\}$ ,  $n_D = 2$ .

### 6.3 Conclusions

In this chapter, we introduced a new scheduling heuristic for joint optimization of MAC scheduling and network coding in wireless multicast networks. We proposed to schedule links of equal capacity in the same realization, hence the term “capacity-bundling”. To schedule transmissions with our proposed scheme, we used the physical interference model developed for multi-rate, multi-power radios, described in Chapter 5. Our simulation results show that scheduling based on capacity-bundling improves the result of both throughput maximization and energy minimization problems compared to the corresponding results from basic scheduling. The improvement is up to about 80% and 55% in the throughput maximization and energy minimization problems, respectively. Throughput improvements are more pronounced as the network size grows and networks get denser. Energy savings are rather constant across network density and have a slightly increasing trend with network size. Overall, we introduced link capacities as a new critical factor in selecting realizations.

## Chapter 7

# Conclusions and Future Work

### 7.1 Conclusions

Applying network coding to wireless networks brings new opportunities in terms of improving the throughput and minimizing the energy consumption as well as unique challenges due to the broadcast nature of the wireless media. In this work, we focused on the problem of joint scheduling and network coding in those networks. The main conclusions for the studied problems are summarized below.

- We formulated the joint scheduling and network coding optimization problem for wireless networks in a linear form with schedule-specific flows. Having flows whose membership to different realizations are known in the optimal solution is necessary for designing network codes. To preserve the broadcast property of the wireless transmissions, we also adapted a wireless broadcast model to correctly model those transmissions.
- We studied the requirements to adapt a network code that is designed for the wired networks to the wireless networks. We showed that if the network code design method is not able to accommodate unequal timeshares, the network might not achieve the predicted performance by the optimization problem. We also explained wireless-aware code constructions that preserve the broadcast property of the transmissions in these networks.

- We compared the performance and run-time complexity of two well-known interference models in wireless networks. We applied our comparison to networks that use transceivers with MQAM modulation. We compared how scheduling based on the physical or the protocol interference models affects the result of maximizing the throughput or minimizing the energy in multicast networks. Also, since the protocol model does not consider interferences in scheduling transmissions, we used an extension of the protocol model, called feasible protocol model, which adjusts the transmission capacities of the transmitters in the protocol model so that they take interferences into account. Based on our simulations we conclude that the achievable protocol model can be used without loss of performance in throughput maximization problems. It also has significantly lower run-time complexity than the physical model. In energy minimization problems however, the physical model might be a better choice because the performance of the achievable protocol model is significantly lower while its complexity is also quite high.
- Finally, we studied an existing heuristic method of scheduling and introduced a new one for wireless multicast networks. We proposed to schedule links based on a “capacity-bundling” method which is scheduling links of equal capacity in the same realization. In our work, we considered the physical interference model suited for variable-power, variable-rate nodes. Our simulation results show that scheduling based on capacity-bundling substantially improves the result of both throughput maximization and energy minimization problems compared to the corresponding results from basic scheduling.

## 7.2 Future Work

The work described in the thesis provides a basis for a number of future research opportunities. Our first suggestion is to study the effect of increasing the number of destination nodes when comparing the performance and run-time complexity of the physical and the

protocol model. Our simulations are performed for networks with two destination nodes. It would be interesting to see if the performance of the protocol model in comparison with the physical model improves or degrades for higher numbers of destination nodes. The problem can be studied for both maximizing the throughput and minimizing the energy. Regarding the effect of increasing  $n_D$  on run-time complexity, we expect that the CPU times increase with an increase in the number of destination nodes. The reason is that the number of relations in (3.4) and (3.6) is a linear function of  $n_D$  and that adds to the complexity of solving the optimization problem for both models. It should be investigated however whether the difference between the run-time complexities of the two models persists, increases, or decreases with an increase in the number of destination nodes for both energy and throughput problems.

Similarly, we propose to study the performance of the “capacity-bundling” scheduling compared to the “basic scheduling” when the number of destination nodes increases. Such a study would help to determine whether the performance gains of the capacity-bundling method persist, increase or decrease with an increase in the number of destination nodes.

Another future work is devising new heuristics for scheduling wireless transmission. Since this problem is NP-hard and is not explored in much depth, we suggest focusing on other criteria (besides interference and capacities) that may be used when heuristically forming realizations. Related to this, one could explore ways to determine bounds on maximum throughput or minimum energy problems in polynomial time to be able to evaluate how good a proposed heuristic is. Our final suggestion about the future work is studying optimization problems and code designs for networks in which only some of the nodes are assigned/able to perform network coding. If some of the nodes can perform network coding and some can only forward messages, then the optimization problems presented in our work can not be readily applied to them. These networks need a new formulation for optimizing any performance measure of interest. The new formulation should integrate the “relay and forward” property of the routing messages at nodes that are not able to code with the “flow conservation constraints” that are written per destination nodes for the ones that are

able to code. Such a formulation for throughput should result in an optimal answer that is upperbounded by the min-cut max-flow throughput and lowerbounded by the routing throughput. Proper considerations are also required for designing network codes for such networks.

## List of References

- [1] R. Ahlswede, C. Ning, S. Y. R. Li and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, Vol. 46, No. 4, pp. 1204-1216, 2000.
- [2] P. Chaporkar, and A. Proutiere, "Adaptive Network Coding and Scheduling for Maximizing Throughput in Wireless Network," *Proc. MobiCom'07*, Sep 9-14, Montreal, Quebec, Canada, pp. 135-146.
- [3] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance," *MobiCom'03*, Sep 14-19, 2003, San Diego, California, pp. 66-80.
- [4] A. Ephremides, and T.V. Truong, "Scheduling Broadcasts in Multihop Radio Networks," *IEEE Transactions on Communications*, Vol. 38, No. 4, pp. 456 - 460, Apr. 1990.
- [5] E. Arikan, "Some complexity results about packet networks," *IEEE Transaction on Information Theory*, Vol. 30, No. 4, pp. 681-685, July 1984.
- [6] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S. Y. Kung, "Network Planning in Wireless Ad-Hoc Networks: A Cross-Layer Approach," *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 1, pp. 136-150, Jan. 2005.
- [7] Y. Wu, P. A. Chou, and S. Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Trans. on Communications*, Vol. 53, No. 11, pp. 1906-1918, Nov. 2005.
- [8] J. S. Park, D. S. Lun, F. Soldo, M. Gerla, and M. Medard, "Performance of Network Coding in Ad hoc Networks," *Proc. of IEEE Military Communications Conference (Milcom)*, 2006, Washington, DC, USA, pp. 1-6.
- [9] Y. E. Sagduyu, and A. Ephremides, "Joint Scheduling and Wireless Network Coding," *Proc. First Workshop on Network Coding, Theory and Applications*, Riva del Garda, Italy, Apr. 2005.
- [10] Y. E. Sagduyu, and A. Ephremides, "On Joint MAC and Network Coding in Wireless Ad Hoc Networks," *IEEE Transactions on Information Theory*, Vol. 53, No. 10, pp. 3697-3713, Oct. 2007.

- [11] D. Traskov, M. Heindlmaier, M. Medard, R. Koetter, and D. S. Lun, "Scheduling for Network Coded Multicast: A Conflict Graph Formulation", *4th IEEE Workshop on Broadband Wireless Access*, December 2008, New Orleans, LA, USA, pp. 1-5.
- [12] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, and E. Ahmed, "Minimum-cost multicast over coded packet networks," *IEEE Transactions on Information Theory*, Vol. 52, No. 6, pp. 2608-2623, June 2006.
- [13] M. Bama, S. Bhashyam, A. Thangaraj, "Node-Based Scheduling for Network-Coded Wireless Networks," *Proc. of IEEE COMSWARE 2008, Workshop on Wireless Systems: Advanced Research and Development (WISARD 2008)*, Bangalore, India, January 2008 (Poster).
- [14] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "On the Benefits of Network Coding for Wireless Applications," *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2006, Boston, MA, USA, pp. 1-6.
- [15] C. Fragouli, and E. Soljanin, "Network Coding Fundamentals," now Publishers Inc., 2007, ISBN: 1601980329.
- [16] R. W. Yeung, "Information Theory and Network Coding," *Springer*, 2008, ISBN:9780387792330.
- [17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in The Air: Practical Wireless Network Coding," In *ACM SIGCOMM 2006*, Pisa, Italy, Sep. 11-15, 2006, pp. 243-254.
- [18] L. R. Ford, D. R. Fulkerson, "Flows in Networks," *Princeton University Press*, 1962. ISBN: 9780691146676.
- [19] E. Lawler "4.5. Combinatorial Implications of Max-Flow Min-Cut Theorem, 4.6. Linear Programming Interpretation of Max-Flow Min-Cut Theorem". *Combinatorial Optimization: Networks and Matroids*, Dover, pp. 117-120, 2001.
- [20] T. Ho, and D. S. Lun, "Network Coding, An Introduction," Cambridge University Press, 2008, ISBN: 9780521873109.
- [21] R. Koetter, and M. Medard, "Beyond Routing: An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, Volume 11, Issue 5, pp. 782-796, October 2003.
- [22] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithm for multicast code construction," *IEEE Transactions on Information Theory*, Vol. 51, No. 6, pp. 1971-1982, June 2005.
- [23] R. Koetter, M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, Vol. 11, No. 5, pp. 782-795, Oct. 2003.
- [24] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, D. Karger, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Information Theory*, Vol.52, No.10, pp. 4413-3340, 2006.

- [25] G. Kang, Y. Taniguchi, G. Hasegawa, and H. Nakano, "Extending the Protocol Interference Model Considering SINR for Wireless Mesh Networks," in *Proc. of The Advanced International Conference on Telecommunications (AICT)*, St. Maarten, The Netherland Antilles, March 20-25, 2011, pp. 26-31.
- [26] G. Kang, Y. Taniguchi, G. Hasegawa, and H. Nakano, "Parameter tuning of the protocol interference model using SINR for time slot assignment in wireless mesh networks," *IEICE Trans. Inf. Syst.*, Vol. E94-D, No. 11, Nov. 2011, pp. 2191-2200.
- [27] A. Iyler, C. Rosenberg, and A. Karnik, "What is the Right Model for Wireless Channel Interference?," in *IEEE Trans. on Wireless Comm.*, Tenerife, Canary Islands, Spain, Oct. 26-30, 2009, pp. 2662-2671.
- [28] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, "How to correctly use the protocol interference model for multi-hop wireless networks", *MobiHoc09*, May 18-21, 2009, New Orleans, LA, USA, pp.239-248.
- [29] R. Maheshwari, S. Jain, and S. R. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks, in *Proceedings of ACM SenSys 2008.*, Nov. 2008, Raleigh, NC, USA, pp. 141-154.
- [30] A. Kuntz, F. Schmidt-Eisenlohr, O. Graute, H. Hartenstein, M. Zitterbart, "Introducing probabilistic radio propagation models in OMNeT++ mobility framework and cross validation check with NS-2," *SIMUTools*, March 2008.
- [31] C.H. Papadimitriou, and K. Steiglitz, "Combinatorial Optimization," *Englewood Clis, NJ: Prentice Hall, 1082.*, ISBN: 9780486402581.
- [32] B. Hajek, and G. Sasaki, "Link Scheduling in Polynomial Time," *IEEE Transactions of Information Theory*, Vol. 34, No. 5, pp. 910-918, Sep. 1988.
- [33] S. C. Ergen, and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, Vol. 16, No. 4, pp. 985-997, May 2010.
- [34] R. Bhatia, and E.L. Li "Characterizing achievable multicast rates in multi-hop wireless networks," *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc 2005, New York, NY, USA, pp. 133-144.
- [35] A. Ephremides, and T.V. Truong, "Scheduling Broadcasts in Multihop Radio Networks," *IEEE Transaction on Communications*, Vol. 38, No. 4, April 1990, pp. 456-460.
- [36] X. Y. Li, J. Zhao, Y. W. Wu, S. J. Tang, X. H. Xu, X. F. Mao, "Broadcast capacity for wireless ad hoc networks," *Proc. 5th International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2008*, Atlanta, GA, Sep. 29-Oct. 2, 2008, pp. 114-123.
- [37] S. Jaggi, P. A. Chou, and K. Jain, "Low Complexity Algebraic Multicast Network Codes," *Proc. ISIT*, June 2003, Yokohama, Japan, pp. 368-368.



- [38] R. Niati, A. H. Banihashemi, and T. Kunz, “On Code Design in Joint MAC Scheduling and Wireless Network Coding,” *Proceedings of the 25th Biennial Symposium on Communications*, Queens University, Kingston, Canada, May 12-14, 2010, pp. 456-460.
- [39] A. Keshavarz-Haddad, R. Riedi, “On the Broadcast Capacity of Multihop Wireless Networks: Interplay of Power, Density and Interference”, in *Proc. 4th IEEE SECON'07*, June 18-21, 2007, San Diego, CA, USA, pp. 314-323.
- [40] C. Chekuri, C. Fragouli, and E. Soljanin, On average throughput and alphabet size in network coding, *IEEE Transactions on Information Theory*, Vol. 52, No. 6, pp. 2410-2424, June 2006.
- [41] A. Goldsmith, Wireless Communications, *Cambridge University Press*, 2005, ISBN: 978-0-521-83716-3, pp. 31, 289.
- [42] X. Lin, N. B. Shroff, and R. Srikant, “A tutorial on cross-layer optimization in wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452-1463, Aug. 2006.

## Appendix A

### Derivations for Chapter 3

#### A.1 Determining the Size of the Simulation Field Based on the Average Number of Neighbor Nodes

Assume we have a network with  $N$  nodes. The nodes are randomly placed in a  $2D$  rectangular field. We assume the length ( $l$ ) of the network is equal to  $\alpha w$ .  $w$  is the width of the field and we assume  $\alpha$  is fixed (in our simulation  $\alpha = 3$ ).

$$l = \alpha w$$

We want to determine the field size when we know the number of network nodes and also we have fixed the average number of neighbor nodes. Each node has two radio ranges  $R_1$  and  $R_2$ . Nodes inside  $R_1$  of a wireless node are connected to it, and the nodes outside  $R_2$  are not. Connectivity between  $R_1$  and  $R_2$  is distributed uniformly.

Now if we want to determine the probability of having a connection from node  $i$  towards node  $j$  in a network with  $N$  nodes, we can define it as a Bernoulli distribution. "Success" is having a connection between the two nodes and "Failure" is not having a connection. This probability distribution function is a function of the distance between the two nodes

as well. Therefore, we have a joint distribution for success and failure probabilities. The distance between  $i$  and  $j$  is  $d_{ij}$ .

We can define these probabilities as follows:

Note that  $P(C) = P((i, j) \in E)$  and  $P(D) = P((i, j) \notin E)$  and  $E$  is the set of network edges.

$$p = P(C, d_{ij}) = P(C|d_{ij} \leq R_1)P(d_{ij} \leq R_1)$$

$$+P(C|R_1 \leq d_{ij} \leq R_2)P(R_1 \leq d_{ij} \leq R_2)$$

$$q = P(D, d_{ij}) = P(D|R_1 \leq d_{ij} \leq R_2)P(R_1 \leq d_{ij} \leq R_2)$$

$$+P(D|d_{ij} \geq R_2)P(d_{ij} \geq R_2)$$

in which:

$$P(C|d_{ij} \leq R_1) = 1$$

$$P(C|R_1 \leq d_{ij} \leq R_2) = \frac{R_2 - d_{ij}}{R_2 - R_1}$$

$$P(D|R_1 \leq d_{ij} \leq R_2) = \frac{d_{ij} - R_1}{R_2 - R_1}$$

$$P(D|d_{ij} \geq R_2) = 1$$

Also

$$P(d_{ij} \leq R_1) = \frac{\pi R_1^2}{\alpha L^2}$$

$$P(R_1 \leq d_{ij} \leq R_2) = \frac{\pi (R_2^2 - R_1^2)}{\alpha L^2}$$

$$P(d_{ij} \geq R_2) = 1 - \frac{\pi R_2^2}{\alpha L^2}$$

For simplicity we replace  $d_{ij}$  with  $y$ . If we calculate  $P(C)$  from  $P(C, d_{ij})$

$$\begin{aligned}
P(C) &= \int P(C, y) dy \\
&= \int_0^{R_1} \frac{\pi R_1^2}{\alpha L^2} dy + \int_{R_1}^{R_2} \frac{\pi(R_2 + R_1)(R_2 - y)}{\alpha L^2} dy \\
&= \frac{\pi R_1^3}{\alpha L^2} + \frac{\pi(R_1 + R_2)}{\alpha L^2} \left( R_2^2 - \frac{R_2^2}{2} - R_1 R_2 + \frac{R_1^2}{2} \right) \\
&= \frac{\pi R_1^3}{\alpha L^2} + \frac{\pi(R_1 + R_2)(R_2 - R_1)^2}{2\alpha L^2}
\end{aligned}$$

Now the probability of having  $n$  neighbors near node  $i$  (having  $n$  nodes that have connection to node  $i$ , is a binomial distribution with  $p = P(C)$  and  $q = 1 - P(C)$ . Assume the binomial random variable is  $z$ . The expected value of our binomial distribution is equal to  $(N - 1)p$ , because if  $N$  is the total number of nodes in the network, a node can have a maximum of  $(N - 1)$  neighbors.

It means:

$$\mathbf{E}[z] = (N - 1) \left( \frac{\pi R_1^3}{\alpha L^2} + \frac{\pi(R_1 + R_2)(R_2 - R_1)^2}{2\alpha L^2} \right)$$

now if we know the average number of neighbors that the network nodes should have, the dimensions of the rectangular field are as follows:

$$Width = \sqrt{\frac{(N - 1)\pi}{\alpha \mathbf{E}[z]} \left( R_1^3 + \frac{(R_1 + R_2)}{2} (R_2 - R_1)^2 \right)}$$

$$Length = \alpha L$$

In our simulation we showed the average number of neighboring nodes (i.e.  $\mathbf{E}[z]$ ) with  $D$ .