Link scheduling and multi-path routing in wireless mesh networks

Fabio Rocha Jimenez Vieira

Universidade Federal do Rio de Janeiro
PESC COPPE

Université Pierre et Marie Curie
LIP6

École doctorale informatique, télécommunications et électronique de Paris

Thèse de doctorat

Discipline : Informatique

présentée par

Fabio Rocha Jimenez Vieira

Ordonnancement de liens et routage de multiples chemins pour les réseaux maillé sans fil

---

dirigée par José Ferreira de Rezende,
Valmir Carneiro Barbosa,
Serge Fdida

À soutenir le 25 mai 2012 devant le jury composé de :

M. Valmir C. Barbosa	Universidade Federal do Rio de Janeiro	président
M. José F. de Rezende	Universidade Federal do Rio de Janeiro	directeur
M. Serge Fdida	Université Pierre et Marie Curie	directeur
M. Edmundo A. de S. e Silva	Universidade Federal do Rio de Janeiro	rapporteur
M. Eric Fleury	Ecole Normale Supérieure de Lyon	rapporteur
M. Michel Minoux	Université Pierre et Marie Curie	examinateur

Lab. d'Informatique de Paris 6
Université Pierre et Marie Curie
4, Place Jussieu, 75252
Paris Cedex 05, France

Prog. Eng. S. Comp., COPPE
Univers. Fed. do Rio de Janeiro
Caixa Postal 68504, 21941-972
Rio de Janeiro - RJ, Brazil

Université Pierre et Marie Curie
École doctorale EDITE ED130
Maison de la pédagogie
4, place Jussieu, 75252
Paris Cedex 05, France

À minha mãe e ao meu pai pelo dom da vida e pelo amparo ao longo desses anos.
À minha esposa Juliana pelo amor.
Ao professor Fdida pela oportunidade.
Aos meus orientadores Rezende e Valmir por tudo, simplesmente tudo.

Le Loup et l'Agneau (La Fontaine).
Se a educação sozinha não transforma a sociedade, sem ela, tampouco, a sociedade muda (Paulo Freire).
Life is the art of drawing sufficient conclusions from insufficient premises (Samuel Butler).

Acknowledgments

We acknowledge partial support from CNPq, CAPES, a FAPERJ BBP grant, and a scholarship grant from Université Pierre et Marie Curie. Also, I specially thank Professor Otto Carlos Muniz Bandeira Duarte for include me in his COFECUB project. All computational experiments were carried out on the Grid'5000 experimental testbed, which is being developed under the INRIA ALADDIN development action with support from CNRS, RENATER, and several universities as well as other funding bodies (see `https://www.grid5000.fr`).

Abstract

Résumé

Nous présentons des solutions algorithmiques pour deux problèmes liés à l'interférence de réseau sans fil. D'abord on propose de ordonnancer les liens d'un ensemble de routes données en vertu de l'hypothèse d'un modèle à fort trafic. Nous considérons un protocole TDMA qu'offre une source d'intervalles de temps synchronisés et cherchent à ordonnancer les itinéraires des liens afin de maximiser le nombre de paquets qui sont livrés à leurs destinations par chaque intervalle de temps. Notre approche consiste à construire un graphe non orienté G et à obtenir multiples colorations pour les noeuds de G qui peuvent induire aux ordonnancement de liens efficaces. En G chaque noeud représente un lien à être ordonnancer et les arcs sont mis en place pour représenter toutes les interférences possibles pour un ensemble d'hypothèses d'interférence. Nous présentons deux heuristiques de multiples colorations et étudions leurs performances grâce à de nombreuses simulations. L'un des deux heuristiques est fondée sur l'assouplissement des dynamiques de multiples colorations en exploitant la disponibilité des possibilités de communication qui seraient autrement perdues. Nous avons constaté que, par conséquent, sa performance est nettement supérieure à la celle des autres. Dans la deuxième proposition, nous considérons les réseaux maillés sans fil et le problème de routage bout à bout du trafic sur les chemins multiples pour la même paire origine-destination avec un minimum d'interférences. Nous introduisons une heuristique pour la détermination des chemins avec deux caractéristiques distinctives. Tout d'abord, il fonctionne par le raffinage d'un ensemble existant de chemins, préalablement déterminée par un algorithme de routage de multiples chemins. Deuxièmement, il est tout à fait locale, dans le sens où il peut être exécuté par chacune des origines sur l'information qui est disponible plus loin dans le réseau de voisinage immédiat du noeud. Nous avons mené de nombreuses expériences avec la nouvelle heuristique, en utilisant le protocole OLSR et AODV ainsi que leurs variantes de chemins multiples. Nous avons démontré que la nouvelle heuristique est capable d'améliorer le débit moyen du réseau à l'échelle en utilisant un protocole TDMA sous l'exécution d'un algorithme de ordonnancement des liens orienté à routes et de deux différents paramètres de fonctionnement du protocole CSMA 802.11. En travaillent à partir des trajectoires générées par le chemin provenaient de algorithmes de multiples chemins, l'heuristique est également capable de fournir un modèle de trafic plus équitablement répartie.

Mots-clefs

Réseau maillé sans fil, Ordonnancement des liens, Dynamique de multiples coloration, Ordonnancement par la réversion d'arcs, Routage de multiples chemins, Accouplement des routes, Chemins disjoints, Interférence mutuelle

Link Scheduling and multi-path routing in wireless mesh networks

Abstract

We present algorithmic solutions for two problems related to the wireless network interference. The first one proposes to schedule the links of a given set of routes under the assumption of a heavy-traffic pattern. We assume some TDMA protocol provides a background of synchronized time slots and seek to schedule the routes' links to maximize the number of packets that get delivered to their destinations per time slot. Our approach is to construct an undirected graph G and to heuristically obtain node multicolorings for G that can be turned into efficient link schedules. In G each node represents a link to be scheduled and the edges are set up to represent every possible interference for any given set of interference assumptions. We present two multicoloring-based heuristics and study their performance through extensive simulations. One of the two heuristics is based on relaxing the notion of a node multicoloring by dynamically exploiting the availability of communication opportunities that would otherwise be wasted. We have found that, as a consequence, its performance is significantly superior to the other's. In the second proposal, we consider wireless mesh networks and the problem of routing end-to-end traffic over multiple paths for the same origin-destination pair with minimal interference. We introduce a heuristic for path determination with two distinguishing characteristics. First, it works by refining an extant set of paths, determined previously by a single- or multi-path routing algorithm. Second, it is totally local, in the sense that it can be run by each of the origins on information that is available no farther in the network than the node's immediate neighborhood. We have conducted extensive computational experiments with the new heuristic, using AODV and OLSR as well as their multi-path variants as the underlying routing method. For one TDMA setting running a path-oriented link scheduling algorithm and two different CSMA settings (as implemented on 802.11), we have demonstrated that the new heuristic is capable of improving the average throughput network-wide. When working from the paths generated by the multi-path routing algorithms, the heuristic is also capable to provide a more evenly distributed traffic pattern.

Keywords

Wireless mesh networks, Link scheduling, Node multicolorings, Scheduling by edge reversal, Multi-path routing, Route coupling, Disjoint paths, Mutual interference

Contents

Contents **9**

Introduction **11**

1 Interference in wireless transmissions **15**

1.1 Problem formulation 16

1.2 Graph transformation 19

1.3 Multicoloring-based schedules 22

2 Scheduling by edge reversal **27**

2.1 Improving on SER 29

3 SER and SERA experimentation **35**

3.1 Topology generation 35

3.2 Initial acyclic orientation 36

3.3 Computational results 37

3.4 Discussion 46

4 Interference among wireless routes **49**

4.1 Problem formulation 50

4.2 Selecting paths by the neighbors' independent set 52

4.3 Selecting paths in the absence of multiple routes 56

5 MRA experimentation **59**

5.1 Path discovery 59

5.2 Performance evaluation 61

5.3 Computational results 63
5.4 Discussion 75

6 Conclusion **77**
6.1 SER and SERA 77
6.2 MRA 80

A Supplemental data file 1 **83**

List of Figures **93**

Bibliography **95**

Introduction

Wireless mesh networks (WMNs) have lately been recognized as having great potential to provide the necessary networking infrastructure for communities and companies, as well as to help address the problem of providing last-mile connections to the Internet [NNS+07, SGP+07]. However, mutual radio interference among the network's nodes can easily reduce the throughput as network density grows above a certain threshold [BVB05] and therefore compromise the entire endeavor. Such interference is caused by the attempted concomitant communication among nodes of the same network and constitutes the most common cause of the network's throughput's falling short of being satisfactory (hardly reaching a fraction of that of a cabled network [GK00]). A promising approach to tackle the reduction of mutual interference seems to be to combine routing algorithms with some interference avoidance approach, such as power control, link scheduling, or the use of multi-channel radios [AWW05]. In fact, this type of network interference problem has been addressed by a considerable number of different strategies to be found in the literature [CQYM00, Abo04, CEM+08, SJA+10].

In this thesis, we addressed two wireless network problems, both related with the radio interference. The first problem is related to the interference among network links caused by the activation of theses links. To deal with it, we adopted a common solution to this problem, that is, we assume a contention-free TDMA protocol [DEA06] and we heuristically schedule simultaneous transmissions for activation only if they do not interfere with one another. However, we consider a variation of the problem, which is novel both in its formulation and in the solution type we propose. We start by assuming a heavily loaded network with pre-established set of origin-to-destination routes and whose access is controlled by a TDMA pro-

tocol, Also, each node has a limited buffer size to store network packets. Next, our algorithm schedule only links that belong to the set of routes, thus we try to maximize the throughput of these origin-to-destination routes. Our proposal was published in [VRBF12] and is present here in the following manner. Given the origin-to-destination routes (or paths) to be used, we begin in Chapter 1 with a precise definition of a schedule and a precise formulation of the problem. We also show, through an example, that had the problem been formulated for network-capacity maximization, a conflict with the requirement of finite buffering might arise. Then we move to the specification of the undirected graph that underlies our algorithm's operation. One assumption throughout most of our work, is that the communication and interference radii are the same for the WMN at hand. Moreover, we also assume that the tenets of the protocol-based interference model [BR03, SHLK09], including the possibility of bidirectional communication in each transmission, are in effect. Next, we guide the reader through various multicoloring possibilities, which culminates in Chapter 2 with a preliminary method for scheduling, borrowed from the field of resource sharing [Bar96]. Improving on this preliminary method with the goals of the problem formulated in Chapter 1 in mind finally yields our proposal in Section 2.1. This proposal, essentially, stems from a slight relaxation of the notion of a node multicoloring. The subsequent two sections are dedicated to the presentation of computational results, with the methodology and the results laid down in Chapter 3. Discussion follows in Section 3.4 and we close in the first part of Chapter 6.

In the second problem, we deal with the same interference among link, but from the point of view of the network paths. Considering that, an alternative approach that presents itself naturally is the use of multi-path routing to distribute traffic among multiple paths sharing the same origin and the same destination, since in principle it can help to improve both path recovery and load balance better than single-path strategies. It may, in addition, lead to better throughput values over the entire network [KLF06, ACSR10]. But while these benefits accrue only insofar as they relate to how the multiple paths interfere with one another [TM06, TTAE09], unfortunately this aspect of the problem is not commonly addressed by multi-path strategies. Here we propose a different approach to alleviate the effects of interfer-

ence in multi-path routing. Our approach is based on two general principles. First, that it is to work as a refinement phase over existing routing algorithms, thereby inherently preserving, to the fullest possible extent, the advantages of any given routing method. Second, that it is to rely only on information that is locally available to the common origin of any given set of multiple paths leading to the same destination. That is, only information that the origin can obtain by communicating with its direct neighbors in the WMN should be used. This proposal was submitted to The Computer Networks Journal in march 2012 and is present here as follows. Given the a multi-path set to be selected, we explain in Chapter 4 the problem formulation of selecting non-interfering paths. We also show, through examples, that the classical disjoint sets used in the related works are not interference free. Then we present, in Section 4.2 our proposal for multi-path routing algorithms and in Section 4.3 a small modification that able our algorithm to work with single-path algorithms. Chapter 5 is dedicated to explain our evaluation method and the computational results. We compared our results with some of the most important routing algorithms, such as AODV [PR99], AOMDV [MD02], OLSR [JMC$^+$01] and MP-OLSR [YADP11]. We used our approach to alter the path set of these algorithms and we measured their throughput and fairness [JCH84] against their original path sets. For that, we carried out extensive experimentation by using the network simulator NS2.34 [ns289] and the SERA scheduling algorithm [VRBF12]. We follow in Section 5.4 with the discussion of our improvements, and finally, we conclude in the second part of Chapter 6.

Chapter 1

Interference in wireless transmissions

Owing to their numerous advantages, wireless mesh networks (WMNs) constitute a promising solution for community networks and for providing last-mile connections to Internet users [AWW05, BVB05, NNS+07]. However, like all wireless networks WMNs suffer from the problem of decreased capacity as they become denser, since in this case attempting simultaneous transmissions causes interference to increase significantly [GK00, SGP+07]. One common solution to reduce interference is to adopt some contention-free TDMA protocol [DEA06] and schedule simultaneous transmissions for activation only if they do not interfere with one another. Doing this while maximizing some measure of network usage and guaranteeing that all links are given a fair treatment normally translates into a complicated optimization problem, one that unfortunately is NP-hard [BBK+04].

This scheduling problem has been formulated in a great variety of manners and has received considerable attention in the literature. Prominent studies include some that seek to calculate the capacity of the network [GK00, GWHW09], others whose goal is the study of the time complexity associated with the resulting schedules [MWZ06], and still others that aim at scheduling transmissions in order to achieve as much of the network's capacity as possible [CS03, ABL06, WWL+06, GDP08, HL08, WDJ+08, SMR+09, XT09, WGLA09]. One common thread through most the latter is that, having adopted a graph representation of the network and of

how the various transmissions can interfere with one another, a solution is sought through some form of graph coloring. More often than not the transmissions to be scheduled are represented by the graph's nodes and then node coloring, through the abstraction of an independent set to represent the transmissions that can take place simultaneously, is used. But sometimes it is the graph's edges that stand for transmissions, in which case edge coloring is used, building on the abstraction of matchings to represent simultaneity [BM08].

Here we consider a variation of the problem which, to the best of our knowledge, is novel both in its formulation and in the solution type we propose. We start by assuming a WMN comprising single-channel, single-radio nodes and for which a set of origin-to-destination routes has already been determined, and consider the following question. Should there be an infinite supply of packets at each origin to be delivered to the corresponding destination in the FIFO order, and should all nodes in the network be endowed with only a finite number of buffers for the temporary storage of in-transit packets, how can transmissions be scheduled to maximize the number of packets that get delivered to the destinations per TDMA slot without ever stalling a transmission, by lack of buffering space, whenever it is scheduled? This question addresses issues that lie at the core of successfully designing WMNs and their routing protocols, since it seeks to tackle the problem of transmission interference when the network is maximally strained. The solution we propose is, like in so many of the approaches mentioned above, based on coloring a graph's nodes. Unlike them, however, we use node multicolorings instead [Sta76], which are more general and for this reason allow for a more suitable formulation of the optimization problem to be solved.

1.1 Problem formulation

We consider a collection $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$ of simple directed paths (i.e., directed paths that visit no node twice), each having at least two nodes (a source and a destination). These paths' sets of nodes are $X_1, X_2, \ldots, X_P$, respectively, not necessarily disjoint from one another, and we let $X = \bigcup_{p=1}^{P} X_p$. Their sets of edges are $Y_1, Y_2, \ldots, Y_P$ and we assume that, for $p \neq q$, a member of Y_p and one of Y_q are distinguishable

from each other even if they join the same two nodes in the same direction. Letting $Y = \bigcup_{p=1}^{P} Y_p$, we then see that Y may contain more than one edge joining the same two nodes in the same direction (parallel edges) or in opposing directions (antiparallel edges).

Our discussion begins with the definition of the directed multigraph $D = (X, Y)$, where all P directed paths are represented without sharing any directed edges among them. An example is shown in Figure 1.1. We take D to be representative of a wireless network operating under some TDMA protocol. In this network, each of paths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$ is to transmit an unbounded sequence of packets from its source to its destination. Such transmissions are to occur without contention, meaning that whenever an edge is scheduled to transmit in a given time slot no other edge that can possibly interfere with that transmission is to be scheduled at the same time slot. We assume that each transmission sends at most one packet across the edge in question (more specifically, it sends exactly one packet if there is at least one to be sent but does nothing otherwise). We also assume that each transmission may involve the need for bidirectional communication for error control.

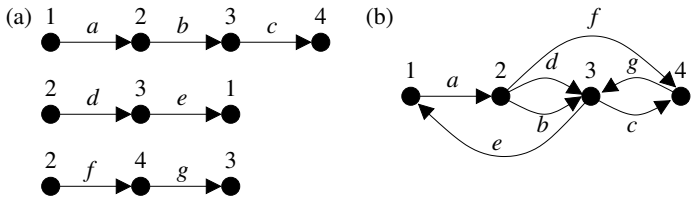

Figure 1.1: A set of $P = 3$ directed paths (a) and the resulting directed multigraph D (b).

We call a schedule any finite sequence $\mathcal{S} = \langle S_0, S_1, \ldots, S_{L-1} \rangle$ such that $S_\ell \subseteq Y$ for $0 \leq \ell \leq L-1$, provided $\bigcup_{\ell=0}^{L-1} S_\ell = Y$ and moreover no two concurrent transmissions on edges of the same S_ℓ can interfere with each other. To schedule the transmissions according to $\mathcal{S}$ is to cycle through the edge sets $S_0, S_1, \ldots, S_{L-1}$, indefinitely and in this order, letting all edges in the same set transmit in the same time slot whenever that set is reached along the cycling. Given $\mathcal{S}$, we let $\text{length}(\mathcal{S}) = L$ and denote by $\text{delivered}(\mathcal{S})$ the number of packets that can get delivered to all paths' destinations during a single repetition of $\mathcal{S}$ in the long run (i.e., in the limit as the number of repetitions grows without bound). Of course, $\text{delivered}(\mathcal{S})$ is bounded from above by the number of times the P paths' terminal edges (those leading directly to a

destination node) appear in $\mathcal{S}$ altogether.

Before we use these two quantities to define the optimization problem of finding a suitable schedule for D, we must recognize that our focus on the source-to-destination packet flows on the paths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$ carries with it the inherent constraint that the nodes' capacity to buffer in-transit packets cannot be allowed to grow unbounded. We then adopt an upper bound B on the number of in-transit packets that a node can store for each of the paths (at most P) that go through it. However, there is still a decision to be made regarding the effect of such a bound on the transmission of packets. One possibility would be to impose that, when it is an edge's turn to transmit it does so if and only if there is a packet to transmit and, moreover, there is room to store that packet if it is received as an in-transit packet. Another possibility, one that seeks to never stall a transmission by lack of a buffer to store the packet at the next intermediate node, is to only admit schedules that automatically rule out the occurrence of such a transmission. We adopt the latter alternative.

The following, then, is how we formulate our scheduling problem on D. Find a schedule $\mathcal{S}$ that maximizes the throughput

$$T(\mathcal{S}) = \frac{\text{delivered}(\mathcal{S})}{\text{length}(\mathcal{S})}, \tag{1.1}$$

subject to the following two constraints:

C1. Every node can store up to B in-transit packets for each of the source-to-destination paths that go through it.

C2. Whenever an edge is scheduled for transmission in a time slot and a packet is available to be transmitted, if the edge is not the last one on its source-to-destination path then there has to be room for the packet to be stored after it is transmitted.

1.1.1 Scheduling for maximum network usage

Before proceeding, recall that, as mentioned in the beginning of this chapter, the most commonly solved problem regarding the selection of a schedule $\mathcal{S}$ is not the

one we just posed, but rather the problem of maximizing network usage. In terms of our notation, this problem requires that we find a schedule that maximizes

$$U(\mathcal{S}) = \frac{\sum_{\ell=0}^{L-1} |S_\ell|}{\text{length}(\mathcal{S})} \tag{1.2}$$

without any constraints other than those that already participate in the definition of a schedule.

It is a simple matter to verify that solutions to this problem often fail to respect constraints C1 and C2 of our formulation. This is exemplified in Figure 1.2.

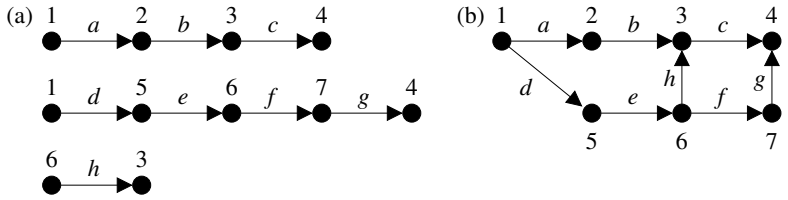

Figure 1.2: A set of $P = 3$ directed paths (a) and the resulting directed multigraph D (b). Using the schedule $\mathcal{S}$ such that $S_0 = \{a, f\}$, $S_1 = \{c, d\}$, $S_2 = \{b\}$, $S_3 = \{e\}$, $S_4 = \{a, g\}$, and $S_5 = \{h\}$ causes unbounded packet accumulation at node 2 when constraint C2 is in effect, thus violating constraint C1. Enforcing constraint C1 for some value of B causes constraint C2 to be violated.

1.2 Graph transformation

We wish to address the problem of optimizing $T(\mathcal{S})$ exclusively in terms of some underlying graph. Clearly, though, the directed multigraph D is not a good candidate for this, since it does not embody any representation of how concurrent transmissions on its edges can interfere with one another. Our first step is then to transform D into some more suitable entity, which will be the undirected graph $G = (N, E)$ defined as follows:

1. The node set N of G is the edge set Y of D. In other words, G has a node for every edge of D. Since D is a multigraph, a same pair of nodes $i, j \in X$ such that $(i, j) \in Y$ or $(j, i) \in Y$ may appear more than once as a member of N.

2. The edge set E of G is obtained along the following four steps:

   i. Enlarge N by including in it all node pairs of D that do not correspond to edges on any of the P source-to-destination paths but nevertheless reflect

that each node in the pair is within the interference radius of the other. We refer to these extra members of N as temporary nodes.

ii. Connect any two nodes in N by an edge if, when regarded as node pairs from D, they share at least one of the nodes of D. In other words, if each of the two pairs $i, j \in X$ and $k, l \in X$ corresponds to a node of G (by virtue of either constituting an edge of D or being a temporary node), then the two get connected by an edge in G if at least one of $i = k$, $i = l$, $j = k$, or $j = l$ holds.

iii. Connect any two nodes in N by an edge if, after the previous steps, the distance between them is 2.

iv. Eliminate all temporary nodes from N and all edges from E that touch them.

Together, these four steps amount to using G to represent every possible interference that may arise under the assumptions of the protocol-based model when communication is bidirectional. Graph G is also known as a distance-2 graph relative to D [BBK+04]. The entire transformation process, from the set of P paths through graph G, is illustrated in Figure 1.3.

It follows from this definition of G that any group of nodes corresponding to parallel or antiparallel edges in D are a clique (a completely connected subgraph) of G. Similarly, every group of three consecutive edges on any of the paths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$ corresponds to a three-node clique in G. As we discuss in Section 3.4, these and other cliques are related to how large $T(\mathcal{S})$ can be under one of the scheduling methods we introduce.

It is also worth noting that Steps 1 and 2 above are easily adaptable to modifications in any of the assumptions we made. These include the assumptions that the communication and interference radii are the same and that communication is bidirectional. Changing assumptions would simply require us to adapt Steps 2.i through 2.iii accordingly.

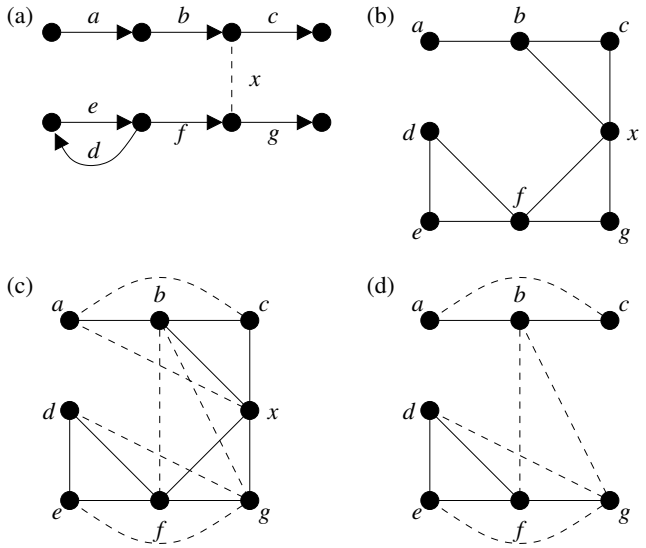

Figure 1.3: The graph-transformation process. We start with the directed multigraph D (a), to which the node pair labeled x is added as a dashed line to indicate the existence of interference that is not internal to any of the initial P paths. Panel (b) contains the undirected graph G as it stands after Step 2.ii. This stage is reached by creating a node for every directed edge in panel (a) (through Step 1) and a node for every undirected edge in panel (a) (through Step 2.i). Note that the latter results in the temporary node x of panel (b). In addition to node creation, reaching panel (b) from (a) requires the creation, through Step 2.ii, of undirected edges to join nodes in (b) whenever in (a) the corresponding edges, directed or otherwise, have a node in common. Panel (c) shows G past Step 2.iii, through which the edge set of G is enlarged by connecting any two nodes that in (b) are two edges apart. These extra edges are shown in dashed lines. Panel (d), finally, results from applying Step 2.iv to the graph in panel (c). This results in the removal of temporary node x, along with all its adjacent edges.

1.3 Multicoloring-based schedules

Graph G allows us to rephrase the definition of a schedule as follows. We call a schedule any finite sequence $\mathcal{S} = \langle S_0, S_1, \ldots, S_{L-1} \rangle$ such that $S_\ell \subseteq N$ for $0 \leq \ell \leq L-1$, provided $\bigcup_{\ell=0}^{L-1} S_\ell = N$ and moreover every S_ℓ is an independent set of G. The appearance of the notion of an independent set in this definition leads the way to a special class of schedules, namely those that can be identified with graph multicolorings [Sta76].

For $q \geq 1$, a q-coloring of the nodes of G is a mapping from N, the graph's set of nodes, to $\mathbb{N}^q$, where $\mathbb{N}$ is the set of natural numbers, such that no two of a node's q colors are the same and besides none of them coincides with any one of any neighbor's q colors. Of course, the set of nodes receiving one particular color is an independent set. If p is the total number of colors needed to provide G with a q-coloring, then N is covered by the p independent sets that correspond to colors and every node is a member of exactly q of these sets. Therefore, letting $L = p$ and identifying each S_ℓ with the set of nodes receiving color ℓ implies that to every q-coloring of the nodes of G there corresponds a schedule $\mathcal{S}$.

These multicoloring-derived schedules constitute a special case in the sense that every node of G can be found in exactly the same number of sets (q) out of the L sets that make up the schedule. Clearly, though, there are schedules that do not correspond to multicolorings. For now we concentrate on those that do and note that $\text{delivered}(\mathcal{S}) \leq Pq$ always holds (recall that P stands for the number of origin-to-destination paths). That is, the greatest number of packets that the P terminal edges of D can deliver during the L time slots of schedule $\mathcal{S}$ is q per terminal edge. These schedules can be further specialized, as follows.

1.3.1 Standard coloring

When $q = 1$ every node of G receives exactly one color and $\text{length}(\mathcal{S}) = L \geq \chi(G)$, where $\chi(G)$ is the least number of colors with which it is possible to provide G with a 1-coloring, known as the chromatic number of G. Using $T^1(\mathcal{S})$ to denote $T(\mathcal{S})$ in this case, we have

$$T^1(\mathcal{S}) \leq \frac{P}{\chi(G)}. \tag{1.3}$$

1.3.2 Standard multicoloring

Coloring G's nodes optimally in the previous case is minimizing the overall number of colors. This stems not only from the fact that $q = 1$, but more generally from the fact that q is fixed. We can then generalize and define $\chi^q(G)$ to be the least number of colors with which it is possible to provide G with a q-coloring. Evidently, $\chi(G) = \chi^1(G) < \chi^2(G) < \cdots$, so the question of multicoloring G's nodes optimally when q is not fixed can no longer be viewed as that of minimizing the overall number of colors needed (as this would readily lead to $q = 1$ and $\chi(G)$ colors). Instead, we look at how efficiently the overall number of colors is used, i.e., at what the value of q has to be so that $\chi^q(G)/q$ is minimized. This gives rise to the multichromatic number of G, denoted by $\chi^*(G)$ and given by $\chi^*(G) = \inf_{q\geq 1} \chi^q(G)/q$. Because this infimum can be shown to be always attained, we use minimum instead and let q^* be the value of q for which $\chi^*(G) = \chi^{q^*}(G)/q^*$.

Using a q-coloring for scheduling amounts to having length$(\mathcal{S}) = L \geq \chi^q(G)$. In this case, letting $T^*(\mathcal{S})$ stand for $T(\mathcal{S})$ yields

$$T^*(\mathcal{S}) \leq \frac{Pq}{\chi^q(G)} \leq \frac{Pq^*}{\chi^{q^*}(G)} = \frac{P}{\chi^*(G)}. \tag{1.4}$$

1.3.3 Interleaved multicoloring

A special class of q-colorings is what we call interleaved q-colorings [BG89, Bar00b, YZ05]. If i and j are two neighboring nodes of G, let $c_1^i < c_2^i < \cdots < c_q^i$ be the q colors assigned to node i by some q-coloring, and likewise let $c_1^j < c_2^j < \cdots < c_q^j$ be those of node j. We say that this q-coloring is interleaved if and only if either $c_1^i < c_1^j < c_2^i < c_2^j < \cdots < c_q^i < c_q^j$ or $c_1^j < c_1^i < c_2^j < c_2^i < \cdots < c_q^j < c_q^i$ for all neighbors i and j. If we restrict ourselves to interleaved q-colorings, then similarly to what we did above we use $\chi^q_{\text{int}}(G)$ to denote the least number of colors with which it is possible to provide G with an interleaved q-coloring, and similarly define the interleaved multichromatic number of G, denoted by $\chi^*_{\text{int}}(G)$, to be $\chi^*_{\text{int}}(G) = \inf_{q\geq 1} \chi^q_{\text{int}}(G)/q$. Once again it is always possible to attain the infimum, so we may take q^* to be the value of q for which $\chi^*_{\text{int}}(G) = \chi^{q^*}_{\text{int}}(G)/q^*$.

As for scheduling based on an interleaved q-coloring, it corresponds to having

length($\mathcal{S}$) = $L \geq \chi^q_{\text{int}}(G)$. As before, we use $T^*_{\text{int}}(\mathcal{S})$ in lieu of $T(\mathcal{S})$ and obtain

$$T^*_{\text{int}}(\mathcal{S}) \leq \frac{Pq}{\chi^q_{\text{int}}(G)} \leq \frac{Pq^*}{\chi^{q^*}_{\text{int}}(G)} = \frac{P}{\chi^*_{\text{int}}(G)}. \tag{1.5}$$

1.3.4 Discussion

It is a well-known fact that

$$\frac{1}{\chi(G)} \leq \frac{1}{\chi^*_{\text{int}}(G)} \leq \frac{1}{\chi^*(G)}. \tag{1.6}$$

The first inequality follows from the definition of $\chi^*_{\text{int}}(G)$, considering that every 1-coloring is (trivially) interleaved. As for the second inequality, it follows directly from the definition of $\chi^*(G)$. By these inequalities, should all of Eqs. (1.3)–(1.5) hold with equalities, we would have

$$T^1(\mathcal{S}) \leq T^*_{\text{int}}(\mathcal{S}) \leq T^*(\mathcal{S}). \tag{1.7}$$

Obtaining equalities in Equations (1.3), (1.4) and (1.5), however, requires both that delivered($\mathcal{S}$) = Pq for $q = 1$ or $q = q^*$, as the case may be, and that length($\mathcal{S}$) = $\chi^q(G)$ with the same possibilities for q or length($\mathcal{S}$) = $\chi^{q^*}_{\text{int}}(G)$.

While the combined requirements involve the exact solution of NP-hard problems (finding any of $\chi(G)$, $\chi^*_{\text{int}}(G)$, and $\chi^*(G)$ is NP-hard; cf., respectively, [Kar72], [BG89], and [GLS81]), the former requirement alone (that delivered($\mathcal{S}$) = Pq) is always a property of schedules based on multicolorings when buffering availability is unbounded. To see that this is so, first recall that the definition of delivered($\mathcal{S}$) refers to a repetition of the whole schedule as far down in time as needed for any transient effects to have waned. So, given any of the P source-to-destination paths, we can prove that delivered($\mathcal{S}$) = Pq by arguing inductively about what happens on such a path during that future repetition of $\mathcal{S}$. The basis case in this induction is the first directed edge on the path (the one leading out of the source). The property that every appearance of this edge does indeed transmit a packet follows trivially from the fact that the source has an endless supply of new packets to provide whenever needed. Assuming that this also happens to the next-to-last edge on the path (this is our induction hypothesis) immediately leads to the same conclusion regarding the

last edge, the one on which delivered($\mathcal{S}$) is defined. To see this, let e be the last edge and e^- the next-to-last one. Because $\mathcal{S}$ is repeated indefinitely, every time slot t sufficiently far down in time in which e appears is the closing time slot of a window in which both e and e^- appear exactly q times each. By the induction hypothesis, it follows that at least one packet is guaranteed to exist for transmission through e at time slot t.

Buffering availability, however, is not unbounded, so we must argue for its finiteness. We do this by recognizing another important property of multicoloring-based schedules, one that is related to constraints C1 and C2 introduced earlier. Because every edge of D (node of G) appears the exact same number q of times in $\mathcal{S}$, there certainly always is a finite value of B, the number of buffer positions per node per path that goes through it, such that C1 and C2 are satisfied. In all interleaved cases, this value is $B = 1$.

An example illustrating all of this is presented in Figure 1.4, where we give a set of four source-to-destination paths, the graph G that eventually results from them, and also the three schedules that result in equalities in Eqs. (1.3)–(1.5). In this case the two inequalities in Eq. (1.6) are strict, since it can be shown that $\chi(G) = 3$, $\chi^*_{\text{int}}(G) = 8/3$, and $\chi^*(G) = 5/2$ [Sta76, BG89].

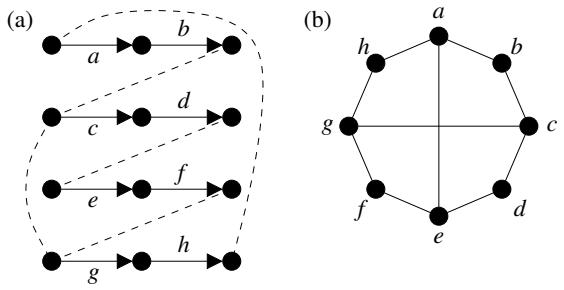

Figure 1.4: A set of $P = 4$ paths (a), with dashed lines indicating all node pairs representing off-path interference. The resulting graph G is shown in panel (b). Depending on the schedule $\mathcal{S}$ it is possible to obtain equalities in all of Eqs. (1.3)–(1.5). The schedules that achieve this while implying strict inequalities in Eq. (1.6) are: $\mathcal{S} = \langle\{a, d, g\}, \{b, f, h\}, \{c, e\}\rangle$ for Eq. (1.3), with $T^1(\mathcal{S}) = 4/3 \approx 1.33$; $\mathcal{S} = \langle\{a, d, f\}, \{b, e, g\}, \{c, f, h\}, \{a, d, g\}, \{b, e, h\}, \{a, c, f\}, \{b, d, g\}, \{c, e, h\}\rangle$ for Eq. (1.5), with $T^*_{\text{int}}(\mathcal{S}) = 4/(8/3) = 1.5$; and $\mathcal{S} = \langle\{a, c, f\}, \{b, e, g\}, \{c, e, h\}, \{a, d, g\}, \{b, d, f, h\}\rangle$ for Eq. (1.4), with $T^*(\mathcal{S}) = 4/(5/2) = 1.6$.

Chapter 2

Scheduling by edge reversal

From the three schedules illustrated in Figure 1.4 it would seem that finding a schedule $\mathcal{S}$ to maximize $T(\mathcal{S})$ requires that we give up on the interleaved character of the underlying multicoloring and, along with it, give up on the equivalent property that edges of D that are consecutive on some source-to-destination path appear in $\mathcal{S}$ alternately. However, once color interleaving is assumed we are automatically provided with a principled way to heuristically try and maximize $T(\mathcal{S})$ by appealing to a curious relationship that exists between multicolorings and the acyclic orientations of G. We now review this heuristic and later build on it by showing how to adapt it to abandon interleaving only on occasion during a schedule, aiming at maximizing $T(\mathcal{S})$.

An orientation of G is an assignment of directions to its edges. An orientation of G is acyclic if no directed cycles are formed. Every acyclic orientation has a set of sinks (nodes with no edges oriented outward), which by definition are not neighbors of one another. An acyclic orientation's set of sinks is then an independent set. The heuristic we now describe, known as scheduling by edge reversal (SER) [BG89, Bar96], is based on the following property. Should an acyclic orientation be transformed into another by turning all its sinks into sources (nodes with no edges oriented inward), and should this be repeated indefinitely, we would obtain an infinite sequence of independent sets, each given by the set of sinks of the current orientation. Though infinite, this sequence must necessarily reach a point from which a certain number of acyclic orientations gets repeated indefinitely. This follows from the facts that there are only finitely many acyclic orientations of G and that turning

one of them into the next is a deterministic process.

The orientations that participate in this cyclic repetition, henceforth called a period, have the property that every node of G appears as a sink in the same number of orientations. Furthermore, any two neighboring nodes of G are sinks in alternating orientations, regardless of whether the period has already been reached or not. It clearly follows that the sets of sinks in a period constitute a schedule that is based on an interleaved multicoloring. Depending on the very first acyclic orientation in the operation of SER more than one period can eventually be reached. The different periods' properties vary from one to another, but it can be shown that at least one of them corresponds to the optimal interleaved multicoloring, i.e., the one that yields $\chi^*_{\text{int}}(G)$ [BG89]. The heuristic nature of SER is then revealed by the need to determine an appropriate initial acyclic orientation.

Determining a schedule $\mathcal{S}$ by SER follows the algorithm given next, whose functioning is illustrated in Figure 2.1 for a simple G instance. We use $\omega_0, \omega_1, \ldots$ to denote the sequence of acyclic orientations of G. For $t = 0, 1, \ldots$, we denote by $\text{Sinks}(\omega_t)$ the set of sinks of ω_t.

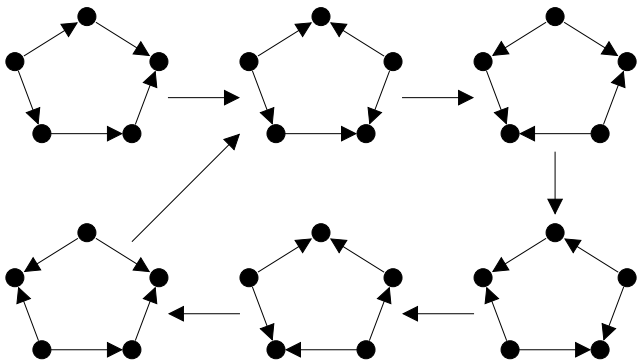

Figure 2.1: The functioning of SER when G is the 5-node cycle and ω_0 is the leftmost orientation in the top row. The period that is reached from ω_0 has $p(\omega_0) = 5$ and $m(\omega_0) = 2$.

ALGORITHM SER:

1. Choose ω_0.

2. $t := 0$.

3. Obtain ω_{t+1} from ω_t.

4. If the period has not yet occurred, then $t := t + 1$; go to Step 3. If it has, then let $p(\omega_0)$ be its number of orientations, $m(\omega_0)$ the number of times any node appears in them as a sink, and $\omega_k, \omega_{k+1}, \ldots, \omega_{k+p(\omega_0)-1}$ the orientations themselves. Output

$$\mathcal{S} = \langle \text{Sinks}(\omega_k), \text{Sinks}(\omega_{k+1}), \ldots, \text{Sinks}(\omega_{k+p(\omega_0)-1}) \rangle$$

and

$$T(\mathcal{S}) = \frac{Pm(\omega_0)}{p(\omega_0)}.$$

In this algorithm, the explicit dependency of both $p(\omega_0)$ and $m(\omega_0)$ on ω_0 is meant to emphasize that, implicitly, the two quantities are already determined when in Step 1 the initial orientation ω_0 is chosen. As with the very existence of the period, this follows from the fact that the algorithm's Step 3 is deterministic, so there really is no choice regarding the period to be reached once ω_0 has been fixed. The role played by the two quantities is precisely to characterize the interleaved multicoloring mentioned above. That is, the period reached from ω_0 can be regarded as assigning $q = m(\omega_0)$ distinct colors to each node of G using a total number $p = p(\omega_0)$ of colors. Equivalently, it can be regarded as a schedule $\mathcal{S}$ for which delivered$(\mathcal{S}) = Pq = Pm(\omega_0)$ (where the first equality is true of all multicoloring-based schedules, as we discussed in Section 1.3) and length$(\mathcal{S}) = L = p = p(\omega_0)$. By Eq. (1.1), the final determination of $T(\mathcal{S})$ follows easily.

As a final observation, we note that, although the knowledge of $p(\omega_0)$ and $m(\omega_0)$ after Step 1 is only implicit, it can be shown that the ratio $m(\omega_0)/p(\omega_0)$ can be known explicitly at that point [BG89]. It might then seem that the remainder of the algorithm is useless, since the value of $T(\mathcal{S})$ can be calculated right after Step 1. But the reason why the remaining steps are needed, of course, is that $\mathcal{S}$ itself needs to be found, not just the $T(\mathcal{S})$ that quantifies its performance.

2.1 Improving on SER

In Figure 2.2 we provide an example to illustrate why giving up interleaving may yield a schedule $\mathcal{S}$ of higher $T(\mathcal{S})$. The general idea is that, given B, it may be

possible to schedule a given transmission sooner than it normally would be scheduled by SER, provided there is a packet to be transmitted in the buffers of the sending node in D and the receiving node has an available buffer position for the path in question. While under SER two transmissions sharing a buffer alternate with each other in any schedule (and then $B = 1$ always suffices), disrupting this alternance implies that all buffering is to be managed in detail.

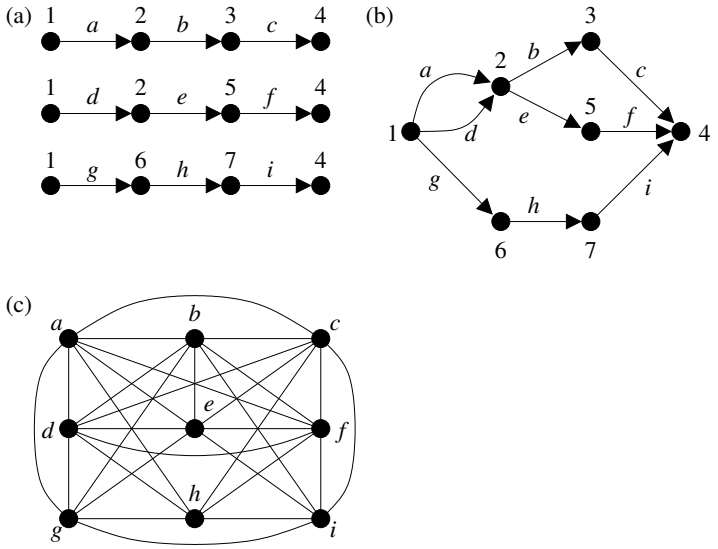

Figure 2.2: A set of $P = 3$ directed paths (a), the resulting directed multigraph D (b), and the resulting undirected graph G (c). The optimal SER schedule is $\mathcal{S} = \langle\{a\}, \{b\}, \{c, g\}, \{d, i\}, \{e, h\}, \{f\}\rangle$, yielding $T(\mathcal{S}) = 3/6 = 0.5$. An alternative schedule that does not comply with the SER alternance condition, with $B = 2$, is $\mathcal{S} = \langle\{g, c\}, \{g, f\}, \{h, b\}, \{h, e\}, \{i, a\}, \{i, d\}\rangle$, which results in an improvement to $T(\mathcal{S}) = 4/6 \approx 0.67$.

In the example of Figure 2.2 transmissions g, h, and i are scheduled without regard to alternance if $B = 2$. While this results in improved performance (more packets delivered to node 4 per time slot), it is important to realize that this is in great part made possible by the structure of D in this example. Even though all three paths lead from node 1 to node 4, two of them are poised to interfere with each other particularly heavily by virtue of sharing node 2. The consequence of this is that transmissions on these paths will occur less in parallel than they might otherwise. But since $B = 2$ buffering positions are available per node per path, the path that goes through nodes 6 and 7 can compensate for this by transmitting twice as much traffic (thence the double occurrence of g in a row, and also of h and i, for each repetition of the schedule). This, however, is never detrimental to the traffic

on the other two paths: all that is being done is to seize the opportunity to transmit in time slots that would otherwise go unused.

Implementing the careful buffer management alluded to above requires that we look at the dynamics of acyclic-orientation transformation under SER in more detail. Given any acyclic orientation ω of G, the node set N of G can be partitioned into independent sets $I_1, I_2, \ldots, I_d$ such that I_1 is the set of all sinks in G according to ω, I_2 is the set of all sinks we would obtain if all nodes in I_1 were to be eliminated, and so on. In this partition, known as a sink decomposition, d is the number of nodes on a longest directed path of G according to ω. When ω is turned into ω' by SER a new sink decomposition is obtained, call it $I'_1, I'_2, \ldots, I'_{d'}$, such that $I'_1 = I_2$, $I'_2 \supseteq I_3$, etc., with $d' \leq d$. As we illustrate shortly, the reason why equality need not hold in all cases, but set containment instead, is that each I_k may get enlarged by some of the previous orientation's sinks before becoming I'_{k-1}.

It is then possible to regard the operation of SER as simply a recipe for manipulating sink decompositions. At each iteration the set containing the current sinks is eliminated and its nodes are redistributed through the other sets. The remaining sets are renumbered by decrementing their subscripts by 1 and a new, greatest-subscript set may have to be created. The rule for redistributing each of the former sinks is to find the set of greatest subscript containing one of the node's neighbors in G, say I_k, and then place the node in set I_{k+1}. This is illustrated in panels (a) and (b) of Figure 2.3. From panel (a) we have $I_1 = \{a, b\}$, $I_2 = \{c\}$, $I_3 = \{d\}$, and so on, and from panel (b) we have $I'_1 = \{c\}$, $I'_2 = \{b, d\}$, etc. Clearly, then, $I'_1 = I_2$ while $I'_2 \supset I_3$. In the latter case we may think of I_3 as being enlarged by node b, a sink from I_1, and then becoming I'_2.

Altering this redistribution rule is the core of our modified SER, henceforth called SER with advancement (SERA). If i is the sink in question, the operation of SERA is based on placing node i in the least-subscript set that does not contain a neighbor of i in G. This clearly maintains acyclicity just as the previous rule does, but now the former sink is not necessarily turned into a source, but rather into a node that can have edges oriented both inward and outward by the current orientation, respectively from nodes in sets of greater subscripts and to nodes in sets of lesser subscripts. Additionally, while this alternative placement of node i does favor it by

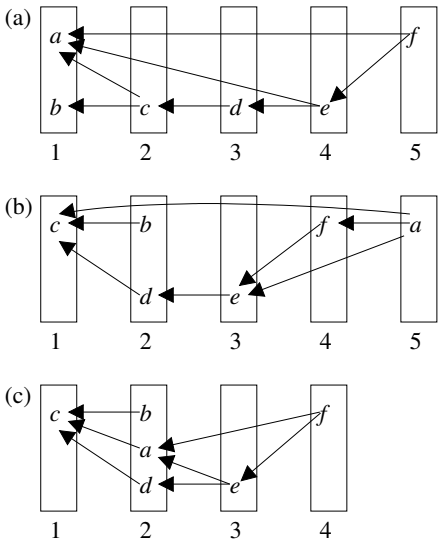

Figure 2.3: Each set in a sink decomposition is represented by a rectangular box and numbered to indicate the set's subscript. Note that directed edges refer to acyclic orientations of G. Applying SER to the sink decomposition in panel (a) yields the one in panel (b). In this transformation both a and b are turned into sources. The alternative of using SERA, on the other hand, makes it possible for a to be placed in a lower-subscript set, avoiding the transformation into a source and yielding the sink decomposition in panel (c). This can be done only because the set to which a is added contains none of its neighbors in G. Assuming that transmissions a, e, and f are initially arranged consecutively in one of the P paths in the order e, a, f, we have $i^- = e$, $i = a$, and $i^+ = f$. We also have, in reference to panel (a), $k^- = 4$, $k = 2$, and $k^+ = 5$. Then the additional conditions for the move from (a) to (c) to occur are that the buffers shared by transmissions e and a contain at least one packet (since $k < k^-$), and that those shared by a and f have room for at least one packet (since $k < k^+$).

virtue of lowering the number of time slots that need to go by before it is a sink once again, clearly there is no detriment to any of the other transmissions, which will assuredly become sinks no later than they would otherwise.

As we mentioned, however, unlike SER this rule can only be applied as a function of B and the buffering-related constraints we mentioned. Suppose that i is preceded by transmission i^- and succeeded by transmission i^+ on the original path out of $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$ to which it belongs. Of course, both i^- and i^+ are nodes of G as well. Suppose further that these two nodes are in sets I_{k^-} and I_{k^+}, respectively, and that we are attempting to place node i in set I_k. The further constraints to be satisfied are the following. If $k < k^-$, then the buffers shared by transmissions i^- and i must contain at least one packet to be transmitted. If $k < k^+$, then the buffers shared by transmissions i and i^+ must contain room to store at least one more packet. This can all be implemented rather easily by keeping a dynamic record of all buffers. A simple case of evolving sink decompositions in the SERA style is shown in panels (a) and (c) of Figure 2.3.

SERA, like SER, operates on finitely many possibilities and deterministically. A "possibility" is no longer simply an acyclic orientation, but instead an acyclic orientation together with a configuration of buffer occupation. In any event, periodic behavior is still guaranteed to occur and we go on denoting by $p(\omega_0)$ the number of possibilities in the period that one reaches from ω_0. The notion behind $m(\omega_0)$, however, has been lost together with the certainty of interleaving, since SERA does not guarantee that every node of G is a sink in the period the same number of times. For $i \in N$, an alternative definition is that of $m_i(\omega_0)$, which we henceforth use to denote the number of times node i is a sink in the period, not necessarily the same for all nodes.

Determining the schedule $\mathcal{S}$ through SERA proceeds according to the following algorithm.

Algorithm SERA:

1. Choose ω_0.
2. $t := 0$.
3. Obtain ω_{t+1} from ω_t, employing advancement as described.

4. If the period has not yet occurred, then $t := t + 1$; go to Step 3. If it has, then let $p(\omega_0)$ be its number of orientations (with associated buffer-occupation configurations), $m_i(\omega_0)$ the number of times node i appears in them as a sink, and $\omega_k, \omega_{k+1}, \ldots, \omega_{k+p(\omega_0)-1}$ the orientations themselves. Output

$$\mathcal{S} = \langle \mathrm{Sinks}(\omega_k), \mathrm{Sinks}(\omega_{k+1}), \ldots, \mathrm{Sinks}(\omega_{k+p(\omega_0)-1}) \rangle$$

and

$$T(\mathcal{S}) = \frac{\sum_{i \in T} m_i(\omega_0)}{p(\omega_0)},$$

where T is the set of the nodes of G that correspond to terminal edges of the paths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$.

In this algorithm, note that the determination of $T(\mathcal{S})$ generalizes what is done in SER. This is achieved by adopting $\mathrm{delivered}(\mathcal{S}) = \sum_{i \in T} m_i(\omega_0)$ while maintaining $\mathrm{length}(\mathcal{S}) = p(\omega_0)$ in Eq. (1.1). Particularizing this to the case of SER yields $\mathrm{delivered}(\mathcal{S}) = Pm(\omega_0)$, as desired, since $m_i(\omega_0)$ becomes $m(\omega_0)$ for any node i of G and moreover $|T| = P$.

Chapter 3

SER and SERA experimentation

We have conducted extensive computational experiments to evaluate algorithms SER and SERA, the latter with a few different values for the buffering parameter B. Before we present results in Section 3.3, here we pause to introduce the methodology that was followed. This includes selecting the network topology that eventually leads to graph G and the choice of the initial acyclic orientation of G.

3.1 Topology generation

We generated 1600 networks by placing n nodes inside a square of side 1500. For each network the first node was positioned at the square's center. Given the nodes' communication (or interference) radius R, and with it the neighborhood relation among nodes (i.e., two nodes are neighbors of each other if and only if the Euclidean distance between them is no greater than R), we proceeded to positioning the remaining nodes randomly, one at a time. Positioning a node was subject to the constraints that it would have at least one neighbor, that no node would have more than Δ neighbors, and moreover that no two nodes would be closer to each other than 25 units of Euclidean distance. Repeated attempts at positioning nodes while satisfying these constraints were not allowed to number more than 1000 per network. When this limit was reached the growing network was wiped clean and a new one was started. The value of R was determined so that, had the nodes been positioned uniformly at random, a randomly chosen radius-R circle would have expected density proportional to Δ/R^2 and about the same density as the whole network, i.e.,

$\Delta/R^2 \propto n$. Choosing the proportionality constant to yield $R = 200$ for $n = 80$ and $\Delta = 4$ results in the formula $R = 200\sqrt{20\Delta/n}$. Of the 1600 networks thus generated, there are 100 networks for each combination of $n \in \{60, 80, 100, 120\}$ and $\Delta \in \{4, 8, 16, 32\}$.

For each network we generated $50n$ sets of paths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$, each 100 sets corresponding to a different value of P. Each of the sets resulted in a different D, then in G, as explained in Sections 1.1 and 1.2. The $50n$ sets comprise 100 groups of $n/2$ sets each. The first of these sets for a group has $P = 1$ and the single path it contains is the shortest path from a randomly chosen node to another in the network. Each new set in the group is the previous one enlarged by the addition of a new path, obtained by selecting two distinct nodes randomly, provided they do not already participate in the previous set. This goes on until $P = n/2$, so in the last set every one of the n nodes participates as either the origin or the destination of one of the P paths. For the sake of normalization, the results we present for $T(\mathcal{S})$, given for $P = 1, 2, \ldots, n/2$, are shown against the ratio $P' = 2P/n \in (0, 1]$.

3.2 Initial acyclic orientation

Once G has been built for a fixed network and a fixed set of paths, the acyclic orientation ω_0 of G has to be determined. Our general approach is to label every node of G with a different number and then to direct each edge from the node that has the higher number to the one that has the lower. Although the resulting orientation is clearly acyclic, we are left with the problem of labeling the nodes. We approach this problem by resorting to the paths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P$ from which G resulted, since the nodes of G are in one-to-one correspondence with the directed edges on the paths. It then suffices to number the paths' edges.

We consider four numbering schemes:

ND-BF. The paths are organized in the nondecreasing order of their numbers of edges (ties are broken by increasing path number). The edges are then numbered breadth-first from the path's origins, given this organization of the paths.

ND-DF. The paths are organized in the nondecreasing order of their numbers of

edges (ties are broken by increasing path number). The edges are then numbered depth-first from the paths' origins, given this organization of the paths.

NI-BF. The paths are organized in the nonincreasing order of their numbers of edges (ties are broken by increasing path number). The edges are then numbered breadth-first from the paths' origins, given this organization of the paths.

NI-DF. The paths are organized in the nonincreasing order of their numbers of edges (ties are broken by increasing path number). The edges are then numbered depth-first from the paths' origins, given this organization of the paths.

3.3 Computational results

We divide our results into two categories. First we give statistics on the 1600 networks generated for evaluation of the algorithms. Then we report on the values obtained for $T(\mathcal{S})$ by SER and SERA.

One of the statistics is particularly useful: despite its simplicity, we have found it to correlate with the SERA results in a fairly direct way. This statistic is based on a function of G that aims to quantify how the interference among the initial P directed paths is reflected in the structure of G. We denote this function by $\rho(G)$ and let it be such that

$$\rho(G) = \frac{P|E'|}{\sum_{p=1}^{P} |Y_p|}. \tag{3.1}$$

In this equation, recall that the sets $Y_1, Y_2, \ldots, Y_P$, one for each of the initial directed paths, contain the edges that ultimately become the nodes of G. Thus, $\sum_{p=1}^{P} |Y_p|/P$ is the average number of edges on a path. Moreover, we let $E' \subseteq E$ be the set of G's edges whose end nodes correspond to edges of distinct paths. In words, then, $\rho(G)$ is the average number of off-path transmissions that interfere with the transmissions of a path having the average number of edges.

3.3.1 Properties of the networks generated

The 1600 networks we generated comprise a wide variety of topological traits, as exemplified in Figure 3.1, including those that make up star topologies, rings, and grids. Our results on the value of $T(\mathcal{S})$ for SER and SERA, though statistical in

nature, are therefore to be regarded as reflecting how the presence of such variability affects our algorithms' performance.

The networks' distributions of node degrees are given in Figure 3.2, which contains one panel for each of the four values of Δ and all four values of n. Their distributions of the numbers of edges on the P paths for $P = n/2$ are given in Figure 3.3, again with one panel for each of the four values of Δ and all four values of n.

We see from Figure 3.2 that the degree distributions peak at the degree Δ, falling approximately linearly toward the lower degrees (except for $\Delta = 32$, where a plateau is observed midway). Furthermore, the lowest observed degree grows with Δ, which is expected from the formula that gives the radius R as an increasing function of Δ. We also see from the figure that these distributions are approximately independent of the value of n for fixed Δ; in reference to that same formula, we see that letting R decrease with n does indeed have the expected effect of maintaining an approximately uniform node density throughout the containing square.

We also expect path sizes to be smaller as Δ increases, and this is in fact what Figure 3.3 shows. In fact, larger Δ values decreases the variability of path sizes, which moreover get concentrated around an ever smaller mean. For fixed Δ, what we see in the figure is a consistent shift to the right (i.e., greater path sizes) as n grows. This reflects the fact that larger n for fixed Δ leads to smaller R, thus to longer paths.

These observations are summarized in Table 3.1, where the mean degree and mean path size are given for each combination of n and Δ values. This table also shows the average value of $\rho(G)$, defined above as an indicator of how much interference there is in G among all P paths, when G refers to $P = n/2$. For fixed n, it is curious to observe that $\rho(G)$ decreases as Δ is decreased from 32 through 8, but then appears to flatten out or even rebound slightly as Δ is further decreased to 4. Each of these averages corresponds to 10^4 G instances (100 instances corresponding to the $P = n/2$ case of each of the 100 networks for fixed n and Δ) and is significant to the extent of the confidence interval reported for it in the table's rightmost column. As we demonstrate shortly, the peculiar behavior of $\rho(G)$ helps explain a lot of what is observed with respect to how $T(\mathcal{S})$ behaves in the case of SERA.

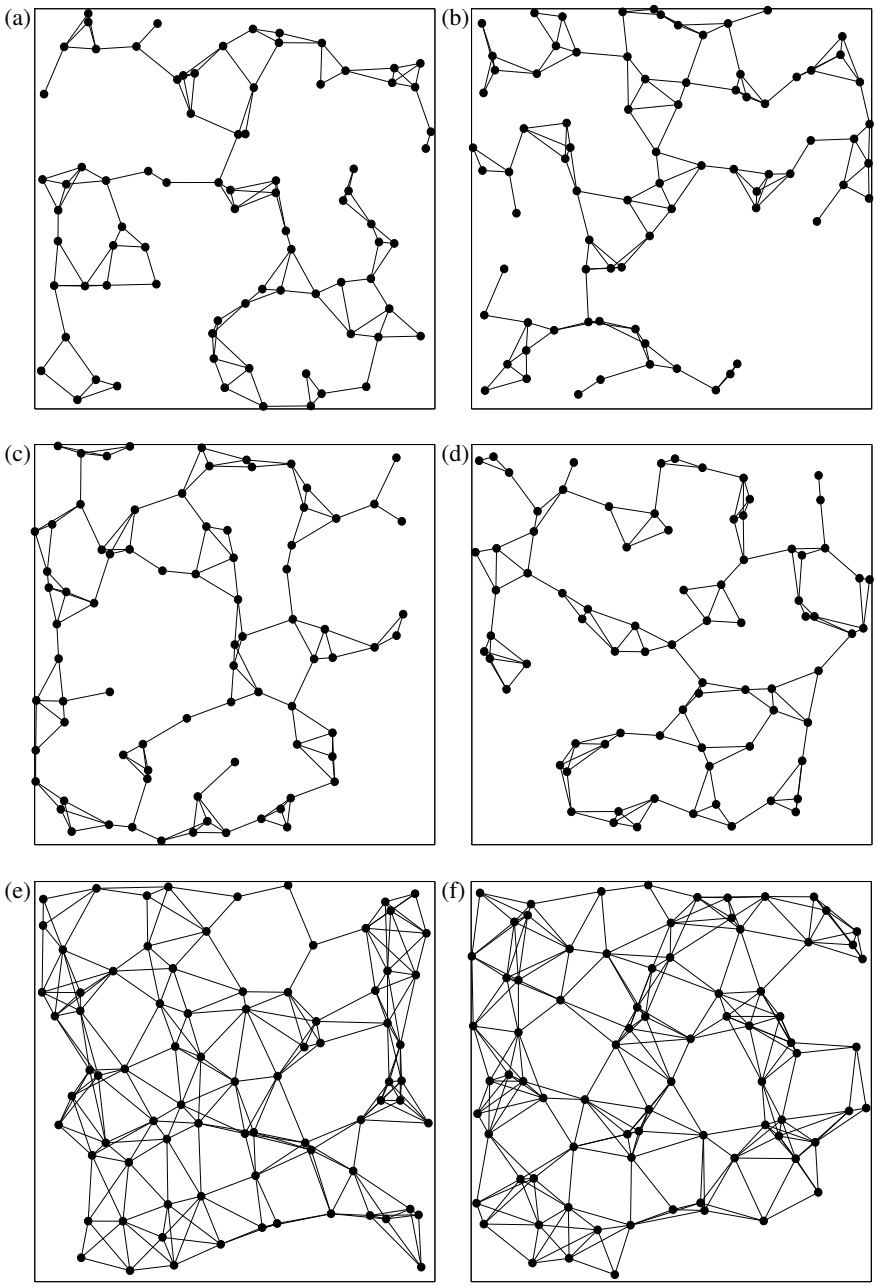

Figure 3.1: A sampler of the networks that were generated for $n = 80$. A great variety of arrangements is included, ranging from those in which nodes coalesce into groups that form a star-like topology (a, b), to those in which such groups tend to form rings (c, d), to those that are somewhat grid-like (e, f).

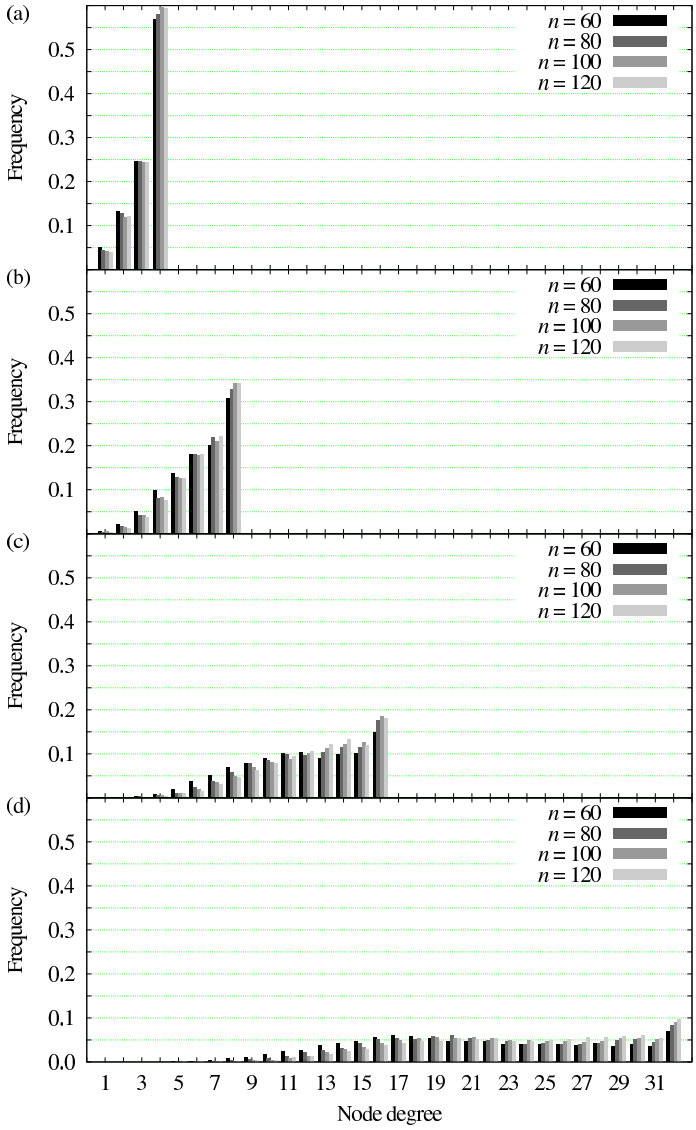

Figure 3.2: Degree distributions of the 1600 networks, for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). For each combination of n and Δ the distribution refers to 100 networks.

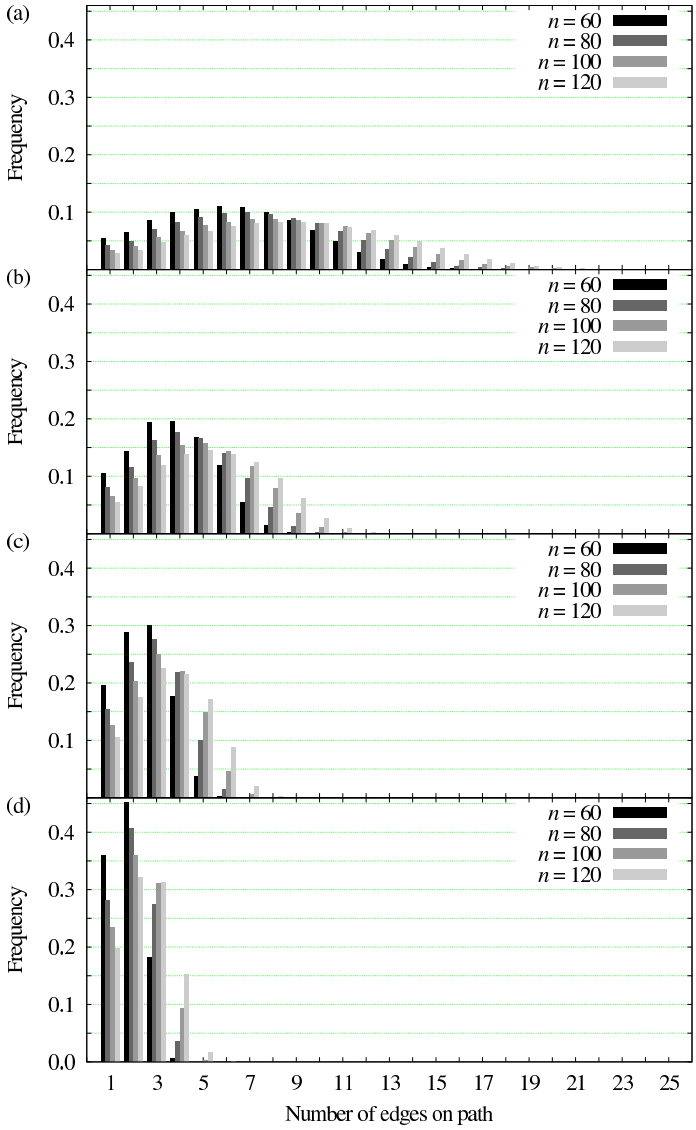

Figure 3.3: Path-size (number of edges) distributions of the 1600 networks, for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). For each combination of n and Δ the distribution refers to 100 networks and to 100 path sets for each network, each set comprising $P = n/2$ paths.

Table 3.1: Mean values of the distributions in Figures 3.2 and 3.3, and the average $\rho(G)$ values for the 10^4 G instances corresponding to each combination of n and Δ when $P = n/2$. Confidence intervals refer to these averages and are given at the 95% level.

n	Δ	Mean degree	Mean path size	$\rho(G)$	
				Average	Conf. int.
60	4	3.33	7.46	0.4	0.06
	8	6.22	4.85	0.37	0.06
	16	11.67	3.57	0.4	0.05
	32	21.23	2.84	0.58	0.05
80	4	3.36	8.32	0.37	0.06
	8	6.37	5.36	0.35	0.06
	16	12.17	3.92	0.39	0.05
	32	22.36	3.06	0.59	0.05
100	4	3.40	9.3	0.36	0.06
	8	6.40	5.86	0.34	0.07
	16	12.40	4.22	0.38	0.06
	32	23.09	3.27	0.6	0.05
120	4	3.40	9.95	0.34	0.06
	8	6.45	6.28	0.33	0.07
	16	12.50	4.52	0.38	0.06
	32	23.59	3.47	0.58	0.05

3.3.2 Results

Our results for SER are given in Figure 3.4 as plots of $T(\mathcal{S})$ against the P' ratio introduced earlier in this chapter. Each of the figure's four panels is specific to a fixed Δ value and shows a plot for each value of n combined with either the ND-BF or the ND-DF numbering scheme. All results relating to the NI-BF and NI-DF schemes are omitted, as we found them to be statistically indistinguishable from their ND counterparts. From this figure it seems clear that, as P' increases (i.e., as the number of paths P grows towards $n/2$), the superiority of the BF schemes over the DF schemes becomes apparent, more pronouncedly so for the lower values of Δ. The reason why the BF schemes tend to perform better than the DF schemes should be intuitively clear: the BF schemes number the transmissions that are closer to the paths' origins first, therefore with the lowest numbers. As the initial acyclic orientation of G is built from these numbers, the first sinks during the operation of SER will correspond to starting parallel traffic on as many paths as possible. Overall it also seems that larger values of n lead to better performance for fixed Δ, but the distinction appears to be only marginal and is sometimes obscured by the

confidence intervals.

A similar set of plots is given in Figure 3.5, now displaying our results for SERA as plots of $T(\mathcal{S})$ against the ratio P'. Once again there is one panel for each value of Δ, and once again several possibilities regarding the numbering schemes are omitted because of statistical indistinguishability. This is also true of the various possibilities for the value of B, with the single exception we mention shortly. Thus, most plots correspond to the ND-BF numbering scheme with $B = 1$. The single exception is that of $n = 60$ with $\Delta = 4$, for which we also report on the $B = 2$ case. For fixed Δ and P', increasing n also leads to increased $T(\mathcal{S})$. In the particular case of $n = 60$ and $\Delta = 4$, increasing B from 1 to 2 also causes $T(\mathcal{S})$ to increase.

As we fix Δ, n, and P', Figures 3.4 and 3.5 reveal that $T(\mathcal{S})$ is consistently higher for SERA than it is for SER (by a factor of about 2 to 4) across all values for these quantities, thereby establishing the superiority of the former algorithm over the latter. For sufficiently large n this occurs for the same value of B (that is, for $B = 1$), which furthermore establishes that this superiority does not in general depend on the availability of more buffering space. It is, instead, determined solely by the elimination in SERA of the mandatory alternance of interfering transmissions in SER.

Fixing n and P' while varying Δ (i.e., moving between panels) yields further interesting insight about the two algorithms. While for SER increasing Δ under these conditions causes $T(\mathcal{S})$ to increase monotonically (though sometimes almost imperceptibly) for the same numbering scheme, doing the same for SERA for constant B leads $T(\mathcal{S})$ to behave in a markedly non-monotonic way. In fact, as Δ is increased from 4 to 8 there is also an increase in $T(\mathcal{S})$, but increasing Δ further through $\Delta = 32$ leads to decreases in $T(\mathcal{S})$. As we anticipated earlier, this is fully analogous to the behavior of $\rho(G)$ as Δ is increased in the same way while all else remains constant. This suggests that what determines the relative behavior of $T(\mathcal{S})$ in these circumstances is the intensity of inter-path interference as it gets shaped by the structure of G. In other words, $T(\mathcal{S})$ and $\rho(G)$ tend to vary along somewhat inverse trends with respect to each other.

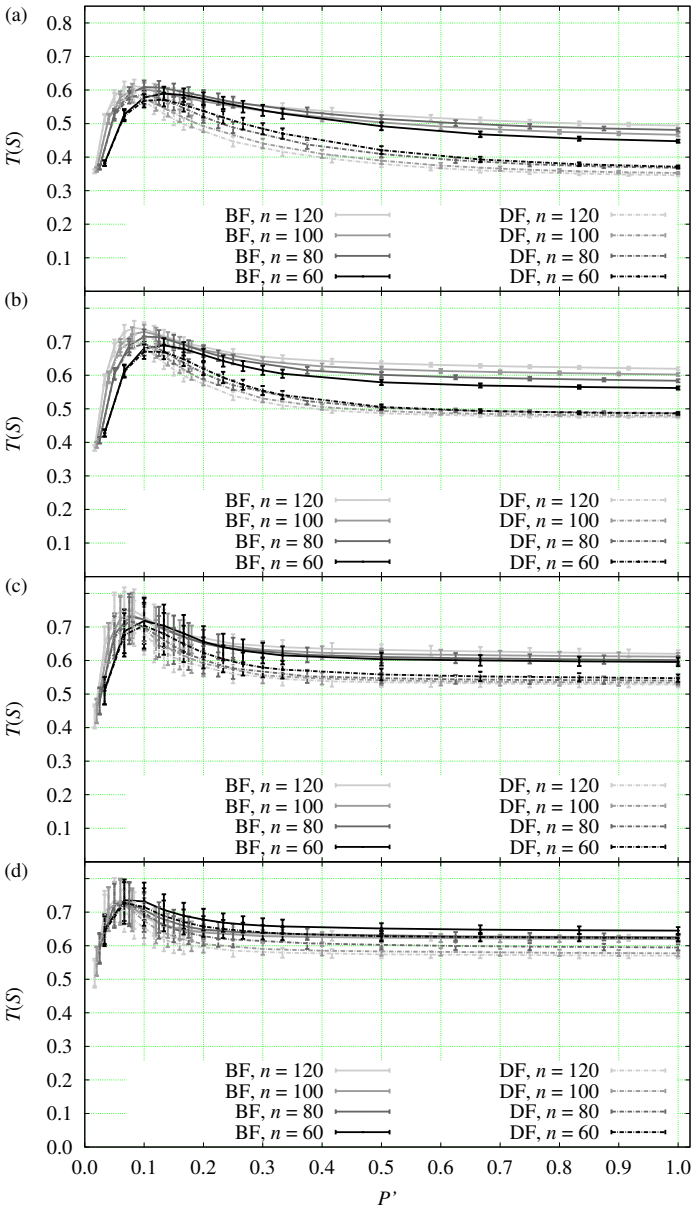

Figure 3.4: Behavior of $T(\mathcal{S})$ for SER under the two numbering schemes ND-BF and ND-DF, with $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 G instances that correspond to each combination of n and Δ for each value of P. Error bars are based on confidence intervals at the 95% level.

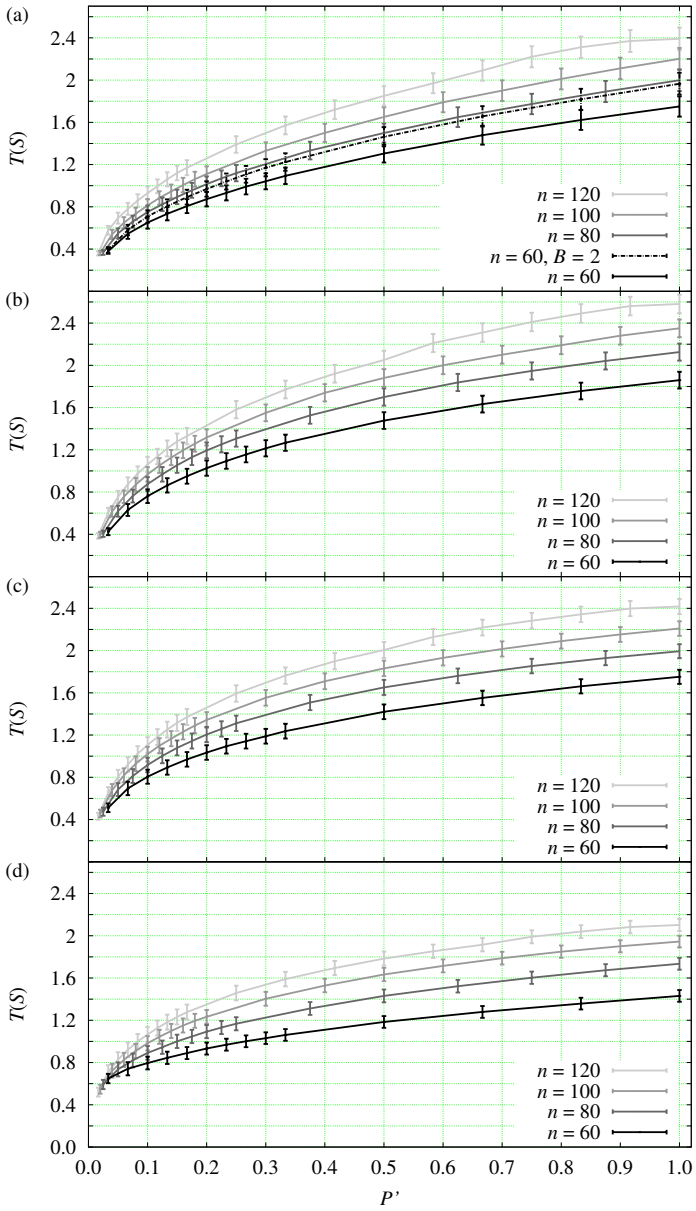

Figure 3.5: Behavior of $T(\mathcal{S})$ for SERA under the numbering scheme ND-BF, with $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 G instances that correspond to each combination of n and Δ for each value of P. Error bars are based on confidence intervals at the 95% level.

3.4 Discussion

When SER is used, it follows from our discussion in Chapter 2 that $T(\mathcal{S}) = T^*_{\text{int}}(\mathcal{S})$. By Eq. (1.5), we then have

$$T(\mathcal{S}) \leq \frac{P}{\chi^*_{\text{int}}(G)}, \tag{3.2}$$

where achieving equality requires that we choose ω_0 optimally. Now let $\varphi(G) = \max\{\omega(G), |N|/\alpha(G)\}$, where $\omega(G)$ is the number of nodes in the largest clique of G and $\alpha(G)$ is the number of nodes in the largest independent set of G. It can be shown that $\chi^*_{\text{int}}(G) \geq \varphi(G)$,[1] whence

$$T(\mathcal{S}) \leq \frac{P}{\varphi(G)} = \frac{P'n}{2\varphi(G)}, \tag{3.3}$$

where we have taken into account the way we handle P in all our experiments. We see then that $T(\mathcal{S})$ is bounded from above by the fraction of $n/2$ given by $P'/\varphi(G)$. For fixed P', this fraction tends to be small if the largest clique of G is large or its largest independent set is small, whichever is more influential on $\varphi(G)$. Either possibility bespeaks the presence of considerable interference among the transmissions represented by the $|N|$ nodes of G.

Of course, in general we have no practical way of knowing how close each ω_0 we choose is to being the optimal starting point for SER, nor of knowing how different $\chi^*_{\text{int}}(G)$ and $\varphi(G)$ are for the G instances we use. So the bound given in Eq. (3.3), located somewhere between $30P'/\varphi(G)$ and $60P'/\varphi(G)$ for our values of n, cannot be used as a guide to assessing how low the $T(\mathcal{S})$ values shown in Figure 3.4 really are. But the bound's sensitivity to growing interference in G does provide some guidance, since all plots in the figure become flat from about $P' = 0.3$, regardless of the value of n or the numbering scheme used. Perhaps every G corresponding to such values of P' share some structural property, like a very large clique or only very small independent sets, that renders the resulting values of $T(\mathcal{S})$ oblivious to all else.

As for SERA, since the schedules it produces depart from a strict characterization as multicolorings of G, no upper bounds on $T(\mathcal{S})$ are known to us. Nevertheless, a

1. See [Lin03], where the interleaved multichromatic number of G is referred to as G's circular chromatic number, and references therein.

comparison with SER as provided by Figures 3.4 and 3.5 reveals that $T(\mathcal{S})$ for SERA surpasses $T(\mathcal{S})$ for SER by a substantial margin, and also that SERA is capable of finding ways to improve $T(\mathcal{S})$ somewhat even as P' grows. If our observation above regarding the structure of G as an inherent barrier to improving $T(\mathcal{S})$ as P' grows is true, then the barrier's effects under SERA are considerably attenuated. This, we believe, is to be attributed to SERA's aggressively opportunistic approach of abandoning the interleaving that is the hallmark of SER.

Another course of action regarding the discovery of how close SER and SERA get to the optimal value of $T(\mathcal{S})$ in each case (that is, how close their initial acyclic orientations come to being optimal) is to concentrate on suitably small problem instances. If such instances prove amenable to some method to discover what the optimal orientations are, then we may be able to draw some meaningful conclusion regarding the performance of SER and SERA despite the modest instance sizes. Unfortunately, we have no exact method to optimize our algorithms' initial conditions, except for resorting to an explicit enumeration of all possibilities and recording the best value of $T(\mathcal{S})$ that is found. Although the number of a graph's distinct acyclic orientations grows rather quickly with graph size [Bar00a], being in the worst case given by the factorial of the number of nodes, we have found that such enumeration is feasible for small values of n and Δ. We have therefore proceeded with it for $n = 10$ and $\Delta = 4$, following the exact same methodology introduced earlier in this chapiter, and give results in Figure 3.6. While these results confirm that SERA outperforms SER also at this reduced scale, clearly both algorithms come very close to performing optimally, i.e., to achieving the value of $T(\mathcal{S})$ that results from the optimal initial acyclic orientation in each case.

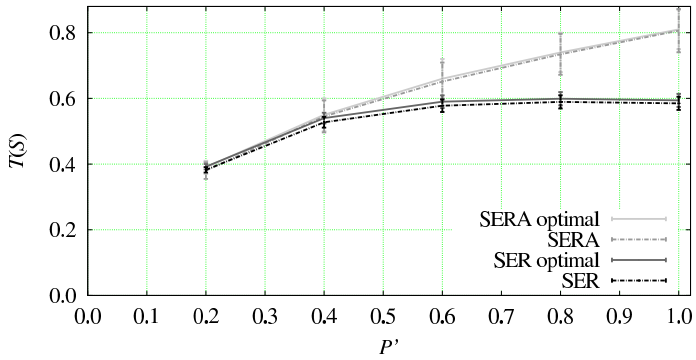

Figure 3.6: Behavior of $T(\mathcal{S})$ for SER and SERA with $n = 10$ and $\Delta = 4$. For each algorithm, results are given based on the numbering scheme ND-BF and also for the optimal numbering scheme. SERA results are for $B = 2$. Data are averages over 10^4 G instances for each value of P. Error bars are based on confidence intervals at the 95% level.

Chapter 4

Interference among wireless routes

A roadmap on reducing mutual radio interference is to join routing algorithms with an interference avoidance approach such as: power control, link scheduling or multi-channel radios [AWW05]. This type of network interference problem has been faced by a considerable amount of different strategies over the literature [Abo04, SJA+10, CQYM00, CEM+08]. Some studies show that the use of multi-path routing to distribute traffic helps to improve route recovery and load balance more than single-path strategies and it may lead to better throughput values over the entire network [KLF06, ACSR10]. Nevertheless, these benefits are directly related to how multiple paths interferes with each other, an aspect not addressed by multi-path strategies. Therefore, the lack of performance of most employed and promising routing algorithms is a direct consequence of the interference among routes, because they do not tackle directly this interference during route discover process. The single exception is the algorithm reported in [WB06], however it uses geographic information (like localization aided by GPS) to find paths with sufficient spacial separation, so they do not interfere with each other. Unfortunately, this information is available only in some types of sensor networks [DEA06] and the algorithm reported is normally related to larger instances of a NP-hard problem [WB08].

Here we propose a different approach to avoid interference among multi-path routes which, to the best of our knowledge, is a novel formulation and solution. We start by extracting the network path set from a routing algorithm. Next, we build a new path set that is a sub-set of the paths discovered by the algorithm. For that, we select paths that avoid mutual interference and by consequence they may

transmit more packets. Our objective is to improve the throughput performance of the path set, that is, to increase the number of packets delivered to the last node of each path (the throughput on the route destinations). This approach lies at the principal problem of successfully design routing protocols for wireless networks, because it deals with the interference when the network is heavily loaded. The solution we propose is to build a multi-path set for each possible pair of nodes in a WMN that avoids mutual interference. Our approach is easily adaptable to the majority of existing multi-path and single-path routing algorithm through some minor modifications, not only from the fact that it uses only local information from nodes' neighbors, but also due to its simplicity, since it works without the knowledge of the entire path of the available routes.

4.1 Problem formulation

We consider a pair of wireless network nodes i,j ($i \neq j$ not neighbor of j) and a collection $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P\}$ of P simple wireless network paths, each having at least i as source and j as destination, and let $\mathcal{P}_{ij} = \bigcup_{p=1}^{P} \mathcal{P}_p$. These path sets of nodes are represented by $N_1, N_2, \ldots, N_P$, respectively, not necessarily disjoint from one another, and we let $N_{ij} = \bigcup_{p=1}^{P} N_p$. Their sets of edges (wireless network links) are $E_1, E_2, \ldots, E_P$ and we define that, for $p \neq q$, a member of E_p and one of E_q are the same only if they join the same two nodes. Letting $E_{ij} = \bigcup_{p=1}^{P} E_p$, $\mathcal{P}_{ij}$ is an edge disjoint path set if $|E_{ij}| = \sum_{p=1}^{P} |E_p|$, that is, each member of E_{ij} belongs only to one of the $E_1, E_2, \ldots, E_P$ sets of edges. Also, $\mathcal{P}_{ij}$ is a node disjoint path set if $|N_{ij}| = \sum_{p=1}^{P} |N_p|$, that is, each member of N_{ij} belongs only to one of the $N_1, N_2, \ldots, N_P$ sets of nodes.

Before we formulate the problem of selecting non-interfering paths, we have to introduce the concept of the *interference disjoint path set*. We call an interference disjoint path set any set $\mathcal{P}_{ij}^{itf} = \bigcup_{p=1}^{P} \mathcal{P}_p$ such that a member $\mathcal{P}_p$ does not interfere with one $\mathcal{P}_q$ for $p \neq q$. Thus, for two link activations (network transmissions) t_p and t_q belonging to different paths, each packet transmitted by t_p and t_q is received successfully, even if t_p and t_q are simultaneous. The two exceptions are i's and j's links (origin and destination of all paths), since we assume that each node has only

one radio/orthogonal frequency of transmission. Note that, an interference disjoint path set is also a node disjoint set, and a node disjoint set is also an edge disjoint set. Each one has an improvement over the subsequent with respect to the mutual interference avoidance (a lower level of route coupling) [PHST00]. Nevertheless, $\mathcal{P}_{ij}^{itf}$ is the only set that takes into account the interference directly and consequently it is the only interference free disjoint set. We present in Figure 4.1 an example of how a node and an edge disjoint sets suffer from mutual interference.

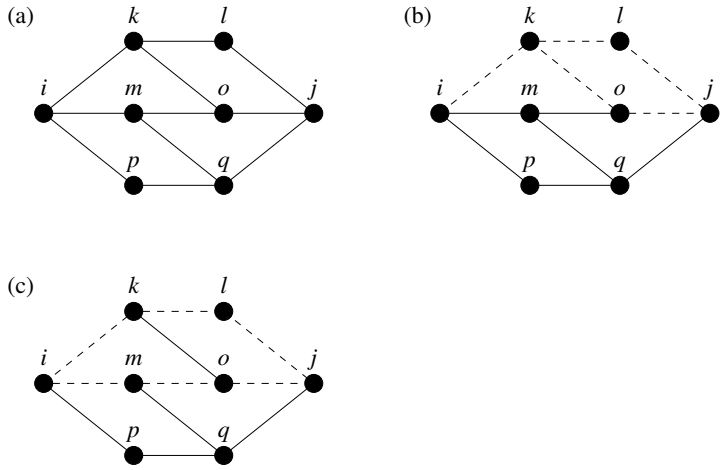

Figure 4.1: A graph representation of a wireless network (a), an edge disjoint set in (b) and a node disjoint path set in (c), both represented by dashed lines. Neither of the paths in (b) may be used simultaneously, because node k cannot transmit two packets at the same time. The same occurs in (c) for k and m, due to the fact that o is in the communication radius of k and m, then it cannot receive two parallel transmissions.

Now we can define our combinatorial optimization problem as to transform $\mathcal{P}_{ij}$ into $\mathcal{P}_{ij}^{itf}$ over a WMN by finding the subset in $\mathcal{P}_{ij}$ that maximizes the throughput between i and j (the number of packets arriving in j sent by i). We recall that, the word throughput refers to the sum of packets arriving at all destination of a path set per one unit of time. We do this for conciseness and no confusion should arise, since the other forms of throughput calculus no longer matter for our problem.

The importance of finding non-interfering paths is that the interference among the multiple paths can drastically decrease the capacity of the path set and eliminate the advantages of multi-path routing [TM06, TTAE09]. However, for the best of our knowledge, the most commonly multi-path route discovery approach produces at most a node disjoint path sets and moreover none of them produces a $\mathcal{P}_{ij}^{itf}$ set [LG01, WWL$^+$06, WGLA09, LR07, WDJ$^+$08, ABL06, LSG01, TH01, SR06, CS03].

Also, $\mathcal{P}_{ij}^{itf}$ needs information of the interference levels at each link location for every link activation, so it is more complex to be obtained than the others disjoint sets. On the other hand, only an exchange of tokens between nodes are necessary to produce a node or an edge disjoint set [LC04]. Moreover, a node disjoint path set is a particular case of $\mathcal{P}_{ij}^{itf}$, hence the problem of finding an interference disjoint path set between i and j is also NP-hard [WB08].

4.2 Selecting paths by the neighbors' independent set

Obtaining $\mathcal{P}_{ij}^{itf}$, however, requires to adopt an interference model and its assumptions [SHLK09], since it is only a conceptual definition of a path set as described before. Under the assumptions of the protocol interference model, each link may transmit in both directions for error control and two transmissions t_p and t_q do not mutually interfere if t_p's origin and destination are neither neighbors of t_q's origin or destination [BBK+04]. We also assume that a node's communication and interference radii are equal and that their value R is the same for all nodes. We say that two nodes are neighbors of each other in the WMN if and only if the Euclidean distance between them is no greater than R. Therefore, we define a graph $G_{ij} = (N_{ij}, E_{ij}^*)$ to be representative of a wireless network under the protocol interference model, where all P paths in $\mathcal{P}_{ij}$ are presented. E_{ij}^* is the set E_{ij} enlarged by including all edges that do not correspond to any of the P paths (members of $\mathcal{P}_{ij}$), but nevertheless reflect that any two members of N_{ij} are within the interference radius of other. In G_{ij}, every possible interference that may arise between i to j paths is directly related to the neighborhood of each member of N_{ij}. But before we present an algorithm to build $\mathcal{P}_{ij}^{itf}$ in G_{ij}, we need to introduce a new concept of independent sets.

An *independent path set* between i and j is a set denominated here by $\mathcal{P}_{ij}^{ind}$, where $\mathcal{P}_{ij}^{ind} = \mathcal{P}_{ij}$ if $\mathcal{P}_{ij}$ is a node disjoint path set and $E_{ij} = E_{ij}^*$. It follows from this definition that, any two pair of paths in $\mathcal{P}_{ij}^{ind}$ share no nodes or edges, with the exception of i, j and their edges. An example of $\mathcal{P}_{ij}^{ind}$ is shown in Figure 4.2 as an alternative for the path sets represented in Figure 4.1. This set provides the greatest throughput performance because i may use all these paths simultaneously to transmit packets to j, without the inconvenience of the mutual interference commonly

present in wireless transmissions. Also, $\mathcal{P}_{ij}^{ind}$ is the preferential set for *fail-over* purposes, due to the fact that none of its paths share resources with the others, with the exception of i and j. Note that, if a neighbor k of i belongs to N_{ij}, then i has at least one path through k to j, thus G_{ij} not necessarily have all neighbors of i in the wireless network.

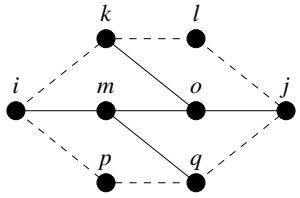

Figure 4.2: A graph model of a wireless network, in which an independent path set is represented by dashed lines.

We use the protocol model assumptions to define the path neighborhood of $\mathcal{P}_{ij}^{itf}$ obtaining $\mathcal{P}_{ij}^{ind}$. Adopting a different model would simply require us to define another neighborhood according to the set of available paths between i and j.

We wish to address the problem of build $\mathcal{P}_{ij}^{ind}$ in terms of the neighborhood of i and use a set of pre-discovered paths between i and j. Moreover, we desire an algorithm that uses only local information to the origin and be easily adapted in any multi-path routing algorithm. Clearly, the graph G_{ij} is not a good candidate for this, since it assumes that i has a knowledge over every node of every path in $\mathcal{P}_{ij}$. In our first stage to obtain a graph more suitable, G_{ij} will be reduced to the sub-graph $G'_{ij} = (N'_{ij}, E'_{ij})$ defined as follows:

1. Initially, $N_{ij} = N'_{ij}$ and $E^*_{ij} = E'_{ij}$.

2. Eliminate j and all edges from E'_{ij} that touch it (the interference caused by j is not tacked by $\mathcal{P}_{ij}^{ind}$).

3. Eliminate all nodes from N'_{ij} if the distance to i is greater than two (the smallest path to i has three or more hops) and all edges from E'_{ij} that touch them.

4. Eliminate all edges from E'_{ij} connecting any two nodes in N'_{ij}, where each one has distance to i greater than one.

Step 2 and 3 above restrict i from access any knowledge over a distance greater than

two. This process, from the set $\mathcal{P}_{ij}$ of paths through graph G'_{ij}, is illustrated in Figure 4.3.

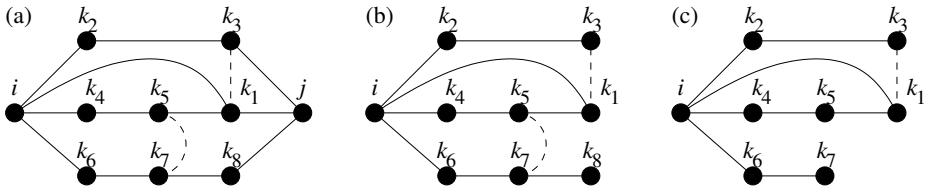

Figure 4.3: The reduction of G_{ij} from the given path set $\mathcal{P}_{ij} = \{\{i, k_1, j\}, \{i, k_2, k_3, j\}, \{i, k_4, k_5, k_1, j\}, \{i, k_6, k_7, j\}\}$. In panel (a), the graph G_{ij} based on $\mathcal{P}_{ij}$, where the dashed edge k_1k_3 and k_5k_7 represents the interference (through Step 1, $G'_{ij} = G_{ij}$). Step 2 remove j and all edges that touch it (b). Panel (c) presents Steps 3 and 4 (the last one removes only the edge k_5k_7).

Our next and last stage is then to create a graph $D_{ij} = (X_{ij}, Y_{ij})$ based on G'_{ij} and $\mathcal{P}_{ij}$ according to the following procedure:

1. Include in X_{ij} one node for each path in $\mathcal{P}_{ij}$.

2. The edge set Y_{ij} is obtained by the following steps:

   i. Connect any two nodes in X_{ij} if the paths they represent share, at least, one node $\in N'_{ij}$.

   ii. Connect any two nodes in X_{ij} if there is one or more edges $\in E'_{ij}$ that connect the paths they represent.

3. Associate the cost C_p (assigned to the path p) with each node in X_{ij} that have come from p, where C_p can be the number of hops, latency, or any other type of network path cost.

The graph D_{ij} provides a simple way that leads to $\mathcal{P}_{ij}^{ind}$ by selecting a subset of i's neighbors that at least do not interfere with each other (each path in $\mathcal{P}_{ij}$ is represented by a node in D_{ij}). Thus, we select the best candidate for the subset of paths in $\mathcal{P}_{ij}$, henceforth called $\mathcal{P}_{ij}^{R}$ (refined $\mathcal{P}_{ij}$), that heuristically tries to maximize the throughput between i and j, by computing the maximum weighted independent set of the graph D_{ij}. The weight of an independent set is

$$W = \sum_{p \in IS} \frac{1}{C_p}, \tag{4.1}$$

where IS is the set of all nodes of the independent set. The maximum weighted independent set is the intuitive solution, since it is the interference free set with the maximum number of neighbors (each neighbor is an index of the future selected paths) and with the minimum sum of costs. Figure 4.4 presents the final stage, where G'_{ij} is transformed into D_{ij}.

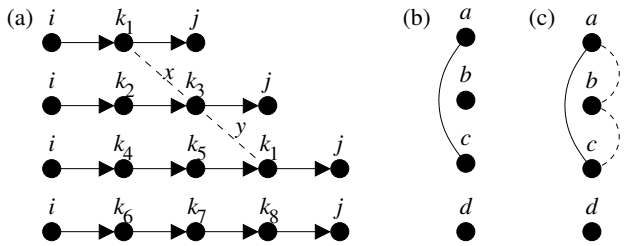

Figure 4.4: The construction of the graph D_{ij}. We start in panel (a) with the same set $\mathcal{P}_{ij}$ of Figure 4.3, where dashed edges x and y are representations of the interference between nodes k_1 and k_3 in G'_{ij} of Figure 4.3-(c) (edge k_5k_7 is not present in G'_{ij}, so it was not included). Next, we create one node for each path through Step 1 in panel (b) (nodes a, b, c and d represent paths $\{i, k_1, j\}$, $\{i, k_2, k_3, j\}$, $\{i, k_4, k_5, k_1, j\}$ and $\{i, k_6, k_7, j\}$, respectively). In the same panel we join nodes by an edge if the paths they represent share one or more nodes through Step 2.i. Finally, we join nodes by an edge if the paths they represent are connected by one or more edges through Step 2.ii in panel (c) (dashed edges). If we associate the same cost to each node in D_{ij} (Step 3), then the maximum weighted independent set will be the one with the greatest number of nodes. In this case its easy to see that $\{a, d\}$, $\{b, d\}$ and $\{c, d\}$ are suitable candidates. If we select $\{a, d\}$ as the independent set, then $\mathcal{P}^R_{ij} = \{\{i, k_1, j\}, \{i, k_6, k_7, k_8, j\}\}$.

This entire process of graph transformation and selection of the $\mathcal{P}_{ij}$ subset by the maximum weighted independent set is our heuristic approach to approximate $\mathcal{P}^{ind}_{ij}$, hereafter called multi-path refinement algorithm (MRA). However, MRA has an assumption over the original problem or an approximation aiming at building $\mathcal{P}^{ind}_{ij}$, where it is only necessary to find an interference free set of the first hop. This is the subgraph G'_{ij} formed by all neighbors of i or equivalently the independent set of D_{ij}. It is clear that one of the independent set candidates corresponds to the optimal $\mathcal{P}^{ind}_{ij}$ in terms of throughput. The key to determine the appropriate candidate fall into the costs of each path in $\mathcal{P}_{ij}$, since the selected candidate will be that one with the maximum weight.

Until now we have used only a pair of nodes to describe our algorithm. To work with an entire wireless network, it is sufficient to apply MRA on all possible or desirable pairs. Moreover, brought from its simplicity, only a few modifications

have to be made in the majority of multi-path routing algorithms proposed in the literature. To do so, first i should have one $\mathcal{P}_{ij}$ for each possible j destination as part of its a routing table (or any information that distinguishes multiple paths such as the two first hops nodes). Next, to obtain the distance 2 knowledge (the second hop), i requests to each neighbor k to sends its routing table, then only neighborhood information is used. If k belongs only to one path and it is the destination, i may discard k's table or do not request it. Thus, node i builds G'_{ij} according to the routing tables where j is the destination of the entries. Finally, i transforms G_{ij} into D_{ij} (according to the process describe earlier) and selects paths based on the maximum weighted independent set of D_{ij}. Indeed, the routing algorithms must provide costs for paths and provide multiple paths for at least some origin-destination pairs of nodes in the network. In the special case of single-path routing algorithms, we present next simple modification to transform them into a multi-path version, but before proceeding we tackle another special case. The one is when $\mathcal{P}_{ij}$ contains the single-link path that connects i directly to j. In this case, we let $\mathcal{P}^R_{ij}$ be the singleton that contains only that path. It is easy to see that we cannot improve the throughput over the single direct link to the destination by including other paths, no matter how many interference free paths $\mathcal{P}_{ij}$ has.

4.3 Selecting paths in the absence of multiple routes

In the particular case of the single-path routing algorithms every node i has only one path to each destination of the network, then $\mathcal{P}_{ij} = P_1$. To overcome this restriction, node i has to request the path set of all its neighbors (the same procedure of the multi-path routing case). However, i builds G'_{ij} considering only paths given by its neighbors whose costs are equal or inferior to the cost of the single path i has to j (for this reason i acquires only loop-free paths). Note that, the original single path i has to j will be mandatorily passed by one of its neighbors, so i does not need to consider it for the construction of G'_{ij}.

The following, then, is how we formulate the single-path routing case on G_{ij}. Given an origin-destination pairs of nodes i,j the new path set $\mathcal{P}_{ij}$, will be constituted by the single paths from all i neighbors suitably prefixed by the new ori-

gin i, where their costs are less than or equal to C_{ij} (cost from i to j), that is, $\mathcal{P}_{ij} = \bigcup_{k \in \{i \text{ neighborhood}\}} \mathcal{P}_x$, where

$$\mathcal{P}_x = \begin{cases} \mathcal{P}_{kj} \text{ prefixed by } i, & \text{if } C_{kj} \leq C_{ij}; \\ \{\} & \text{otherwise.} \end{cases} \tag{4.2}$$

Now, $\mathcal{P}_{ij}$ is a multi-path set and we shall proceeds likewise the multi-path case as explained previously.

Chapter 5

MRA experimentation

We evaluated the throughput of the path sets through extensive experimentation of the following routing algorithms: AODV [PR99], AOMDV [MD02], OLSR [JMC+01] and MP-OLSR [YADP11]. For conciseness purposes, we denominated R-AODV, R-AOMDV, R-OLSR and R-MP-OLSR as the new pool of *refined algorithms* obtained after the application of MRA over the path sets of the original routing algorithms. Note that, we applied the procedure of Section 4.3 to transform AODV and OLSR into multi-path versions as previous discussed. Our experiments were executed in the network simulator NS2.34 (NS2) [ns289] and in a simulator that employs SERA link scheduling algorithm [VRBF12], introduced in Section 2.1. We used two different configuration sets of the routing algorithm parameters to compare how them affect the throughput, and likewise two different configurations of NS2 parameters (one for the path discovery process and another for the performance evaluation). These configurations were selected by a wide variation of parameters in the tuning experiments, but before we present them, we shall note that we used the same network topologies used in the experiments for SER and SERA (see Section 3.1). These network topologies eventually leaded to the graph D_{ij}.

5.1 Path discovery

For each one of the 1600 networks we generated 100 sets of n random origin-destination pair of nodes, where each node appears only once as an origin in a set. Each set of node pairs originates an instance of the path set $\mathcal{P}$ discovered by an

original routing algorithm. These path sets were posteriorly used in the experiments of the originals. In terms of our notation, we can define this path set as

$$\mathcal{P} = \bigcup_{ij \in OD} \mathcal{P}_{ij}, \tag{5.1}$$

where OD is the set of n random origin-destination pairs of nodes.

In the experiments of the refined algorithms, every $\mathcal{P}$ was refined to the corresponding set $\mathcal{P}^R$ by the application of MRA on each origin-destination pair of nodes $\in OD$ (as explained in the beginning of Section 4.2 for multi-routing algorithms and in Section 4.3 for single-routing ones), therefore

$$\mathcal{P}^R = \bigcup_{ij \in OD} \mathcal{P}^R_{ij}. \tag{5.2}$$

In a practical way, we construct $\mathcal{P}$ by letting the routing algorithm to build the routing table of each node and by uniting these tables to produce a single path set, thus we cover all possible combinations of origin-destination pair of nodes. For that, we load a network topology into NS2, and we start a simulation of 15 seconds with one flow agent for one of the possible origin-destination combinations (a CBR with packets of 1000 bytes generated every one second). Next, we start a wiped clean new simulation and do the same for another different origin-destination combination until we cover all possibilities. This procedure minimizes the packet loss due to path overload problems and it avoids that a previously discovered path interferes with the discovery of a new one. For example, if a node k is the first to reply a path request from i to an already known destination j, i will use k's path even if the optimum path passes through another node. This argument is only valid for on-demand routing algorithms (AODV and its variations depend on the network load), due to the fact of the paths found by OLSR algorithms are identical for the same network topology. Another advantage is that we obtain always the same path set, which makes the experiments reliable and reproducible.

The NS2 was used with the default configuration, however we employed the DRAND protocol [RWMX06] to avoid collision in the path discovery process (*macType* parameter). To adjust the communication radius R, we also configured the

parameter *RXThresh_* (RXT) with the appropriate value given by the program *threshold.cc* (see NS2 manual). We used the implementation of AODV, OLSR and AOMDV routing agents available in the version 2.34 of the NS2, and the MP-OLSR routing agent available at [mpo08]. For AOMDV, we made a small modification proposed by [YCJ05] to discover only node disjoint paths with at most K paths for each origin-destination pair of nodes. We adopted these modifications, since node disjoint paths are clearly more interference free. We chose $K = 5$ because it achieved the best throughput values for $2 \leq K \leq 7$. The same modifications were made on MP-OLSR (proposed by [ZLX05]) with the same K's range for the same reason, but $K = 3$ and $K = 5$ where adopted for $\Delta \in \{4, 8\}$ and $\Delta \in \{16, 32\}$ instead.

5.2 Performance evaluation

In general, we wish to evaluate the throughput of $\mathcal{P}$s against the corresponding $\mathcal{P}^R$s according to the size of OD under a heavy traffic network. To see how the number of paths affects the performance of MRA, we varied the size of OD from 1 to n (vary the size of OD is a simple matter of randomly remove an element from it until $|OD| = 1$). We use node weights of D_{ij} such that, C_p is the path's hop count. Clearly, though, any other desired metric can be used as well. Since $\mathcal{P}$ and a random set of node pairs in OD have been build for a network, we loaded the corresponded topology into NS2 and adjusted RXT accordingly. We configured the *macType* parameter to 802.11 and the routing agent to NOAH [noa04] (*adhocRouting* parameter). NOAH works only with fixed routes and do not send routing related packets, so we evaluated only $\mathcal{P}$ without the interference of control packets. Next, we started a CBR from each origin-destination pair of nodes in $\mathcal{P}$ and measured the amount of successfully arrived packets during 120 of 135 seconds of simulation (the first 15 seconds were used to assure that NS2 had warmed-up properly all CBR agents).

Before the execution of the performance experiments, we executed a large set of tuning experiments to choose the best configuration of NS2 parameters. We varied widely three NS2 parameters: the carrier sense threshold (CST) to increase the spatial re-use and consequently to increase the throughput [KLH06] (802.11

*CSThresh_* parameter), the CBR parameter and the network transmissions rates to obtain the maximum throughput and the maximum fairness possible (802.11 *dataRate_* and *basicRate_* parameters). Table 5.1 presents the first stage of the tuning experiments, where the CBR interval was varied to find the highest rate without gain of throughput (CBR1) and the lowest rate without gain of fairness (CBR2) [1]. Table 5.2 presents the values used in the performance experiments chose from the best results of the tuning experiments. For the sake of normalization, the results we present for the refined and original algorithms in the performance experiments, given for $|OD| = 1, 2, \ldots, n$, are shown against the ratio $\theta = |OD|/n \in (0, 1]$.

Table 5.1: Different parameters used in our tuning experiments.

NS2 parameter	variation	increment
CBR interval	0.001s .. 0.005s	0.0001
CST	$0.1RXT$ .. $2RXT$	$0.1RXT$
data/basic rate	1/1,2/1,11/2 Mbps	

Table 5.2: Parameters adopted in the performance experiments.

NS2 parameter	Δ			
	4	8	16	32
CBR1 interval	0.0025s	0.0027s	0.0029s	0.0031s
CBR2 interval	0.0045s			
CBR packet size	1000b			
CST	$0.6RXT$	$0.7RXT$	$0.8RXT$	$0.9RXT$
data/basic rate	11/2 Mbps			

The experiments with the NS2 evaluated the refined and original algorithms under the 802.11, a CSMA protocol. As a second evaluation method, we chose SERA that gives a link schedule under some TDMA protocol to provide a background of synchronized time slots [GL00]. We configured SERA with parameters ND-BF numbering scheme and $B = 2$. Even though SERA is a link scheduling algorithm, it seeks to schedule a set of routes' links to maximize the number of packets delivered to their destinations. If we calculate the throughput of the resulting schedule, then we can naturally extract the number of packets arriving in the path destinations during a schedule cycle (exactly our evaluation objective). Now, to obtain the throughput

1. We observed that when we varied the CBR *rate_* NS2 parameter, it tends to influence somewhat inversely the fairness and directly the throughput in all simulations with NS2.

we have to compute the time slot size of the experiments under NS2 simulations. For that, we considered a network composed by only two neighbor nodes and calculated the number of seconds necessary to send a packet from one node to the other (this is exactly the definition of SERA time slot). The value found was 0.002 seconds for the simulations configured with the parameters showed in Table 5.2.

5.3 Computational results

We divide our results into two categories. First we present a statistical analysis of the networks generated and the path sets obtained by the refined and original routing algorithms. Then, we give the ratio of the refined algorithms' throughput to the throughput of their originals (absolute values are presented in Figures 1, 2 and 3 of the supplementary data file 1). For conciseness purposes, we report only the results for $n = 120$, since the results of the other network sizes shared great qualitative similarities with $n = 120$ results (quantitative differences are discussed in Section 5.4).

5.3.1 Properties of the networks generated

We generated many different network topologies including some close to rings, stars and grids among the 1600 network instances. Such variety of networks helped us to analyze how the network density affects the results of MRA over the original routing algorithms. The few statistics we present here also helped to explain what is observed in relation to the throughput values obtained. For more statistical details, such as topologies' examples and the networks' distributions of node degrees, see Section 3.3.1.

The average number of multiple paths of $\mathcal{P}$ and $\mathcal{P}^R$ given in Table 5.3 was summarized over 10^4 path sets (100 networks $\times$ 100 sets of random origin-destination pair of nodes) produced by each routing algorithm for every combination of n, Δ and $|OD|$. MP-OLSR is absent in this table, because the K parameter is not an upper bound like in AOMDV algorithm, but rather as the fixed value for the number of multiple paths to be found. We observe from the table that the average number of multiple paths increases monotonically with Δ, which is expected from

the well known fact that the number of possible multiple paths grows with Δ in arbitrary graphs [Hof63]. But, the average does not vary significantly across the values of $|OD|$. This is maybe due to the path discovery process we applied. It handles origin-destination pair of nodes separately, so the discovery of a path set cannot be influenced by path sets previously discovered. We also observed that the refined algorithms have always lower average values than the original multi-path ones, which indicates that MRA decreased the number of multiple paths due to the high interference on the neighborhood of each node.

Table 5.3: Average number of multiple paths per origin-destination pair of nodes, where $n = 120$ for 10^4 path sets corresponding to each combination of n, Δ and $|OD|$.

Δ	$\|OD\|$	R-AODV	AOMDV	R-AOMDV	R-OLSR	R-MP-OLSR
4	1 .. 10	1.5	3.3	1.7	1.4	1.5
	11 .. 90	1.5	3.4	1.8	1.5	1.6
	91 .. 120	1.6	3.4	1.8	1.5	1.6
8	1 .. 10	1.6	3.6	1.9	1.6	1.7
	11 .. 20	1.7	3.6	2.1	1.6	1.7
	21 .. 70	1.7	3.7	2.1	1.6	1.7
	71 .. 80	1.7	3.7	2.1	1.6	1.8
	81 .. 120	1.7	3.7	2.2	1.6	1.8
16	1 .. 10	2	3.9	2.9	2	2
	11 .. 50	2.1	3.9	2.9	2	2
	51 .. 90	2.1	4	3	2.1	2
	91 .. 120	2.1	4	3	2.1	2.1
32	1 .. 10	2.2	4.1	3.1	2.2	2.2
	11 .. 60	2.2	4.1	3.1	2.2	2.3
	61 .. 70	2.3	4.2	3.2	2.2	2.4
	71 .. 80	2.3	4.2	3.2	2.2	2.5
	81 .. 90	2.3	4.2	3.2	2.2	2.6
	91 .. 120	2.3	4.2	3.2	2.3	2.7

Figures 5.1 and 5.2 show the networks' path size distribution, respectively for the refined routing algorithms and their originals, again over 10^4 path sets for $|OD| = n$ and $n = 120$. Analyzing sequentially the four panels given for each one of these figures (one panel for each fixed value of Δ), we see that the path size of the distributions' peak value decreases as Δ grows for all routing algorithms. This behavior was anticipated, since the average number of neighbors is related with the size of paths in a graph [Dir52]. Another expected result is that, OLSR and its variants have longest paths than AODVs, because the multi-point relay concept of OLSR (MPR) tends to enlarge paths more than the shortest path algorithm (SPA) used

by AODV. It is also noteworthy that the application of MRA did not change the distributions.

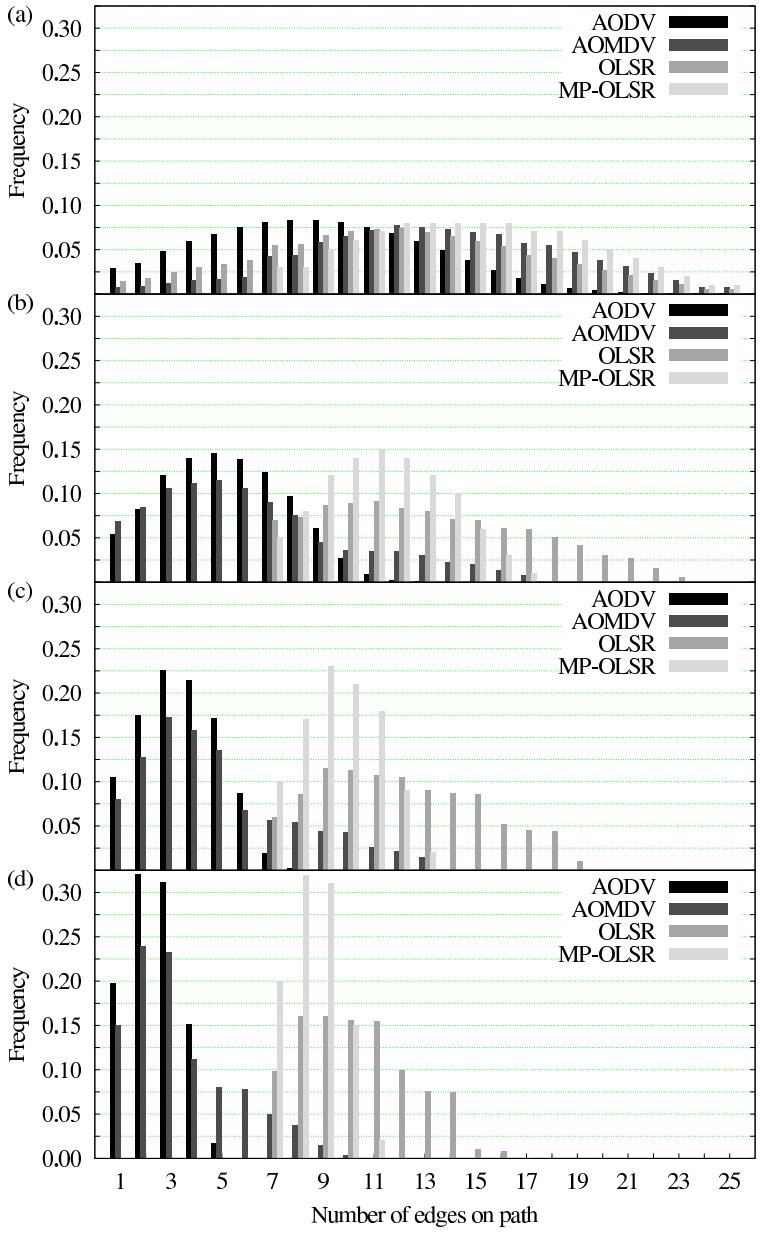

Figure 5.1: Original algorithms' path size histogram with $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). For each value of Δ the distribution refers to 100 networks and 100 path sets per network, each corresponding to n origin-destination pairs.

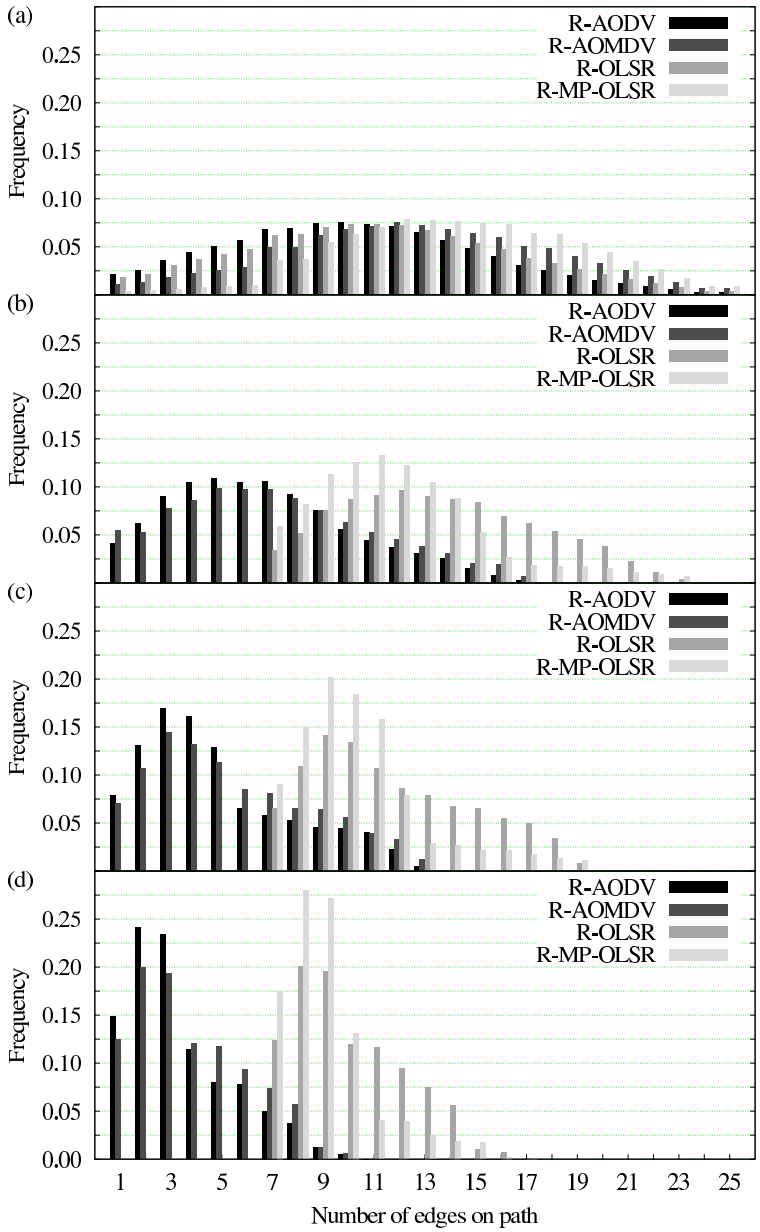

Figure 5.2: Refined algorithms' path size histogram with $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). For each value of Δ the distribution refers to 100 networks and 100 path sets per network, each corresponding to n origin-destination pairs.

5.3.2 Results

Our computational results are summarized in Figures 5.3, 5.4 respectively for NS2 with CBR1 and CBR2 (see Table 5.2), and in Figure 5.5 for SERA. Each figure is organized as a set of four panels, like in Figures 5.1 and 5.2, representing Δ in the non-decreasing order of its fixed values. Each panel shows a plot of θ against the ratio of the refined algorithm throughput to the original throughput (σ). We included a highlighted horizontal line in each plot for $\sigma = 1$ to highlight when a refined algorithm overtakes its original.

Figure 5.3 shows a set of plots with the same structures of panels representing different values of Δ, but now displaying our results for the routing algorithms with CBR1. Recall that, as mentioned in Section 5.2, CBR1 is the rate where the throughput of some paths was sacrificed in order to increase the sum of all path throughputs in the CSMA based network. For a minority of values observed, we see a transient stage in all plots when $\theta < 0.1$. This indicates that some properties related to the throughput vary greatly in these small path sets, such as the average path size and the mutual interference. Moving between panels from $\Delta = 4$ to $\Delta = 32$, we see also that σ slowly decreases as Δ grows and that σ is almost constant when $\theta > 0.5$ for any algorithm and value of Δ. Figure 5.4, on the other hand, shows not very constant σ values, thereby establishing that the network load produced by CBR2 is too low for MRA constantly improve it (the mutual interference is not the core problem in these experiments), even though MRA improved nearly all path sets.

It seems clear from Figure 5.5 that, MRA successfully improved all routing algorithms in a network regulated by a TDMA protocol. This is not valid for a minority of values observed when $\theta < 0.1$, where we observe again the transient stage in all plots. When $\theta \geq 0.1$, there are also a minority of AOMDV σ values, in which part of their confidence intervals is under the threshold line (four of twenty-tree values of R-AOMDV/AOMDV for $\Delta = 32$). When $0.1 \leq \theta \leq 1$, it is worth to observe that all plots have an oscillation behavior, what is not unexpected from ratios (not absolute values). Even so, the widest variation is no greater than 0.16 (OLSR σ values for $\Delta = 16$, between $\theta = 0.8$ and $\theta = 1.5$). Furthermore, if we compare the

plots of each algorithm separately, then we observe different behaviors depending on the algorithm, but σ not vary substantially while changing from one panel to the next. The greater difference between two panels is 0.2% (OLSR σ values for $\theta = 1$, between $\Delta = 4$ and $\Delta = 16$), again out of the transient stage.

As we remarked earlier, MRA proved to increase the path set throughput of all routing algorithms, but this proves nothing about an important performance criterion, not always related to the throughput. The criterion we now shortly describe, known as fairness index [JCH84], is neither a qualitative measure so common in the literature nor a quantitative value designed to some specific cases. Instead, it looks how efficiently is the use of paths to deliver packets by simply measuring the ratio $(\sum_p^{P''} x_p)^2 / P'' \sum_p^{P''} (x_p)^2$, where x_p is the number of packets delivered by the path p and P'' is the number of paths (in our experiments $P'' = |\mathcal{P}|$ or $P'' = |\mathcal{P}^R|$, depending on the experiment). The influence of MRA and the types of algorithms tested over the fairness index is showed in Figures 5.6, 5.7 and 5.8 for the experiments with CBR1, CBR2 and SERA, respectively.

In these figures, R-OLSR and R-AODV (R-AODV, R-OLSR) had a statistical indistinguishable fairness index, even for CBR1. However, we expected that in a heavily loaded network, AODV and OLSR based algorithms had very different values due to SPA and MPR strategies. The same result was also obtained for AODV and OLSR (AODV, OLSR). For each panel presented, the fairness index decreases as θ grows, independent of the protocol (SERA, CBR1 or CBR2), which is expected that larger sets (greater θs) are more likely to unbalance packet flows than smaller ones. Therefore, we can safely assume that the fairness index is affected by the path set size. Moreover, if we compute the fairness index from all paths having the same origin and destination by considering them as a single path (origin-destination fairness index), then we observe the same behavior, although the fairness index is significantly greater. AODV and OLSR are exceptions because they are single-path routing algorithms. An example is shown in Figure 4 of the supplementary data file 1 for the origin-destination fairness index with SERA.

As we fix Δ and θ, Figures 5.6, 5.7, 5.8 revel that SERA has higher fairness index than the others due to the nature of its scheduling mechanism. This mechanism will never let a path suffer to unbounded packet accumulation or absence of

packets. Next, we observe that single-path routing algorithms have the best fairness indexes, since they have at least 50% less paths than the others for the same θ value. However, it is important to note that MRA improved the fairness of all multi-path routing algorithms. It is also interesting to note that, by comparing these figures with Figures 5.3 to 5.4 some times the fairness index even varies inversely with the throughput.

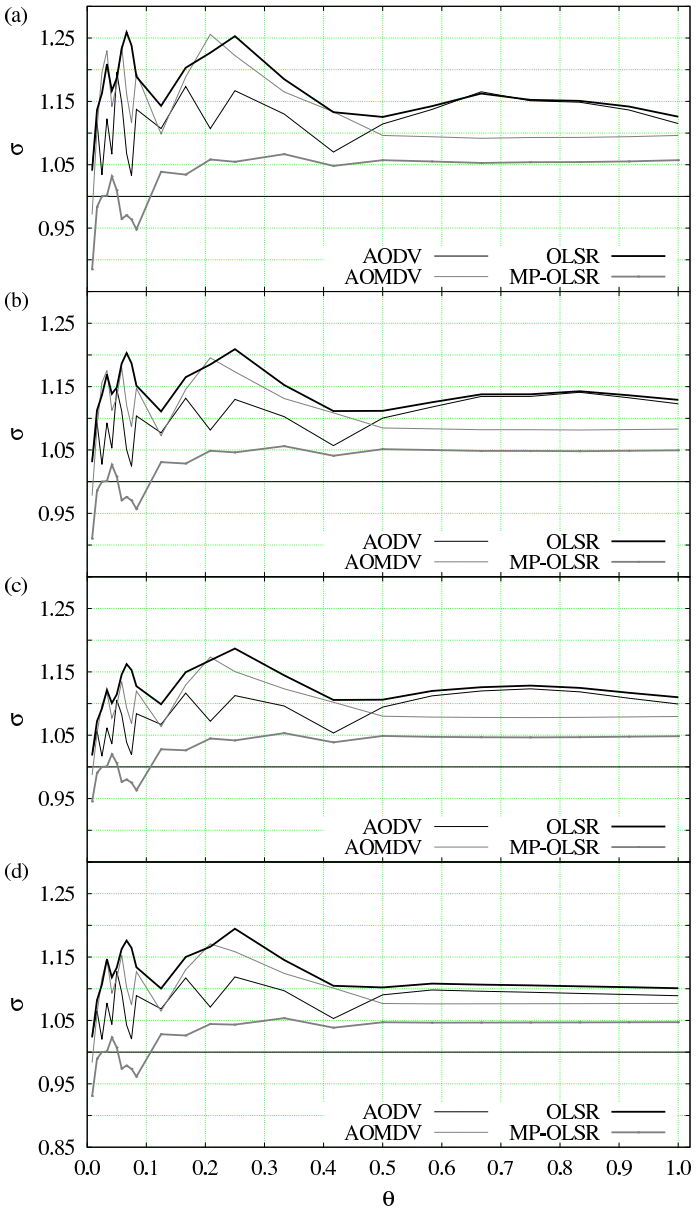

Figure 5.3: Ratio σ for 802.11 with CBR1 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 path sets that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

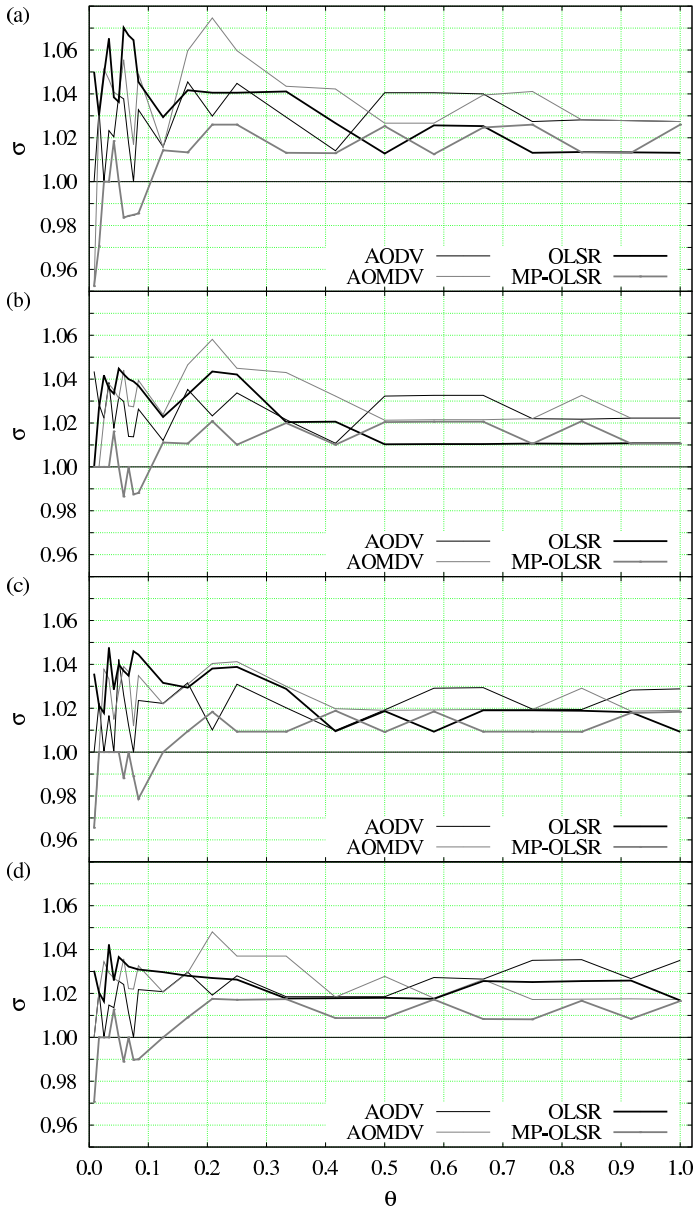

Figure 5.4: Ratio σ for 802.11 with CBR2 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 path sets that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

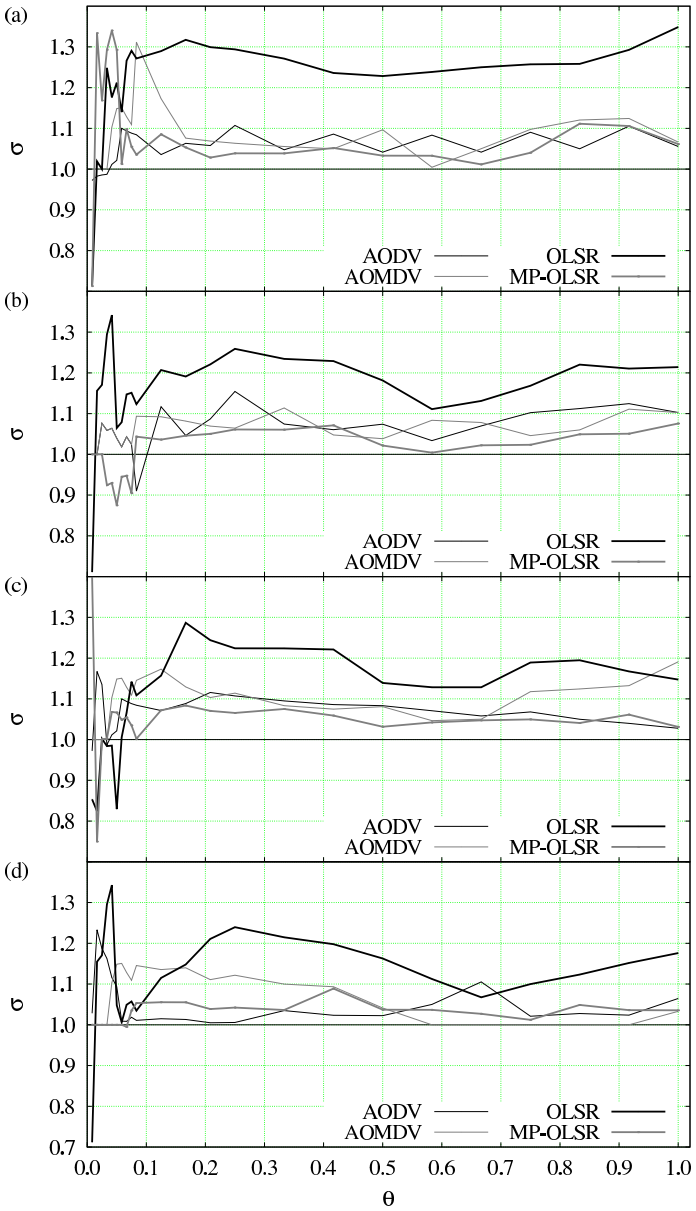

Figure 5.5: Ratio σ for SERA with $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 path sets that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

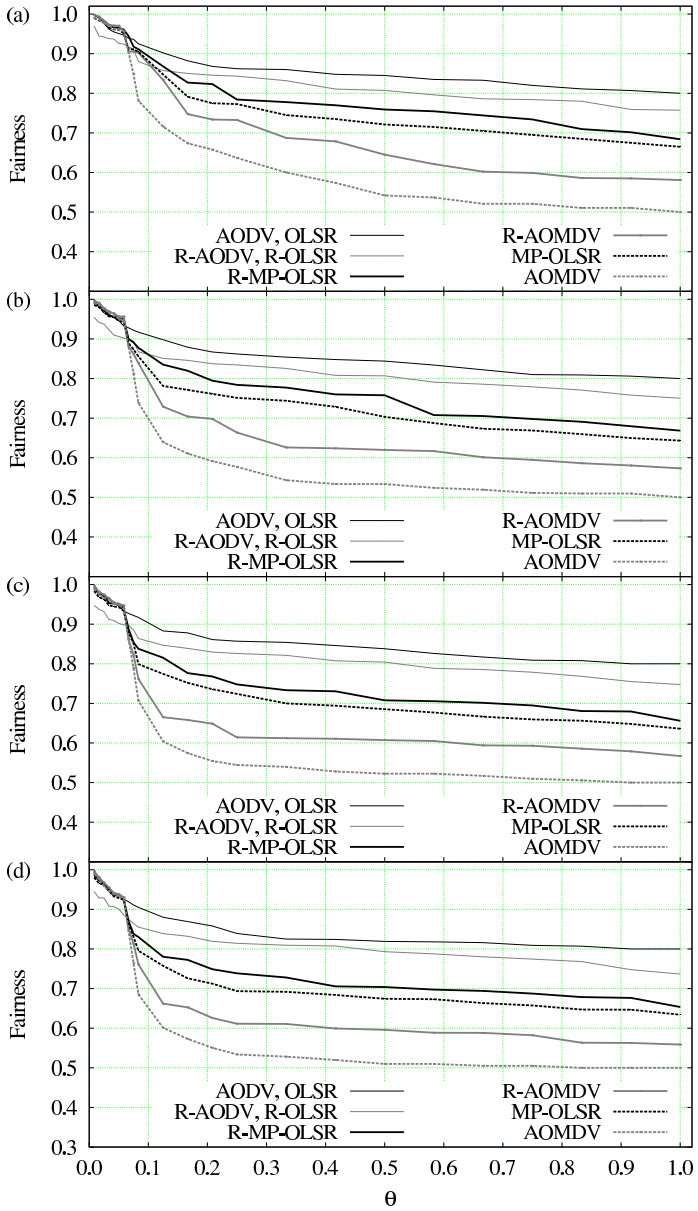

Figure 5.6: The fairness index for 802.11 with CBR1 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 path sets that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

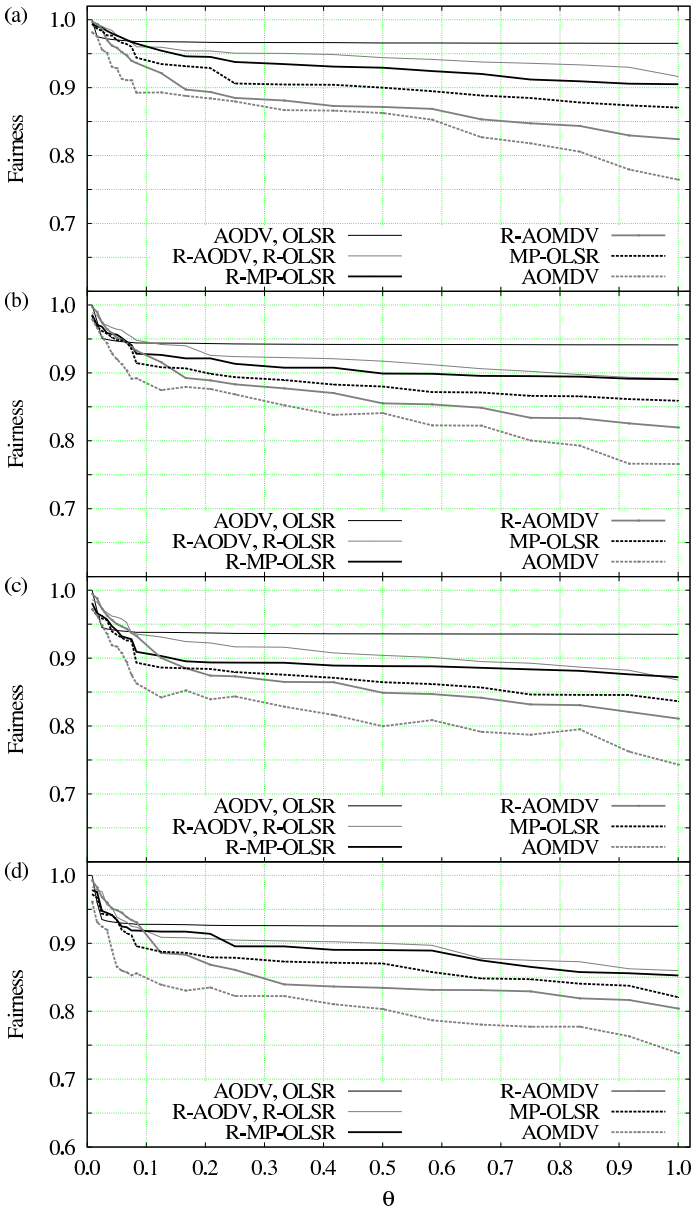

Figure 5.7: The fairness index for 802.11 with CBR2 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 path sets that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

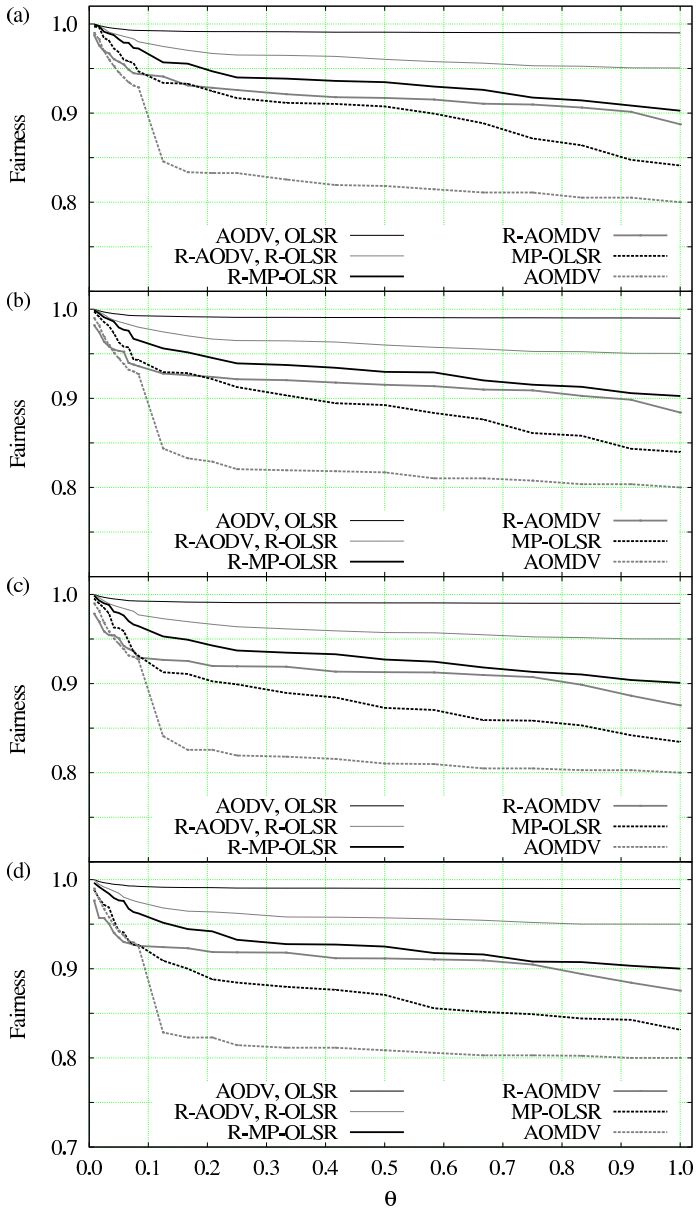

Figure 5.8: The fairness index for SERA with $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 path sets that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

5.4 Discussion

As we briefly discussed before, the absence of the results for $n \in \{60, 80, 100\}$ are justified due to the behavior of MRA (relatively independent from n) and the excessive amount of figures produced by the results. In such instances, those results where $n = 60$ can be seen, for example, as a sub-section of $n = 120$ results. More precisely, if we crop all plots of the Figure 5.3 at $\theta = 0.5$, then we would see no difference in the order of the plots (first OLSR followed by AOMDV, next AODV and at last MP-OLSR), but σ values for $n = 60$ are greater than those for $n = 120$. This stems not only from the fact that the average path size decreases as n decreases, but more generally from the fact that the number of links for the same θ is lower for the networks where $n = 60$ than for those where $n = 120$ (less links to interfere and less to be schedule). Thus, the improvement obtained by MRA seems to be influenced not only by the network density (Δ) and θ, but also by the network path size distribution.

For the fairness index, the results for $n \in \{60, 80, 100\}$ were not included for the same reasons. As σ, it slowly decreases as n grows, but we cannot confirm this relation because the differences are not significative for some routing algorithms. The fairness index also decreases as Δ grows, which suggests that the behavior of the fairness index depends more on other factors, such as the network load and the path size distribution.

Although investigating the performance issues of the routing algorithms is not our objective, the implicit ranking among them is interesting to discuss (see Figures 5.3 to 5.8). Considering that OLSR and MP-OLSR were the most advantageous to be refined under heavy loaded network scenarios (SERA and CBR1), we observe that they have less multiple paths than the others and more longer paths than the others (see Table 5.3). To see that this is so, first recall that OLSR longer paths are a direct consequence of MPR and the number of multiple paths of a node is somewhat inversely related to the interference level of its neighborhood, as we briefly discussed in Section 5.3.1. So, R-OLSR and R-MP-OLSR have better results because MRA and the MRPs. MRA selects less multiple paths than available, thus it decreases the mutual interference in the neighborhood of each node. MPRs guarantees the spacial

separation necessary to avoid that paths interfere with each other in the middle-way to their destinations. Moreover, we suspect that AODVs have paths spatially closer due to SPA and consequently with more mutual interference than OLSRs (MPR forces the discovery of more isolated paths) [BMSC10, YCHP08]. It is also curios to observe another ranking (see Figures 5, 6 and 7 of the supplementary data file 1), in which all routing algorithms achieved better throughput results with SERA (a TDMA protocol) than with CBR1 (a CSMA protocol). The same is true for the fairness index when SERA is compared to CBR2. This may be, in principle, a manifestation of SERA advantages rather than of the superiority of TDMA over CSMA protocols, since a considerable amount of research directed toward the matter did not reach an agreement [DUR09, BM09, GP07, DZMS02].

Another curiosity, now regarding all NS2 simulations with the two configurations described in Section 5.2 (CBR1 and CBR2), we detected that some paths had transmitted no packets even if we extended the simulation time, reordered and removed the synchronization of the multiple CBRs (common problems of NS2 agents). We verified by executing another set experiments that 802.11 CSMA was discarding some paths to increase the throughput of others. Figures 8 and 9 of the supplementary data file 1 shows the distributions of origin-destination pair whose paths did not received packets. Obviously, the single-path routing algorithms had less paths without packets than the multi-paths ones, however R-MP-OLSR and R-AOMDV had less paths without packet than their originals (MRA attenuated the number of paths without packets).

Chapter 6

Conclusion

We now present our conclusions over the two methods we used for link scheduling (discussed in Chapters 1, 2 and 3) and over the refinement method of multi-path sets we introduced and discussed in Chapters 4 and 5, respectively.

6.1 SER and SERA

Algorithms SER and SERA are methods for link scheduling in WMNs. As such, and unlike other methods for link scheduling, they are built around a set of origin-to-destination paths and aim to provide as much throughput on these paths as possible. From a mathematical perspective they are both related to providing the nodes of a graph with an efficient multicoloring, in the sense discussed in Section 1.3. For SER this is strictly true, but for SERA the defining characteristic of a multicoloring, that each node receives the same number of colors, ceases to hold. As we demonstrated through our computational results in Section 3.3, it is precisely this deviation from the strict definition that allows SERA to surpass SER in terms of performance.

The functioning of both SER and SERA is supported by the use of the integer parameter $B \geq 1$, which indicates how many buffering positions each WMN node has to store in-transit packets for each of the paths that go through it. Choosing $B = 1$ suffices for SER because of its inherent property of alternating interfering transmissions, but $B > 1$ may in principle be needed for the advantages of SERA to become manifest. In the simulations we conducted, however, only rarely has this been the case, since on average increasing B beyond 1 provided no distinguishable

improvement. In this regard, we find it important for the reader to refer to Figure 2.2 once again. As we remarked upon discussing that figure, profiting from a $B > 1$ situation under SERA is largely a matter of how uniformly interference gets distributed on the particular set of paths at hand. Our results in Section 3.3, therefore, can safely be assumed to have stemmed from circumstances that, on average, led to highly uniformly distributed interference patterns.

The centerpiece of both SER and SERA is the undirected graph G, which embodies a representation of all the interference affecting the various wireless links represented by the graph's nodes. As we explained in Section 1.2, the steps to building G depend on how one assumes the communication and interference radii to relate to each other, and also on which interference model is adopted. We have given results for a specific set of assumptions, but clearly there is nothing in either method precluding its use under any other assumptions, including for example those of the physical interference model, which incidentally have been used recently in solving problems related to the one we have considered [CCF+10a, CCF+10b, CCGY11]. Whenever different assumptions are in place, all that needs to be done is construct G accordingly.

Analyzing either method mathematically is a difficult enterprise, but since their performance depends on the heuristic choice of an initial acyclic orientation of G, any effort profitably spent in that direction will be welcome. In addition to potentially better decisions regarding initial conditions, further mathematical knowledge on SER or SERA may also come to provide a deeper understanding of how upper bounds on $T(\mathcal{S})$ relate to what is observed. As we mentioned in Section 3.4, one such bound is already known in the case of SER. Obtaining better bounds in this case, as well as some bound in the case of SERA, remains open to further research.

Another issue that is open to further investigation is how to handle the potential difficulties that SER may encounter in the face of a growing number of nodes in G [MR92, MMZ93]. These difficulties refer to the fact that, in the worst case, the time required to detect the occurrence of the period may grow exponentially with the square root of the number of nodes. They are inherited by SERA, since it generalizes SER, and may require the development of further heuristics if they pose a real problem in practice. In a related vein, sometimes it may be the case that

only the value of $T(\mathcal{S})$ is needed, not $\mathcal{S}$ itself. Knowing the achievable throughput without requiring knowledge of the schedule itself can be useful for evaluating WMN topologies or routing algorithms for them.

Should this be the case, then it is possible to estimate $T(\mathcal{S})$ more efficiently than using the full-fledged algorithms we gave. We can do this in the case of SERA by recognizing that $T(\mathcal{S})$ is the limit, as $t \to \infty$, of

$$T_t(\mathcal{S}) = \frac{\sum_{i \in T} m_i(\omega_0, t)}{t+1}, \tag{6.1}$$

where $m_i(\omega_0, t)$ is the total number of times node i appears as a sink in orientations $\omega_0, \omega_1, \ldots, \omega_t$. To see this, let $o(t)$ denote any function of t such that $\lim_{t\to\infty} o(t)/t = 0$. We then have $\sum_{i \in T} m_i(\omega_0, t) = r(t) \sum_{i \in T} m_i(\omega_0) + o(t)$ and $t+1 = r(t)p(\omega_0) + o(t)$, where $r(t)$ is the number of times the period has been repeated up to iteration t. The limit follows easily, and automatically holds also for SER by straightforward extension. The streamlined version of either algorithm consists simply of letting t evolve either through a sufficiently large value determined beforehand or until $T_t(\mathcal{S})$ becomes stable. Any of the two alternatives does away with the need to detect the occurrence of the period.

We also note, that we have found the results given in Figure 3.5 to be practically indistinguishable from those obtained through the strategy outlined above for the computation of $T(\mathcal{S})$. We have verified this by letting $T(\mathcal{S}) = T_{t^+}(\mathcal{S})$, where t^+ is the least value of t for which $|T_t(\mathcal{S}) - T_{t-w}(\mathcal{S})|/T_{t-w}(\mathcal{S}) \leq 0.001$. Here w is a window parameter and in our experiments we used $w = |N|$. As for this particular choice, it comes from realizing that in both SER and SERA it takes at most $|N| - 1$ iterations for a node of G that is currently not a sink to become one. This, in turn, comes from the fact that in each iteration either algorithm necessarily decreases by 1 the number of edges on a longest directed path from any non-sink node to a sink. We can see that this is true of SER by viewing its dynamics in terms of how the orientations' sink decompositions evolve. We can see that it continues to hold in the case of SERA because SERA never places a former sink i into one of the sink-decomposition sets that already contains a neighbor of i in G (cf. Figure 2.3).

6.2 MRA

When MRA is used, it follows from its definition in Section 4.2 that it builds a P_{inf} approximation instead of the exact solution (recall that P_{ind} is a P_{inf} under the assumptions of the protocol interference model). This is due to the difficult enterprise of implementing complex changes on any of the well established routing algorithms without overload the network with more control messages. Moreover, the optimum P_{ind} involves the exact solution of NP-hard problems, as discussed in Section 4.1. Besides the simplicity of MRA solution, it was demonstrated by the distinguishable improvements (throughput and fairness) obtained on the computational results that MRA produces a good approximation of P_{inf}. For instance, OLSR throughput was improved over 20% with SERA and over 15% with CBR1 (average values over all experiments), and with superior fairness index than the original. Of course, this was achieved in some part because MRA refined an already node disjoint path set (recall that we used only the node disjoint version of the routing algorithms), but mostly because the original routing algorithms do not take into account the interference information like MRA.

Another important advantage of MRA is that even involving the NP-hard problem of the maximum weighted independent set, the average neighborhood size of a wireless node is sufficient small to be computed in negligible time compared to arbitrary graphs [PD91]. This leads to an open issue that is how paths from different origins and destinations affect themselves, since the exact solution of P_{inf} does not involve the maximum weighted independent set of the network. Analyzing this involves unpractical instances of this problem and moreover, we have to consider many aspects such as which nodes really need multiple paths by considering the distribution of the network load. Also, we have to consider how many multiple paths are necessary for each node, based on the interference they might produce over the neighborhood. These aspects are even more difficult to analyze due to the unknown relation among them.

Another aspect open for further investigation is the benefits of joining multi-radio network and MRA. In these schemes, orthogonal frequencies can drastically reduce the number of D edges and increase the number of multiple paths free of

interference. MRA can be benefited of this variety and it can minimize the number of radios, since it selects a subset of paths [BAPW04, RC05].

We also remark that, our work depends in great part on the relationship that exists among the variety of interference free paths and the network density, thus the availability of these paths affects the results of MRA more than the size of the network paths. Additionally, the mutual interference is caused by the proximity among paths and not by the path sizes, hence the objective is to discover paths sufficiently spatial separated to avoid interference, but not too far to loose throughput from longer paths.

We note, finally, that joint SERA and MRA over some routing protocol (like OLSR) is a intuitively step that we shall consider for a future work, and in addition all open aspects discussed in this chapter.

Appendix A

Supplemental data file 1

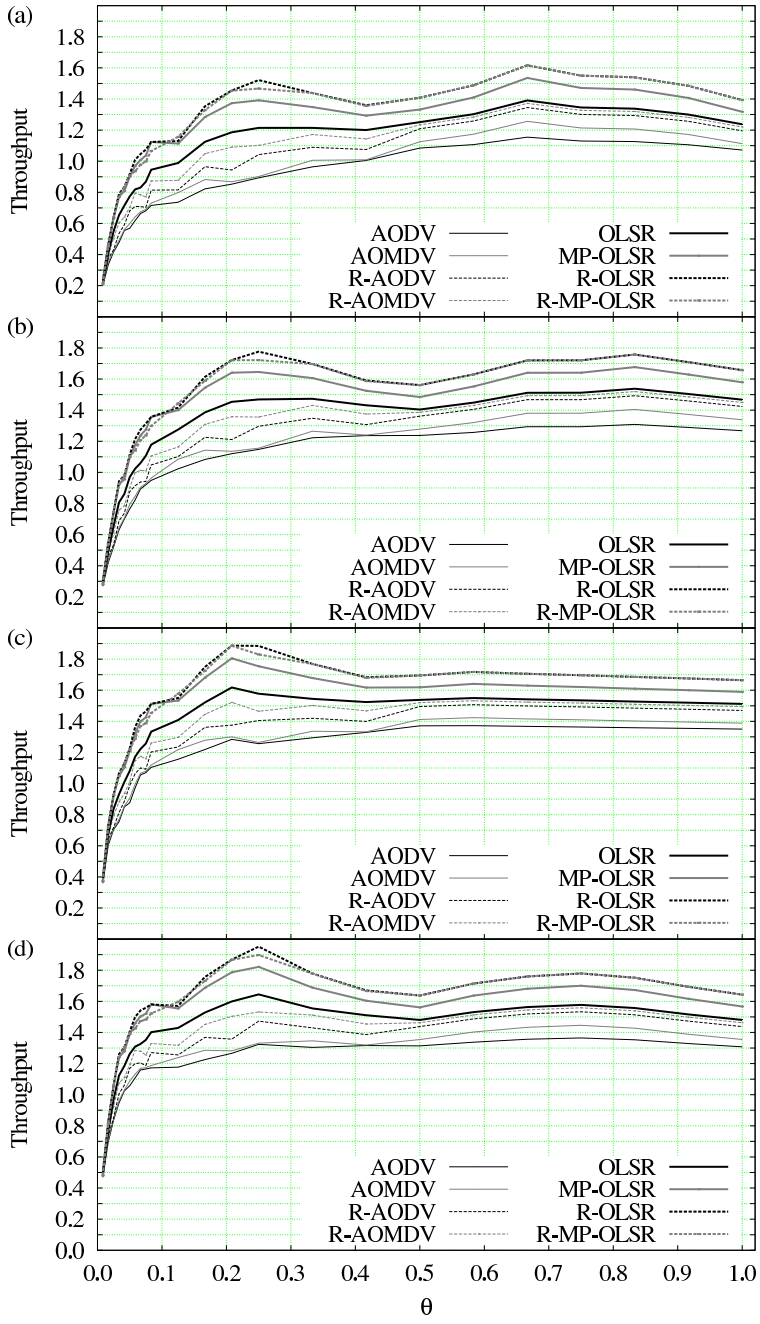

Figure A.1: Throughput (packets/time slot) for CBR1 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

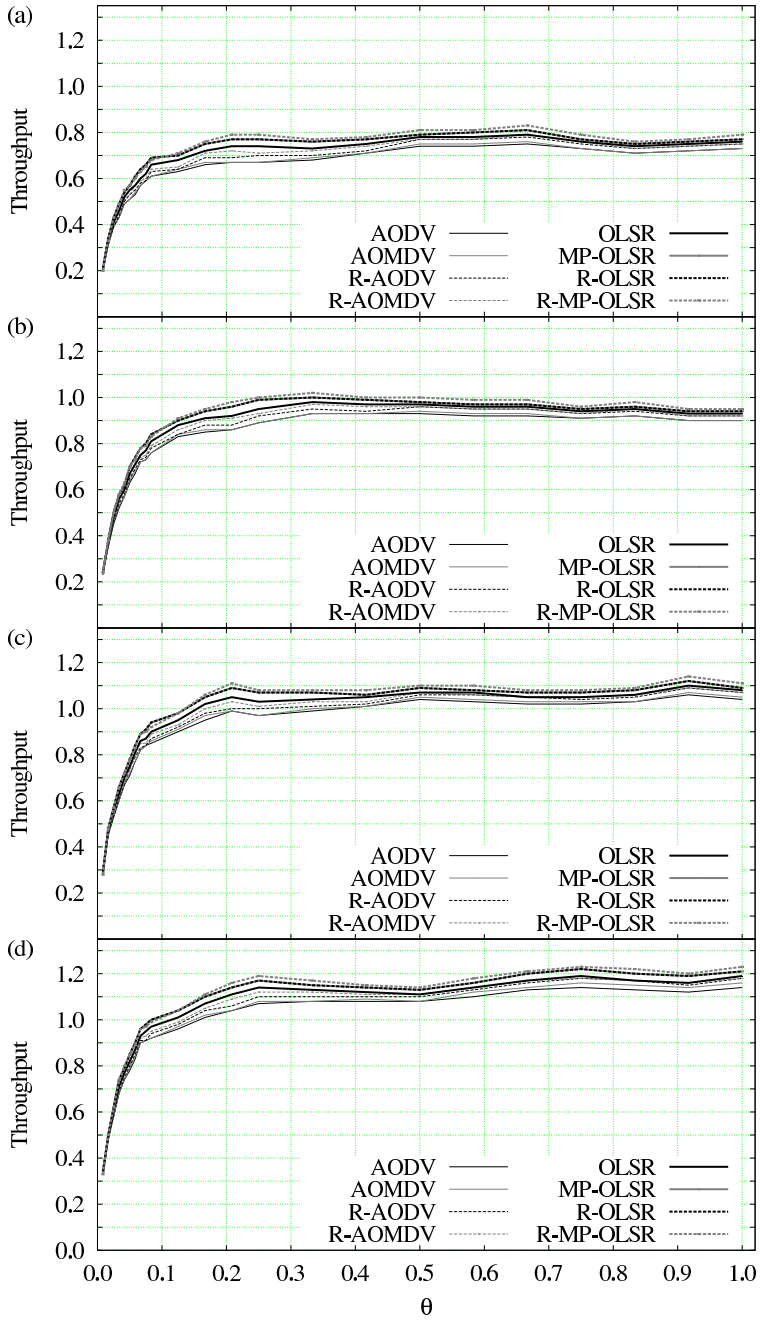

Figure A.2: Throughput (packets/time slot) for CBR2 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

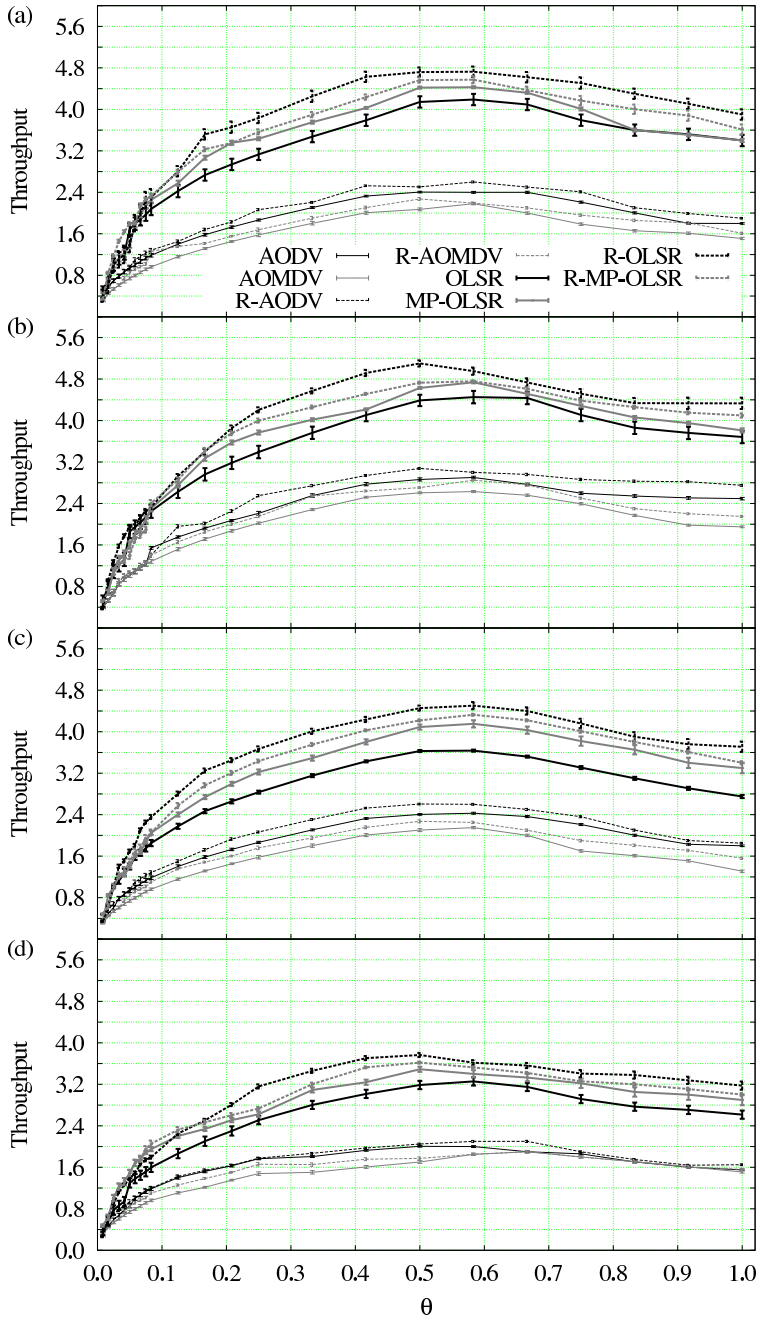

Figure A.3: Throughput (packets/time slot) for SERA with $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ for each value of θ. Error bars are based on confidence intervals at the 95% level.

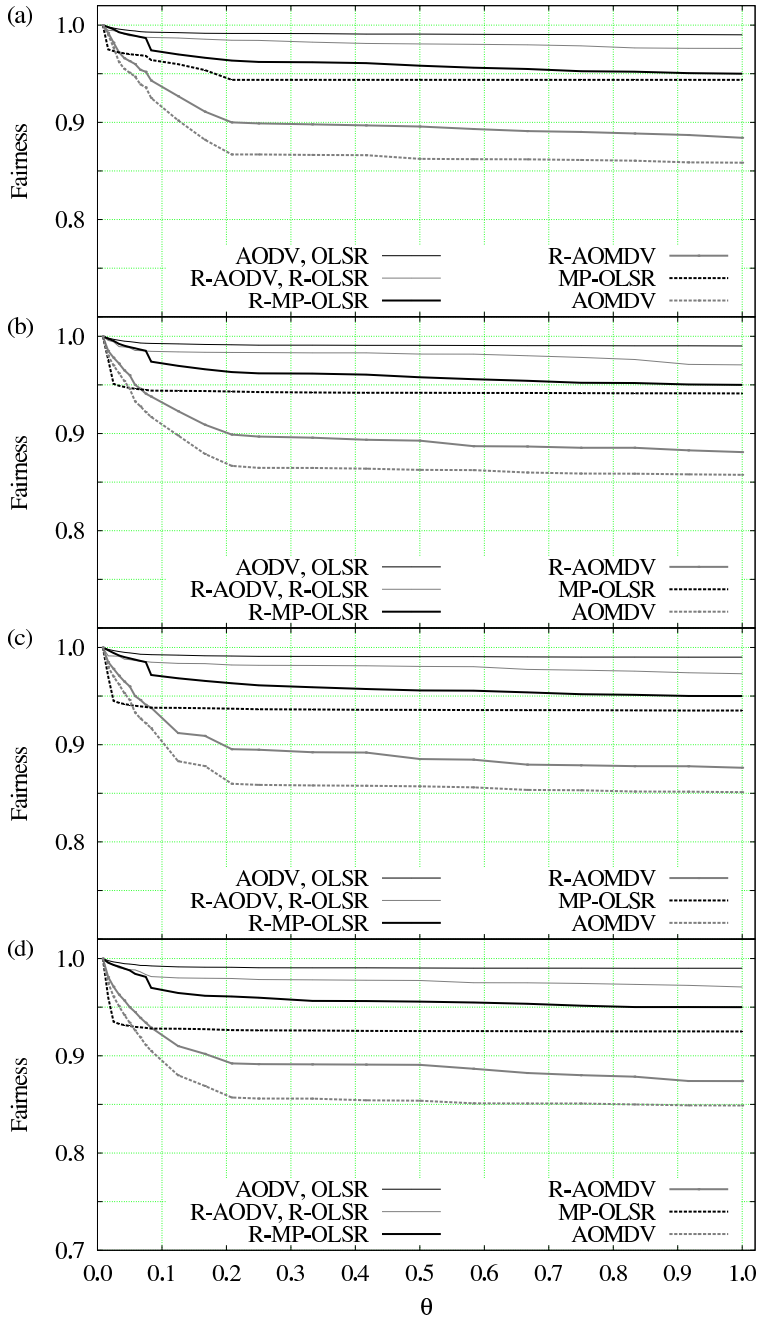

Figure A.4: The origin-destination fairness index for SERA with $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

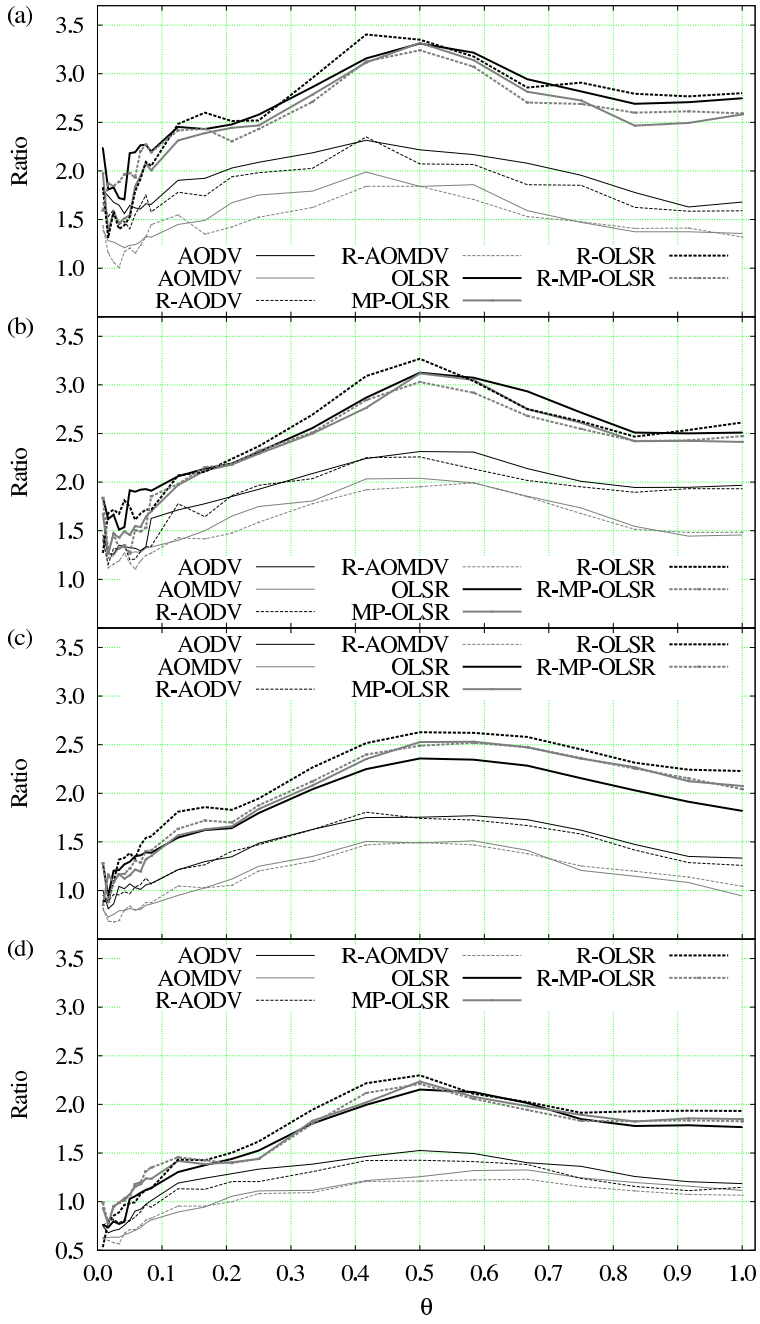

Figure A.5: Ratio (throughput SERA)/(throughput CBR1) for $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

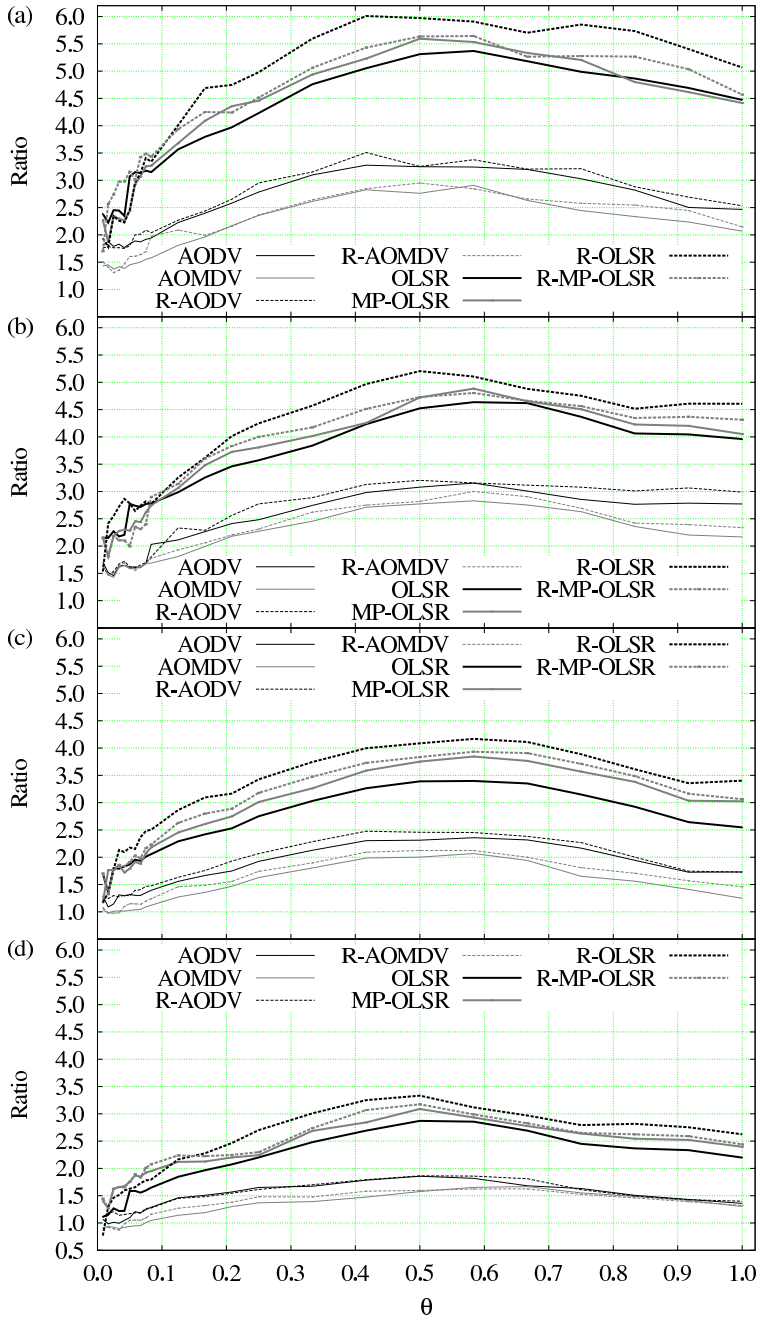

Figure A.6: Ratio (throughput SERA)/(throughput CBR2) for $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

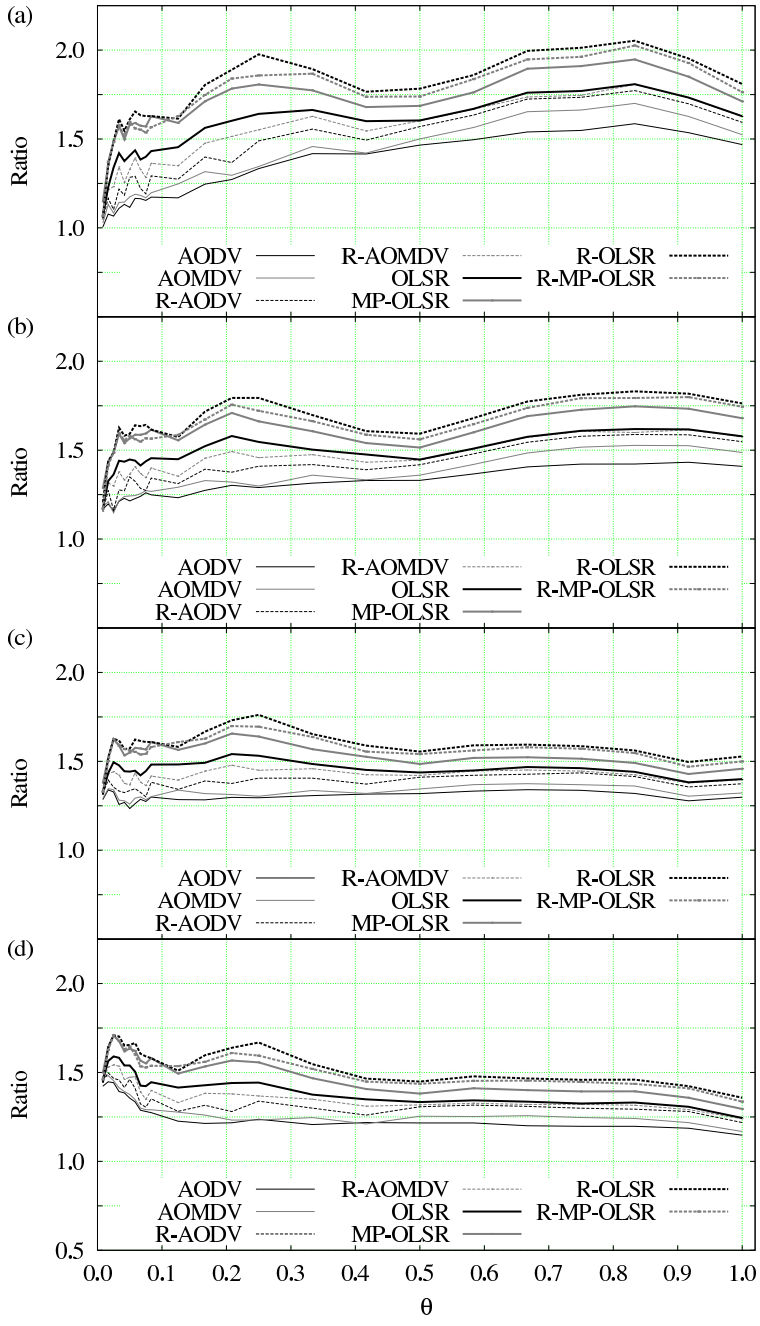

Figure A.7: Ratio (throughput CBR1)/(throughput CBR2) for $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ for each value of θ. Error bars were omitted, since they are less than 1% of the mean for confidence intervals at the 95% level.

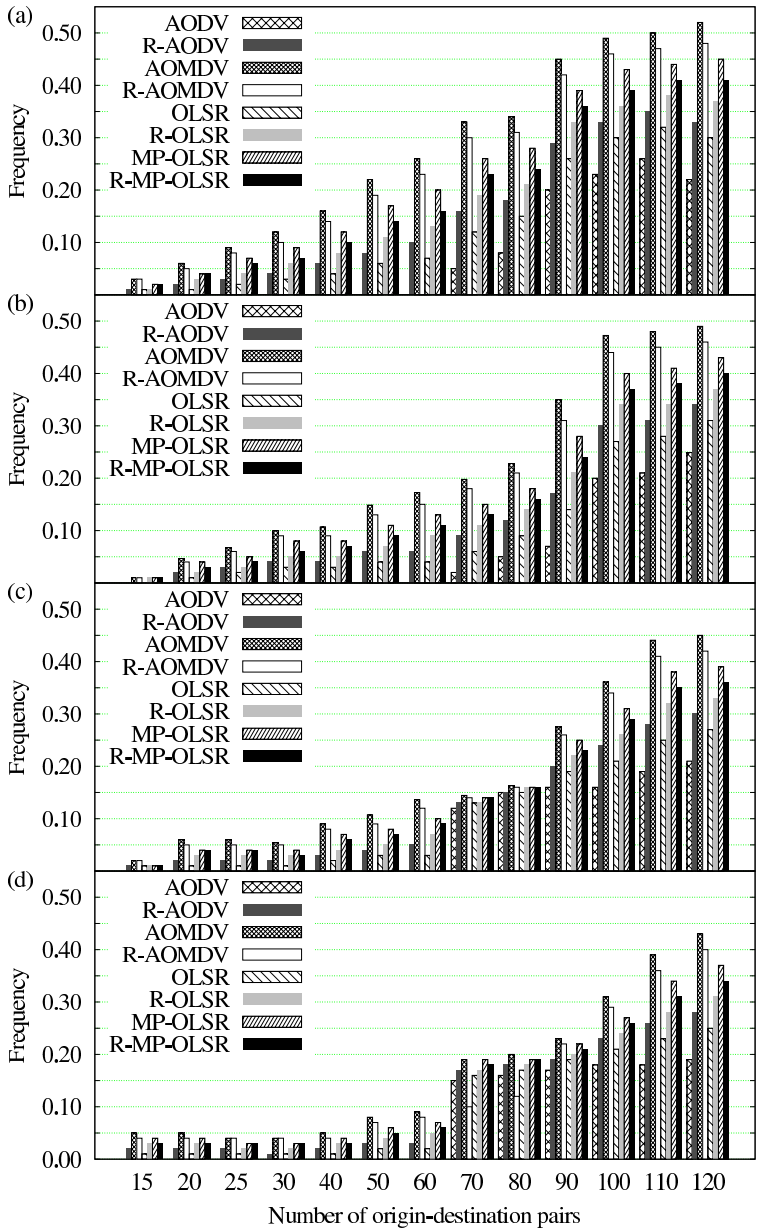

Figure A.8: Paths without packets histogram for CBR1 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ.

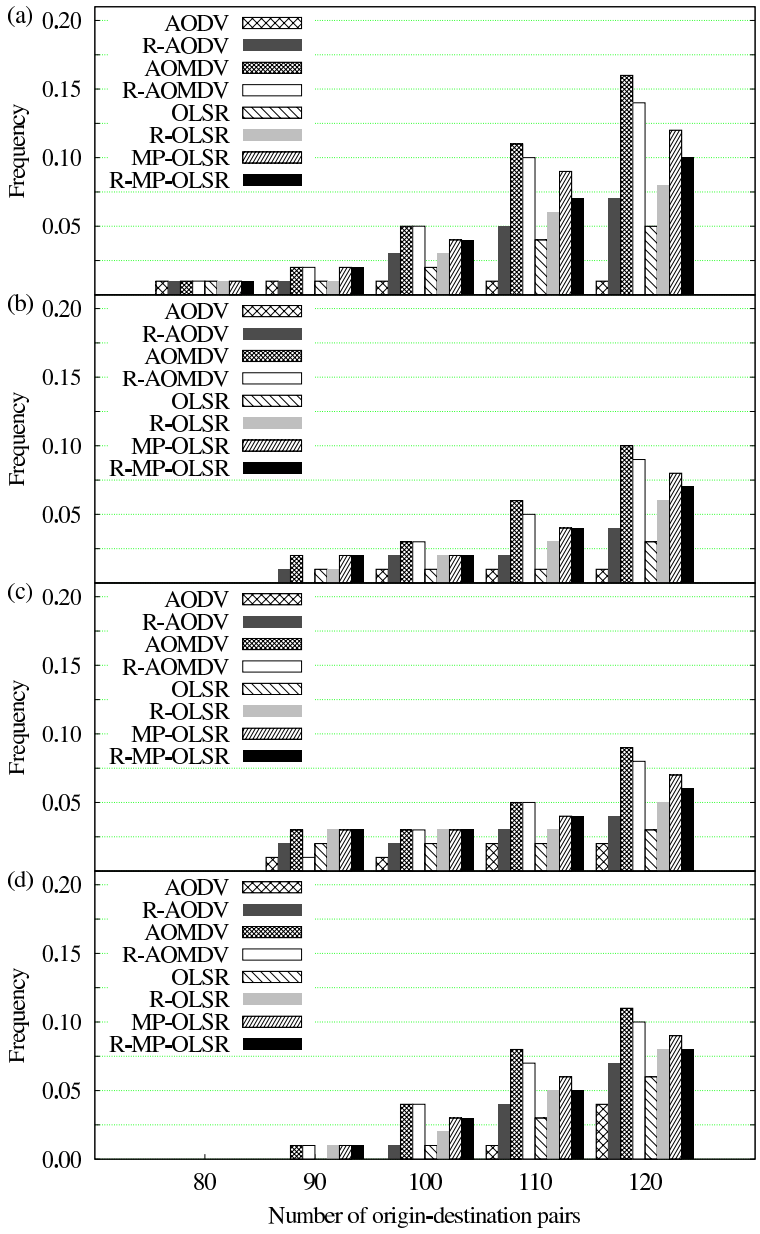

Figure A.9: Paths without packets histogram for CBR2 where $n = 120$ for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). Data are averages over the 10^4 sets of paths that correspond to each value of Δ.

List of Figures

1.1 A set of $P = 3$ directed paths (a) and the resulting directed multigraph D (b). 17
1.2 A set of $P = 3$ directed paths (a) and the resulting directed multigraph D (b). 19
1.3 The graph-transformation process. 21
1.4 A set of $P = 4$ paths (a), with dashed lines indicating all node pairs representing off-path interference. 25

2.1 The functioning of SER when G is the 5-node cycle and ω_0 is the leftmost orientation in the top row. 28
2.2 A set of $P = 3$ directed paths (a), the resulting directed multigraph D (b), and the resulting undirected graph G (c). 30
2.3 Each set in a sink decomposition is represented by a rectangular box and numbered to indicate the set's subscript. 32

3.1 A sampler of the networks that were generated for $n = 80$. 39
3.2 Degree distributions of the 1600 networks, for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). 40
3.3 Path-size (number of edges) distributions of the 1600 networks, for $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). 41
3.4 Behavior of $T(\mathcal{S})$ for SER under the two numbering schemes ND-BF and ND-DF, with $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). 44
3.5 Behavior of $T(\mathcal{S})$ for SERA under the numbering scheme ND-BF, with $\Delta = 4$ (a), $\Delta = 8$ (b), $\Delta = 16$ (c), and $\Delta = 32$ (d). 45
3.6 Behavior of $T(\mathcal{S})$ for SER and SERA with $n = 10$ and $\Delta = 4$. 48

4.1 A graph representation of a wireless network. 51
4.2 A graph model of a wireless network, in which an independent path set is represented by dashed lines. 53

4.3 The reduction of G_{ij} from a given path set. 54
4.4 The construction of the graph D_{ij}. 55

5.1 Original algorithms' path size histogram. 65
5.2 Refined algorithms' path size histogram. 66
5.3 Ratio σ for 802.11 with CBR1. 69
5.4 Ratio σ for 802.11 with CBR2. 70
5.5 Ratio σ for SERA. 71
5.6 The fairness index for CRB1. 72
5.7 The fairness index for CRB2. 73
5.8 The fairness index for SERA. 74

A.1 Throughput (packets/time slot) for CBR1. 84
A.2 Throughput (packets/time slot) for CBR2. 85
A.3 Throughput (packets/time slot) for SERA. 86
A.4 The origin-destination fairness index for SERA. 87
A.5 Ratio (SERA)/(CBR1). 88
A.6 Ratio (SERA)/(CBR2). 89
A.7 Ratio (CBR1)/(CBR2). 90
A.8 Origin-destination pairs without packets histogram for CBR1. 91
A.9 Origin-destination pairs without packets histogram for CBR2. 92

Bibliography

[ABL06] M. Alicherry, R. Bhatia et L. E. Li : Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. *IEEE J. Sel. Areas Commun.*, 24(11):1960–1971, 2006.

[Abo04] M. Abolhasan : A review of routing protocols for mobile ad hoc networks. *Ad Hoc Netw.*, 2(1):1–22, 2004.

[ACSR10] C. H. P. Augusto, C. B. Carvalho, M. W. R. Silva et J. F. Rezende : The impact of joint routing and link scheduling on the performance of wireless mesh networks. *In Proceedings of the IEEE LCN 2010*, pages 80–87, 2010.

[AWW05] I. F. Akyildiz, X. Wang et W. Wang : Wireless mesh networks: a survey. *Comput. Netw.*, 47(4):445–487, march 2005.

[BAPW04] P. Bahl, A. Adya, J. Padhye et A. Walman : Reconsidering wireless systems with multiple radios. *Comput. Commun. Rev.*, 34:39–46, 2004.

[Bar96] V. C. Barbosa : *An Introduction to Distributed Algorithms*. The MIT Press, Cambridge, MA, 1996.

[Bar00a] V. C. Barbosa : *An Atlas of Edge-Reversal Dynamics*. Chapman & Hall/CRC, London, UK, 2000.

[Bar00b] V. C. Barbosa : The interleaved multichromatic number of a graph. *Ann. Comb.*, 6:249–256, 2000.

[BBK+04] H. Balakrishnan, C. L. Barrett, V. S. A. Kumar, M. V. Marathe et S. Thite : The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc wireless networks. *IEEE J. Sel. Areas Commun.*, 22:1069–1079, 2004.

[BG89] V. C. BARBOSA et E. GAFNI : Concurrency in heavily loaded neighborhood-constrained systems. *ACM Trans. Program. Lang. Syst.*, 11:562–584, 1989.

[BM08] J. A. BONDY et U. S. R. MURTY : *Graph Theory*. Springer, New York, NY, 2008.

[BM09] S. BANAOUAS et P. MUHLETHALER : Performance evaluation of TDMA versus CSMA based protocols in SINR models. *In Proceedings of the EW 2009*, pages 113–117, 2009.

[BMSC10] S. R. BIRADAR, K. MAJUMDER, S. K. SARKAR et Puttamadappa C. : Performance evaluation and comparison of AODV and AOMDV. *Int. J. Comput. Sci. Eng.*, 2(2):373–377, 2010.

[BR03] A. BEHZAD et I. RUBIN : On the performance of graph-based scheduling algorithms for packet radio networks. *In Proceedings of IEEE GLOBECOM 2003*, pages 3432–3436, 2003.

[BVB05] A. BALACHANDRAN, G. M. VOELKER et P. BAHL : Wireless hotspots: current challenges and future directions. *Mob. Netw. Appl.*, 10:265–274, 2005.

[CCF+10a] A. CAPONE, G. CARELLO, I. FILIPPINI, S. GUALANDI et F. MALUCELLI : Routing, scheduling and channel assignment in wireless mesh networks: optimization models and algorithms. *Ad Hoc Netw.*, 8:545–563, 2010.

[CCF+10b] A. CAPONE, G. CARELLO, I. FILIPPINI, S. GUALANDI et F. MALUCELLI : Solving a resource allocation problem in wireless mesh networks: a comparison between a CP-based and a classical column generation. *Networks*, 55:221–233, 2010.

[CCGY11] A. CAPONE, L. CHEN, S. GUALANDI et D. YUAN : A new computational approach for maximum link activation in wireless networks under the SINR model. *IEEE Trans. Wirel. Commun.*, 10:1368–1372, 2011.

[CEM+08] M. E. M. CAMPISTA, P. M. ESPOSITO, I. M. MORAES, L. H. M. COSTA, O. C. M. DUARTE, D. G. PASSOS, C. V. N. ALBUQUERQUE, D. C. M. SAADE et M. G. RUBINSTEIN : Routing metrics and protocols for wireless mesh networks. *IEEE Netw.*, 22(1):6–12, 2008.

[CQYM00] Y. CHUN, L. QIN, L. YONG et S. MEILIN : Routing protocols overview and design issues for self-organized network. *In Proceedings of the WCC ICCT 2000*, volume 2, pages 1298–1303, 2000.

[CS03] R. L. CRUZ et A. V. SANTHANAM : Optimal routing, link scheduling and power control in multihop wireless networks. *In Proceedings of the IEEE INFOCOM 2003*, volume 1, pages 702–711, 2003.

[DEA06] I. DEMIRKOL, C. ERSOY et F. ALAGOZ : MAC protocols for wireless sensor networks: a survey. *IEEE Commun. Mag.*, 44(4):115–121, 2006.

[Dir52] G. A. DIRAC : Some theorems on abstract graphs. *Proc. Lond. Math. Soc.*, s3-2(1):69–81, 1952.

[DUR09] A. DHEKNE, N. UCHAT et B. RAMAN : Implementation and evaluation of a TDMA MAC for wifi-based rural mesh networks. *In Proceedings of the ACM SOSP 2009*, 2009.

[DZMS02] J. DING, L. ZHAO, S. R. MEDIDI et K. M. SIVALINGAM : MAC protocols for ultra-wide-band (UWB) wireless networks: impact of channel acquisition time. *In Proceedings of the SPIE ITCOM 2002*, pages 1953–1954, 2002.

[GDP08] S. GANDHAM, M. DAWANDE et R. PRAKASH : Link scheduling in wireless sensor networks: distributed edge-coloring revisited. *J. Parallel Distrib. Comput.*, 68:1122–1134, 2008.

[GK00] P. GUPTA et P. R. KUMAR : The capacity of wireless networks. *IEEE Trans. Inf. Theory*, 46:388–404, 2000.

[GL00] A. Chandra V. GUMMALLA et J. O. LIMB : Wireless medium access control protocols. *IEEE Commun. Surv. Tutor.*, 3(2):2–15, 2000.

[GLS81] M. GRÖTSCHEL, L. LOVÁSZ et A. SCHRIJVER : The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[GP07] G. P. GUPTA et A. K. PANDEY : Performance comparison of ad hoc routing protocols over IEEE 802.11 DCF and TDMA MAC layer protocols. *In Proceedings of the NCC 2007*, volume 1, pages 183–187, 2007.

[GWHW09] O. GOUSSEVSKAIA, R. WATTENHOFER, M. M. HALLDORSSON et E. WELZL : Capacity of arbitrary wireless networks. *In Proceedings of IEEE INFOCOM 2009*, pages 1872–1880, 2009.

[HL08] Q.-S. HUA et F. C. M. LAU : Exact and approximate link scheduling algorithms under the physical interference model. *In Proceedings of DIAL M-POMC 2008*, pages 45–54, 2008.

[Hof63] A. J. HOFFMAN : On the polynomial of a graph. *Amer. Math. Month.*, 70(1):30–36, 1963.

[JCH84] R. JAIN, D.-M. CHIU et W. HAWE : A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Rapport technique, Digital Equipment Corporation, 1984.

[JMC+01] P. JACQUET, P. MUHLETHALER, T. CLAUSEN, A. LAOUITI, A. QAYYUM et L. VIENNOT : Optimized link state routing protocol for ad hoc networks. *In Proceedings of the IEEE INMIC 2001*, pages 62–68, 2001.

[Kar72] R. M. KARP : Reducibility among combinatorial problems. *In* R. E. MILLER et J. W. THATCHER, éditeurs : *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, NY, 1972.

[KLF06] S. KTARI, H. LABIOD et M. FRIKHA : Load balanced multipath routing in mobile ad hoc network. *In Proceedings of the IEEE ICCS 2006*, pages 1–5, 2006.

[KLH06] T.-S. KIM, H. LIM et J. C. HOU : Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. *In Proceedings of the MobiCom 2006*, pages 366–377, New York, NY, 2006. ACM.

[LC04] X. LI et L. CUTHBERT : On-demand node-disjoint multipath routing in wireless ad hoc networks. *In Proceedings of the IEEE LCN 2004*, pages 419–420, 2004.

[LG01] S. J. LEE et M. GERLA : Split multipath routing with maximally disjoint paths in ad hoc networks. *In Proceedings of the IEEE ICC 2001*, volume 10, pages 3201–3205, 2001.

[Lin03] W. LIN : Some star extremal circulant graphs. *Discrete Math.*, 271:169–177, 2003.

[LR07] X. LIN et S. RASOOL : A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks. *In Proceedings of the INFOCOM 2007*, pages 1118–1126, 2007.

[LSG01] S.-J. LEE, W. SU et M. GERLA : Wireless ad hoc multicast routing with mobility prediction. *Mob. Netw. Appl.*, 6:351–360, 2001.

[MD02] M. K. MARINA et S. R. DAS : Ad hoc on-demand multipath distance vector routing. *Mob. Comput. Commun. Rev.*, 6:92–93, 2002.

[MMZ93] Y. MALKA, S. MORAN et S. ZAKS : A lower bound on the period length of a distributed scheduler. *Algorithmica*, 10:383–398, 1993.

[mpo08] MP-OLSR routing agent for NS-2. `http://jiaziyi.com/MP-OLSR.php`, 2008.

[MR92] Y. MALKA et S. RAJSBAUM : Analysis of distributed algorithms based on recurrence relations. *In Distributed Algorithms*, volume 579 de *Lecture Notes in Computer Science*, pages 242–253. Springer, Berlin, Germany, 1992.

[MWZ06] T. MOSCIBRODA, R. WATTENHOFER et A. ZOLLINGER : Topology control meets SINR: the scheduling complexity of arbitrary topologies. *In Proceedings of MobiHoc 2006*, pages 310–321, 2006.

[NNS+07] N. NANDIRAJU, D. NANDIRAJU, L. SANTHANAM, B. HE, J. WANG et D. P. AGRAWAL : Wireless mesh networks: current challenges and future directions of web-in-the-sky. *IEEE Wirel. Commun.*, 14(4):79–89, 2007.

[noa04] NO ad-hoc routing agent (NOAH). `http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/`, 2004.

[ns289] The network simulator NS-2. `http://www.isi.edu/nsnam/ns/`, 1989.

[PD91] P. M. PARDALOS et N. DESAI : An algorithm for finding a maximum weighted independent set in an arbitrary graph. *J. Comput. Math.*, 38(3–4):163–175, 1991.

[PHST00] M. R. PEARLMAN, Z. J. HAAS, P. SHOLANDER et S. S. TABRIZI : On the impact of alternate path routing for load balancing in mobile ad hoc networks. *In Proceedings of the MobiHoc 2000*, pages 3–10, 2000.

[PR99] C. E. PERKINS et E. M. ROYER : Ad-hoc on-demand distance vector routing. *In Proceedings of the WMCSA 1999*, pages 90–100, 1999.

[RC05] A. RANIWALA et T.-C. CHIUEH : Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. *In Proceedings of the IEEE INFOCOM 2005*, volume 3, pages 2223–2234, 2005.

[RWMX06] I. RHEE, A. WARRIER, J. MIN et L. XU : DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks. *In Proceedings of the MobiHoc 2006*, pages 190–201, New York, NY, 2006. ACM.

[SGP+07] M. SIEKKINEN, V. GOEBEL, T. PLAGEMANN, K.-A. SKEVIK, M. BANFIELD et I. BRUSIC : Beyond the future Internet: requirements of autonomic networking architectures to address long term future networking challenges. *In Proceedings of the FTDCS 2007*, pages 89–98, 2007.

[SHLK09] Y. SHI, Y. T. HOU, J. LIU et S. KOMPELLA : How to correctly use the protocol interference model for multi-hop wireless networks. *In Proceedings of the MobiHoc 2009*, pages 239–248, 2009.

[SJA+10] V. SRIKANTH, A. C. JEEVAN, B. A., T. S. KIRAN et S. S. BABU : A review of routing protocols in wireless mesh networks. *Int. J. Comput. Appl.*, 1(11):47–51, 2010.

[SMR+09] P. SANTI, R. MAHESHWARI, G. RESTA, S. DAS et D. M. BLOUGH : Wireless link scheduling under a graded SINR interference model. *In Proceedings of ACM FOWANC 2009*, pages 3–12, 2009.

[SR06] I. SHERIFF et E. Belding ROYER : Multipath selection in multi-radio mesh networks. *In Proceedings of the BroadNets 2006*, pages 1–11, 2006.

[Sta76] S. STAHL : n-tuple colorings and associated graphs. *J. Comb. Theory B*, 20:185–203, 1976.

[TH01] A. TSIRIGOS et Z. J. HAAS : Multipath routing in the presence of frequent topological changes. *IEEE Commun. Mag.*, 39(11):132–138, 2001.

[TM06] J. TSAI et T. MOORS : A review of multipath routing protocols: from wireless ad hoc to mesh networks. *In Proceedings of the ACoRN*, pages 17–18, 2006.

[TTAE09] M. TARIQUE, K. E. TEPE, S. ADIBI et S. ERFANI : Survey of multipath routing protocols for mobile ad hoc networks. *J. Netw. Comput. Appl.*, 32(6):1125–1143, 2009.

[VRBF12] F. R. J. VIEIRA, J. F. REZENDE, V. C. BARBOSA et S. FDIDA : Scheduling links for heavy traffic on interfering routes in wireless mesh networks. *Comput. Netw.*, 2012. To appear.

[WB06] S. WAHARTE et R. BOUTABA : Totally disjoint multipath routing in multihop wireless networks. *In Proceedings of the IEEE ICC 2006*, volume 12, pages 5576–5581, 2006.

[WB08] S. WAHARTE et R. BOUTABA : On the probability of finding non-interfering paths in wireless multihop networks. *In Proceedings of the IFIP TC6 2008*, pages 914–921, Berlin, Heidelberg, 2008. Springer.

[WDJ+08] J. WANG, P. DU, W. JIA, L. HUANG et H. LI : Joint bandwidth allocation, element assignment and scheduling for wireless mesh networks with MIMO links. *Comput. Commun.*, 31:1372–1384, 2008.

[WGLA09] X. WANG et J. J. GARCIA-LUNA-ACEVES : Embracing interference in ad hoc networks using joint routing and scheduling with multiple packet reception. *Ad Hoc Netw.*, 7:460–471, 2009.

[WWL+06] W. WANG, Y. WANG, X.-Y. LI, W.-Z. SONG et O. FRIEDER : Efficient interference-aware TDMA link scheduling for static wireless networks. *In Proceedings of the MobiCom 2006*, pages 262–273, New York, NY, 2006. ACM.

[XT09] X. XU et S. TANG : A constant approximation algorithm for link scheduling in arbitrary networks under physical interference model. *In Proceedings of ACM FOWANC 2009*, pages 13–20, 2009.

[YADP11] J. YI, A. ADNANE, S. DAVID et B. PARREIN : Multipath optimized link state routing for mobile ad hoc networks. *Ad Hoc Netw.*, 9:28–47, 2011.

[YCHP08] J. YI, E. CIZERON, S. HAMMA et B. PARREIN : Simulation and performance analysis of MP-OLSR for mobile ad hoc networks. *In Proceedings of the IEEE WCNC 2008*, pages 2235–2240, 2008.

[YCJ05] Y. YUAN, H. CHEN et M. JIA : An optimized ad-hoc on-demand multipath distance vector (AOMDV) routing protocol. *In Proceedings of the APCC 2005*, pages 569–573, 2005.

[YZ05] H.-G. YEH et X. ZHU : Resource-sharing system scheduling and circular chromatic number. *Theor. Comput. Sci.*, 332:447–460, 2005.

[ZLX05] X. ZHOU, Y. LU et B. XI : A novel routing protocol for ad hoc sensor networks using multiple disjoint paths. *In Proceedings of the BroadNets 2005*, pages 944–948, 2005.