

REUSE: A Combined Routing and Link Scheduling Mechanism for Wireless Mesh Networks

Carlos Henrique Pereira Augusto and Celso Barbosa Carvalho and Marcel William Rocha da Silva
and José Ferreira de Rezende

*Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ)
P.O. Box 68.504 – Rio de Janeiro, RJ 21.941-972
Emails: {chenrique,celso,marcel,rezende}@gta.ufrj.br*

Abstract

Increasing the capacity of wireless mesh networks has motivated numerous studies. In this context, the cross-layer optimization techniques involving joint use of routing and link scheduling are able to provide better capacity improvements. Most works in the literature propose linear programming models to combine both mechanisms. However, this approach has high computational complexity and cannot be extended to large-scale networks. Alternatively, algorithmic solutions are less complex and can obtain capacity values close to the optimal. Thus, we propose the REUSE algorithm, which combines routing and link scheduling and aims to increase throughput capacity in wireless mesh networks. Through simulations, the performance of the proposal is compared to a developed linear programming model, which provides optimal results, and to other proposed mechanisms in the literature that also deal with the problem algorithmically. We observed higher values of capacity in favor of our proposal when compared to the benchmark algorithms.

Keywords: wireless networks, mesh, link scheduling, routing

1. Introduction

The study of the capacity of wireless mesh networks (WMNs) has received a great deal of attention in the last years [1, 2, 3, 4, 5, 6]. These works show that the capacity of WMNs depends on several factors such as nodes density and placement, and traffic demand. Gupta and Kumar [1] show that the throughput per node is given by $\Theta(\frac{1}{\sqrt{n \log n}})$, assuming arbitrary communication pattern between pairs of n identical and randomly placed nodes. Following this fundamental work, different models for a variety of scenarios were proposed to estimate the capacity of WMNs [7, 8].

A methodology commonly used on these works to calculate bounds for the network capacity assumes that link scheduling occurs in slots of time, where links that do not present mutual interfer-

ence may be activated at the same time slot. Hence, optimal link scheduling algorithms aim to determine the links that will be activated in each time slot in such a way that all links of the network are scheduled at least once in a certain minimum number of slots. When used as the TDMA schedule, this minimum sequence of slots corresponds to the maximum capacity of network under asymptotic traffic conditions. Thus, the goal is to minimize TDMA schedule length in order to maximize capacity. On the other hand, this same TDMA schedule length designates the maximum throughput that can be achieved under a certain traffic demand matrix. In this case, only the links with traffic demand should be scheduled. The works in [2, 9, 3, 10, 4, 11, 12, 5, 13, 14, 6] follow this approach.

Some of these works [2, 3, 10, 12, 5, 13, 14] integrate routing and link scheduling in order to obtain improved performance bounds. These works define how links should be scheduled and which routes should be chosen to meet a certain traffic demand in a minimum time. They show that the combined use of routing and link scheduling mechanisms provides a better performance than the independent use of them.

The idea of coupling these mechanisms comes from the fact that one can select routes that suffer and cause less interference, and hence, favor the simultaneity in their links activation. The integrated mechanisms look for a balance between choosing longer routes, which are less prone to interference, but have more links to be scheduled, and selecting shorter routes with fewer links that suffer a larger interference. Conversely, routing has a great impact in the link scheduling since multi-hop paths followed by traffic flows determine the links of the network that need to be activated and their respective demand. Normally, scheduling algorithms deal with this increased demand by assigning weights to links that are traversed by traffic flows.

However, the joint optimization, named cross-layer, of routing and link scheduling mechanisms presented in most of these works uses the linear programming optimization approach to obtain performance bounds. Therefore, there are few works that follow an algorithmic approach to propose combined link scheduling and routing algorithms [13, 14]. The algorithm proposed in this paper follows this approach.

At first, this paper evaluates the impact of the routes used by multi-hop flows in WMNs in the performance of link scheduling algorithms. Following, a joint mechanism of routing and scheduling, called REUSE, that explores the spatial reuse by favoring the simultaneity in links activation is proposed. This mechanism defines both a routing metric and a scheduling algorithm in which the traffic demand and the allowed degree of interference among links have been taken into account.

The main contributions of this paper are twofold, the proposal of the REUSE algorithm, which combines routing and link scheduling, and its eval-

uation and comparison to other proposals and to the joint optimization through linear programming of both mechanisms. The obtained results show the efficiency of the mechanism in comparison to other three algorithms [4, 6, 15], two of which schedule links traversed by shortest path flows [4, 6] and a third one that performs integrated routing and link scheduling [15].

This paper is organized as follows. The next section presents related works. Section 3 describes the proposed mechanism. In Section 4, the simulation environment, the models used in the performance evaluation of the algorithms, and the problem formulation through linear programming are detailed. Section 5 shows the numerical results obtained through simulation and linear programming optimization as well as a discussion about them. Finally, Section 7 draws the conclusions.

2. Related Work

The problem of link scheduling in wireless networks has been tackled for a long time [16, 9, 10, 17, 4, 15, 12, 11, 5, 6, 18, 14]. These works address the problem in an isolated fashion [9, 10, 4] or jointly with routing and topology/power control [16, 17, 15, 5, 12]. According to the used methodology, existing works can be classified as follows: i) optimization by linear/integer programming [17]; ii) graphs coloring [19, 12] iii) approximation algorithms [4, 6]; iv) other (neural networks, genetic algorithms, heuristics, etc.) [16, 15, 11]. Concerning the interference model, these studies use the protocol (or UDG) [9, 12] or physical (or SINR) models [4]. In most cases, algorithms/heuristics are centralized and require global topology knowledge and/or information about links' interference levels. However, some studies propose distributed algorithms [19, 12] based on graph coloring.

A work similar to ours is presented in [14]. The authors propose the use of multipath routing, where each one of the paths followed by a certain flow is sequentially chosen by using a metric that takes into account the scheduling of the links traversed by the previous path. The results of their heuristic are compared to the optimal value obtained by solving an optimization problem that

takes into account both routing and scheduling. However, the authors do not compare their results to others in the literature.

By modeling the wireless network as a graph, the scheduling problem can be translated into a graph coloring problem [19, 3, 12]. This coloring can be made both on edges (edge-coloring) and on vertices (vertex-coloring). In the first case, the coloring is valid if all incident edges of a vertex have distinct colors. This coloring can be easily mapped to a TDMA scheduling, where each edge represents a communication link and each color corresponds to a TDMA slot. Edges that receive the same color correspond to links that can be activated in the same slot. The disadvantage of edge coloring is that despite of ensuring that there are no primary conflicts in scheduling¹, it does not guarantee that there are no collisions caused by mutual interference due to the activation of two or more links that share the same color.

To overcome this problem, the work in [3] introduces the concept of conflict graph, whose vertices correspond to communication links and an edge between two vertices exists if the two links cannot be simultaneously activated. Therefore, it is possible to create edges in the conflict graph that represent the mutual interference between links for a given interference model. Thus, when coloring the vertices of the conflict graph, which establishes that any two adjacent vertices must receive distinct colors, conflicts caused by scheduling interference are avoided. In summary, the problem of interference-free scheduling of links consists on determining the minimum number of colors necessary to color the vertices of the conflict graph. The work in [12] proposes several algorithms, both centralized and distributed, to the coloring problem.

The next three subsections detail the three algorithms that have been used as comparison basis to the proposed one. The first one, the LDF (Lesser Distance First) algorithm [6], was chosen due to its ability to provide the upper-

bound capacity of arbitrary networks. Also, the authors prove that the algorithm approximates to the optimum solution within a factor of $O(\log(n))$, where n is the number of links. The second algorithm, GreedyPhy, deals with heterogeneous traffic demands, presents good performance and was used as basis for our proposal. The third algorithm, IRMA, is the most similar to ours since it defines integrated routing and MAC scheduling algorithms.

2.1. LDF Algorithm

The work in [6] presents a centralized scheduling algorithm that determines the throughput capacity of arbitrary networks. It employs the SINR interference model together with a geometric analysis. The objectives of the algorithm are to maximize the amount of links scheduled concurrently in a slot and to schedule all links in the smallest amount of slots.

The algorithm chooses from a set $L = \{l_1, \dots, l_n\}$ of links to be scheduled the smaller link l_v of length d_{vv} (Euclidean distance between source s_v and receiver r_v). After l_v is added to slot s_1 of set $S = \{s_1, \dots, s_T\}$, the SINR restriction guarantee is performed in two steps. First, every link l_u that is within a radius $c \cdot d_{vv}$ of the receiver r_v is removed from the set L , where c is a constant determined by geometric calculations and depends on the SINR required by link l_v and on the average path loss exponent. In the second step, all remaining links that have *affectedness* greater than or equal to $2/3$ are also removed from the set L . The affectedness of a link l_w measures how this link is affected by transmission of others links already present in one slot and is given by $A_S(l_w) = \beta(\sum_{l_v \in S} I_{vw} + N)/P_{ww}$. In this equation, β is the SINR required by link l_w , I_{vw} is the total interference within one slot of set S , caused by all transmitters s_v in the receiver r_w , N is the background noise power and P_{ww} is the received power in receiver r_w from transmitter s_w . The algorithm is executed repeatedly, taking as input $L = L \setminus S_t$ ($t = 1, \dots, T$), until all links of set L have been scheduled.

¹Occurrence in the same slot of simultaneous transmission and reception at the same node or reception/transmission to/from two or more nodes

2.2. GreedyPhy Algorithm

The work in [4] proposes a centralized algorithm, called GreedyPhysical, which employs the SINR interference model and a conflict graph that models each edge as a vertex. Each edge has a weight representing the signal strength received by other vertices. The idea is used to calculate the interference that each node causes over all others while it is transmitting.

The first step of the algorithm to calculate the schedule S_{Phy} is to determine the interference number of each link, and sort those links in decreasing order of interference number. The interference number of a link $e = (u, v)$ is the number of links e_i that do not own vertex u or v , and cannot be simultaneously scheduled with e due to interference. After ordering them, links are greedily scheduled in the first possible slots. To determine whether it is possible to schedule e_i in slot t_k the algorithm checks if the power added by e_i in each link already scheduled in that slot does not exceed the required SINR threshold. Also, the sum of the power of all transmissions previously scheduled should not exceed the same threshold at e_i . If existing slots are not sufficient to allocate all transmissions of e_i , new slots are added to the end of the S_{Phy} schedule. The remaining transmissions of e_i are scheduled alone in these slots.

2.3. IRMA Algorithm

The work in [15] proposes centralized algorithms to jointly select paths to be followed by the flows and to set up end-to-end TDMA schedules. In [13], the same authors provide distributed algorithms for the same problem. For path selection, an SPF algorithm is used, where the cost of each link is updated with the number of already allocated slots plus blocked slots. Then, min-hop and bandwidth-aware algorithms are used to calculate the path for the flow. Once the path is selected, the TDMA schedule for each link along the path is updated to account for the allocation of slots required by this new flow. Also, those links which cannot be concurrently scheduled with these links have their respective slots blocked. To determine which links cannot be simultaneously scheduled

within the allocated slots, IRMA uses an n -hop neighborhood interference model.

The n -hop neighborhood interference model is derived from the protocol-based interference model, where distance-based constraints are substituted by hop-based ones. The model states that any node k within the n -hop neighborhood of the intended receiver must be not transmitting. Thus, n is used as interference index instead of the interference range. Let α and $SINR_{thresh}$ be the path loss exponent and the minimum required SINR in order to successfully decode a packet, respectively, then n should be the first integer that satisfies $n \geq \sqrt[\alpha]{SINR_{thresh}}$ [15]. Due to the use of this interference model, IRMA does not guarantee an interference-free schedule, i.e. collisions may occur in some slots. For this algorithm be compared to the others, which are interference-free, we choose n as the first integer that satisfies the equation above but that also provides an interference-free schedule, i.e. we increase the value of n until this is achieved.

3. Proposed Mechanism

The proposed mechanism, named REUSE, is based on the use of a routing metric that favors spatial reuse, and of a scheduling algorithm that increases the number of simultaneous activated links. Next, the REUSE mechanism and the assumptions used in its distributed design are described.

3.1. REUSE Mechanism

The REUSE is composed of two mechanisms for routing and link scheduling, which were specially designed to work together aiming to obtain a cross-layer optimization. The routing mechanism consists of a new metric, which assumes the use of an SPF algorithm, while the proposed scheduling algorithm is an extended version of the GreedyPhy [4], which is described in Section 2.2, with the purpose of coupling it with the routing. Although the separate operation of both mechanisms is straightforward in its objectives, their joint operation is the main contribution of the REUSE mechanism.

The extensions to the GreedyPhy algorithm concern the calculation of the interference number of each link. In our proposal, this value is calculated by multiplying the traffic demand of the link by the number of links with which it cannot be simultaneously scheduled since they cause mutual interference. Moreover, only links that have some traffic demand are counted as interfering links, instead of all links as in the original algorithm. However, as said before, the traffic demand imposed on links depends on routes chosen by flows. By doing this extension, a coupling between routing and link scheduling is created, which means that the choice of routes has an impact on how links will be scheduled, i.e. a joint mechanism is designed. Considering a network where all traffic flows from multiple sources to a unique sink, the intuition behind this extension is related to the fact that links to/from the sink (or gateway) are usually the most demanding, and then should be scheduled with priority.

The reasoning behind the routing metric is to balance traffic across the network, increasing the weight of links around routes used by flows according to the level of interference that is allowed. This load balancing favors simultaneous link schedules, increasing spatial reuse and, hence, the network capacity.

In the beginning, all links have their weights set to 1, which are modified as new flows enter or leave the network. We assume the following sets of nodes and links:

- N_i - set of nodes traversed by the flow i ;
- R_i - set of links traversed by the flow i ;
- V_i - set of one-hop neighboring nodes of nodes that belong to N_i ;
- $E(X)$ - set of all incident links to nodes of the set X .

The weight of link j assumes a new value w'_j given by:

$$w'_j = \begin{cases} w_j \pm \alpha & \forall j \in R_i; \\ w_j \pm \beta & \forall j \in E(N_i) \mid j \notin R_i; \\ w_j \pm \gamma & \forall j \in E(V_i) \mid j \notin E(N_i). \end{cases} \quad (1)$$

Where w_j is the previous weight of the link j , and α , β , and γ should be set according to the following reasoning. Firstly, these parameters should assume values of the same order of magnitude of the initial weight since SPF depends on relative weight values. In the case where these parameters are much smaller than the initial weight, they will not influence route calculations. In contrast, if set too large they can completely exclude links already used, and hence create stretched routes, which require lengthy scheduling. Moreover, it is intuitive to have $\alpha > \beta > \gamma$ since their objective is to discourage the use of: i) firstly, the links traversed by the flow; ii) followed by the links related to the nodes traversed by the flow; iii) and finally, the links related to the one-hop neighbor nodes of the nodes traversed by the flow.

Another important issue regarding α , β and γ calculation is the susceptibility of close links to suffer interference. This can be measured by the $SINR_{thresh}$, i.e. the minimum signal-to-noise ratio (in dB) required to decode a packet at the transmission rate used by link j . For this purpose we define the equation 2.

$$r = 1 + \frac{(SINR_{thresh} - minSINR)}{(maxSINR - minSINR)} \quad (2)$$

Equation 2 provides values between $[1, 2]$ for $SINR_{thresh}$ in the range of 5 ($minSINR$) to 30 ($maxSINR$) dB, which roughly corresponds to the SINR values required to successfully decode frames transmitted at 2 Mbps and 54 Mbps rates in the 802.11g, respectively. The higher the value of $SINR_{thresh}$, the higher are the weights assigned. Thus, one can describe α , β and γ as $1 * r$, $1/2 * r$ and $1/4 * r$, respectively.

The intuition behind the definition of parameter r (Equation 2) used in the α , β and γ calculation is that links are more susceptible to interference in scenarios where higher transmission

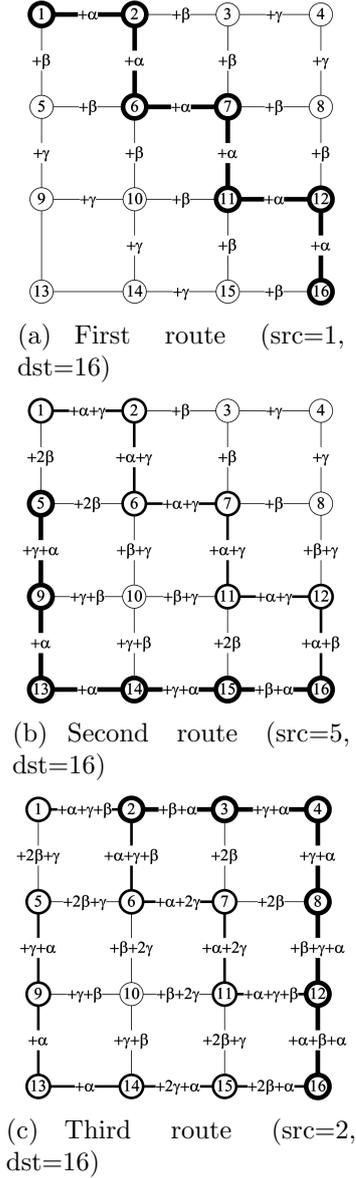


Figure 1: Three consecutive routes created by the proposed routing metric.

rates are used. Therefore, the higher values of $SINR_{thresh}$ needed to decode packets transmitted at higher rates result in an increased r , and hence in higher values for α , β and γ . By doing so, a better spatial spreading of future calculated routes is expected, which favors simultaneity of links scheduling.

Once new weights are determined, the route to be followed by a new admitted flow must be the one with lowest cost calculated by the SPF algorithm. The sign \pm in the Equation 1 indicates

that the value of w_j should be increased (+ sign) at the arrival of a new flow and decreased ($-$ sign) when a flow leaves the network.

Figure 1 exemplifies the sequential establishment of three routes from source nodes 1, 5 and 2 to destination node 16 in a grid topology under the proposed routing metric. The subfigures show the paths taken by each route (indicated by the boldest lines) and the updated link weights after each route creation. The start of each flow increases the weights of links closer to the selected path by the parameters α , β and γ . The SPF algorithm tends to select new paths that pass through lowest cost links, which are farther from the previously created ones. This way, the proposed metric promotes spatial spreading of routes, which favors posterior link scheduling.

3.2. Distributed Design Assumptions

The REUSE mechanism assumes a centralized approach with global information knowledge. However, we argue that a distributed algorithm could be developed. In the following, some considerations in defining this algorithm are given. Let a multi-hop wireless network be modeled as a directed graph $G(V, E)$ composed of a vertex set $V = \{v_1, v_2, \dots, v_n\}$, $n = 1, \dots, |V|$ that represents wireless nodes and a edge set $E = \{e_{1,2}, \dots, e_{i,j}\}$, $i, j = 1, \dots, |E|$ that denotes communication links.

Assuming a global knowledge of all nodes positioning, propagation model and transmission power, each node $v_i \in V$ can determine a power matrix P_r , where each element p_{jk} contains the reception power at node v_k from node v_j for $j, k = 1, 2, \dots, n$. By using this matrix along with the topology graph, each node can calculate interference numbers. However, with the purpose of avoiding making strong assumptions about propagation models, a new approach is described next.

Considering a distributed algorithm that assigns globally unique and ordered integer numbers to nodes (nodeID) [20], a sequence of mini-slots can be used with the aim of each node v_i to obtain its power vector, i.e. the i th row of the power matrix P_r . During this sequence, each node transmits a tone in the mini-slot matching its nodeID. Assuming a conflict-free schedule

and global knowledge of nodeIDs, each node can calculate its own power vector. After flooding their power vector, nodes can calculate interference numbers for all links.

However, interference numbers calculated by the method above do not take into account the traffic demand of each link. In order to estimate this, each node keeps track of the number of active flows and their respective demand for each outgoing link. By piggybacking this information on link state advertisements, each node can locally calculate interference numbers, rank links by this value, and calculate respective schedules. Link weights are also updated by using this flooded information according to equation 1.

In the next section, the interference and propagation models used in the system evaluation are detailed. Also, some details of the simulator used in this assessment are provided. Next, the LP formulation of the problem is presented.

4. Models, Simulation Environment and LP Optimization

To determine the links that do not interfere and therefore can be simultaneously activated, an interference model should be used. Two models are widely used: physical (or SINR) and protocol (or UDG). The first determines whether the signal-to-interference+noise ratio (SINR) is greater than a certain threshold so that the transmitted signal can be decoded with a certain bit error rate (BER) [18]. The protocol model establishes that a successful transmission occurs when a node is within the transmission range of the intended transmitter and away from interference range from others. In this second model, the greatest difficulty is to determine the proper interference range. The work in [18] shows that, with a good choice of this value, both models lead to the same results, and the protocol model is much simpler computationally.

In this study, the simulations consider the physical interference model (or SINR) associated with the log-distance path loss model to determine the feasibility in the concurrent activation of links. The physical model establishes that the probabil-

ity of a successful packet reception is equal to 1 if the SINR is above a certain threshold ($SINR_{thresh}$). Equation 3 shows the necessary condition for successful packet reception, where P_t^i is the signal power of the intended transmitter i and g_{ij} is the path loss from node i to node j on link $e_{i,j}$. The denominator corresponds to the noise N added to the signal power that arrives at node j issued by all possible interfering node k with $k \in N, k \neq i, k = 1, \dots, |K|$, where K is the set of all transmitter k that interferes in the receiver j of the link $e_{i,j}$. In this model, the simultaneous scheduling of a set of links is considered valid (or feasible) if all transmissions do not suffer interference that can cause loss of transmitted packets.

$$\frac{P_t^i * g_{ij}}{\sum_{k \neq i} P_t^k * g_{kj} + N} \geq SINR_{thresh} \quad (3)$$

The log-distance path loss model provides the average received signal power (P_r), given by $\frac{P_t * \gamma}{d^\alpha}$, where P_t is the transmission power. The path loss factor α reflects the influence of the environment, d is the distance between transmitter and receiver and γ factor takes into account the influence of the frequency, antenna gain and reference distance.

In addition to the restrictions imposed by interference, called secondary constraints, the link scheduling must meet the primary constraints, also referred as wireless transceiver constraints in [5]. Typically, these transceivers are half-duplex, which prevents a node to transmit and receive simultaneously. Moreover, a given node can receive packets from only one transmitter at each instant, and can only transmit to one receiver at a time.

To take into account the routes used by the scheduled flows, the traffic demand of each link shall be calculated as the sum of each flow demand passing through it. Once we know the traffic demand of all links, the links that have zero demand can be eliminated and those with non-zero demand must be proportionally scheduled to the amount of the demand.

The evaluation of link scheduling algorithms consists in the generation of topologies and the use of algorithms to determine the TDMA cycle needed to meet the traffic demand. The proposed

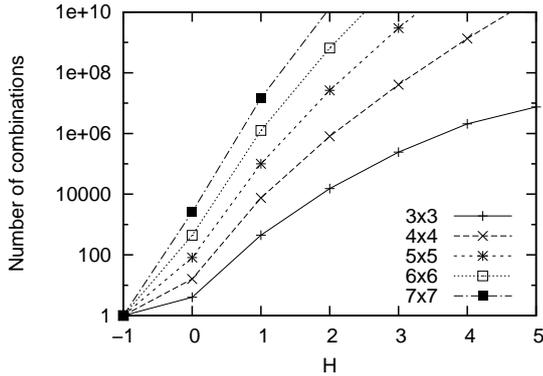


Figure 2: Number of route combinations as a function of H .

mechanism is compared to three algorithms detailed in the subsections 2.1, 2.2 and 2.3. As the first two algorithms do not work together with a specific route selection algorithm, their performance is obtained for different sets of shortest path routes.

Thus, it was implemented in the simulator a variant of the SPF algorithm that takes as input a H parameter which has the following meaning. When $H = -1$, the algorithm returns the first shortest path from source to destination found. When $H = 0$, it returns all shortest paths, and when $H > 0$, all paths with up to H hops higher than the shortest path are found. The implemented algorithms are compared to the proposed mechanism for different values of H . When $H = -1$, only one route for each source will be used by the algorithm, however, when $H \geq 0$, the combination of all routes for each source will be used in the evaluation.

Considering a wireless network that has only one gateway node and where all links have transmission capacity of W bits/s, the scheduling size T is related to the network capacity through $C = W/T$. Thus, the average size of the scheduling in number of slots for that combination of routes will be used as the performance metric to be evaluated. The smaller is the scheduling size the higher is the capacity.

The brute force method to the minimal scheduling problem would be to use all routes, from the shortest up to $N - 1$ hops. However, the

number of route combinations presents a large increase with H , as shown in Figure 2. This figure shows the total number of computed routes when H varies from 1 to 5 in scenarios of square grid networks. The destination node is the one placed at top-right corner of the grid topology and the source nodes are the ones placed at the three other corners. When $H = 1$ there are more than 100 route combinations in a small 3x3 network, and more than 10 million route combinations in a 7x7 network. This result proves the infeasibility of a brute force computation.

4.1. Problem Formulation through Linear Programming

The LP formulation presented here is similar to the one described in [3]. It considers multiple flows, where each flow follows a single path, with a physical-based interference model. However, additional constraints were added with the purpose of obtaining the scheduling size as the performance bound. Also, in our LP formulation, all flows receive equal throughputs provide that they have the same traffic demand.

The complete LP formulation, similar to the one presented in [3], is summarized below for the sake of clarity. Given a wireless network with N nodes, the connectivity graph $C(N_C, L_C)$ is composed by the vertex set N_C (wireless nodes) and the edge set L_C (wireless links). There is a directed link l_{ij} from node n_i to n_j if $SNR_{ij} \geq SNR_{thresh}$ and $i \neq j$, where SNR_{ij} means the signal-to-noise ratio at the node n_j for transmissions received from node n_i , without considering any interference.

Let us consider communication between multiple sources, $n_s \in S$, and a single destination, n_d . Finding the maximum achievable throughput between the sources and the destination using single path can be formulated as a linear program corresponding to a multi-commodity flow problem. Also, let us consider that there are k commodities in $M = \{M_1, M_2, \dots, M_k\}$, defined by $M_i = (i, s)$, where n_s is the source of commodity i . Thus, f_{ijk} denotes the amount of flow on link l_{ij} from commodity k , Cap_{ij} denote the capacity of link l_{ij} ,

and L_C is a set of all links in the connectivity graph.

The maximization states that we wish to maximize the sum of flow out of the sources. The constraint (5) establishes the flow conservation, i.e., at every intermediate node, except the sources and the destination, the amount of incoming flow is equal to the amount of outgoing flow. The constraints (6) and (7) state that the incoming flow to source nodes and the outgoing flow from the destination node is 0. The constraint (8) restricts the amount of flow on each link to be non-negative. The constraint (9) indicates the amount of flow on a link cannot exceed the capacity of the link. The constraints (10) and (11) states that all flows follow a single-path. The constraint (12) establishes that all sources generate the same amount of flow. The constraints from (13) to (15) are related to the wireless interference. They state that the amount of flow on a link is constrained by the sum of the activity periods of the schedulable sets that this link is a member of. A schedulable set is composed by links that can be simultaneously scheduled according to the physical interference model. And, the maximal set is the one that by adding any other link turns the whole set non-schedulable. This LP formulation consider that only one maximal schedulable set can be active at a time (see [3] for more details), which means that each schedulable set receives a fraction of time given by λ_x .

Max:

$$\sum_{\substack{l_{si} \in L_C \\ n_s \in S}} f_{sik} \quad (4)$$

Subject to:

$$\sum_{\substack{l_{ij} \in L_C \\ (k,s) \in M}} f_{ijk} = \sum_{\substack{l_{ji} \in L_C \\ (k,s) \in M}} f_{jik} \quad , \forall n_i \in N_C \setminus (S \cup \{n_d\}) \quad (5)$$

$$\sum_{\substack{l_{is} \in L_C \\ n_s \in S \\ (k,s) \in M}} f_{isk} = 0 \quad (6)$$

$$\sum_{l_{di} \in L_C} f_{dik} = 0 \quad (7)$$

$$f_{ijk} \geq 0 \quad (8)$$

$$\sum_{(k,s) \in M} f_{ijk} \leq Cap_{ij} \quad (9)$$

$$f_{ijk} \leq Cap_{ij} \cdot z_{ijk} \quad , \quad \forall l_{ij} \in L_C \text{ and } z_{ijk} \in \{0,1\} \quad (10)$$

$$\sum z_{ijk} \leq 1 \quad , \quad \forall n_i \quad (11)$$

$$\sum_{\substack{(k,s) \in M \\ n_s \in S}} f_{sjk} = \sum_{\substack{(m,r) \in M \\ n_r \in S}} f_{rjm} \quad (12)$$

$$\sum_{(k,s) \in M} f_{ijk} \leq \sum_{l_{ij} \in H_x} \lambda_x Cap_{ij} \quad (13)$$

$$\sum_{x=1}^{K'} \lambda_x \leq 1 \quad (14)$$

$$\lambda_x \geq 0 \quad , \quad \forall x \in \{1..K'\} \quad (15)$$

Since finding all maximal schedulable sets is a NP-hard problem [3], only a limited number of sets are found in practice. In this case, the LP formulation above provides only a lower bound of the problem. The following algorithm has been used to find K' schedulable sets H_x that are used in the constraints (13) to (15) of the formulation problem:

1. initialize $x = 1$ and *Linkslis*t = sort of all links $\in L_C$ in decreasing order of their interference number;
2. put the first link l of *Linkslis*t in the schedulable set H_x and exclude l from *Linkslis*t;
3. for all $l \in L_C$, insert it in H_x if it is still schedulable;
4. if H_x is equal to any previous H schedulable set: discard it; else: increment x ;
5. if *Linkslis*t is not empty, put next link of *Linkslis*t in new H_x , exclude it from *Linkslis*t and return to step 3;

6. if schedulable sets converge: exit; else: create a new randomized *Linklist* of all links $\in L_C$, and return to step 2.

The algorithm above can find all maximal schedulable sets if it is fed with all links list permutation ($m!$ inputs, where m is the number of network links) instead of randomized lists, as cited in the step 6. We use a convergence criteria to limit the computational cost. If steps from 2 to 5 do not increment the number of schedulable sets for 10 times, we stop the algorithm. Other stop criteria is to use 50 thousands link randomization, or x greater then 200 thousands since a large number of schedulable sets increases computational cost of linear programming solution.

5. Numeric Results

We have developed our own simulator using C language, which implements the SPF algorithm for routing and all link scheduling mechanisms evaluated. Also, it implements the log-distance propagation and physical interference models.

We have performed simulations using two distinct topologies: **grid** and **pseudo-random**. In both scenarios, the data traffic flows in multiple hops from the sources to the gateway, which is always placed in one of the corners of the square area covered by the network. Traffic sources are chosen amongst the nodes placed at the borders diametrically opposed to the gateway, which are the farthest nodes from the gateway. Traffic demands are equal among all source nodes.

At the grid topology, nodes are horizontally and vertically distant from each other by 70% of the maximum communication range. This placement allows direct communication with neighbors placed in the diagonal. The maximum communication range is calculated taking into account the transmission power (P_t), the path-loss exponent (α) and the minimum signal-to-noise ratio ($SINR_{thresh}$).

The pseudo-random topologies are created by using the following rules. The position of a new node is randomly chosen following a uniform distribution. This new node is inserted in the topology only when it is within the communication

range of at least one previously placed node, and the degree of the network graph does not exceed a maximum threshold. Moreover, we also guarantee a minimum distance between all pairs of nodes. The parametrization chosen for this evaluation were the following: P_t is set to 17 dBm; α ranges between 5 and 30 dB; the maximum degree ranges between 6 and 9; and the minimum distance between nodes is equal to 25% of the maximum communication range. We have also performed simulations with varying number of source nodes.

As previously mentioned, the performance metric used to evaluate the mechanisms is the size of TDMA cycle in number of slots, hereafter referred to as the scheduling size. As stated in Section 4, this performance metric is inversely related to the network capacity. The four evaluated mechanisms and their respective acronyms are: GreedyPhy [4] (GPHY), the algorithms presented in [6] (LDF - Lesser Distance First) and in [15] (IRMA - Integrated Routing and MAC Scheduling), and our proposal REUSE. Moreover, we also plot the optimal results obtained by the solution of the Linear Program described in Section 4.1 (LP). We used the open-source GLPK package [21] to solve the LP formulation. However, some results of the LP were omitted because of the limitation of the algorithm used to find the maximal schedulable sets. The algorithm is surely unable to find all maximal schedulable sets whenever its execution is stopped because 50 thousand link randomization were reached or 200 thousand schedulable sets were found. At those cases, the lower bounds provided by the LP formulation were assumed to be far from the optimal result and were omitted in the graphs.

To evaluate the impact of routing at the performance of the link scheduling mechanisms, we have performed simulations in grid topologies using all possible combinations of routes. The results presented in Figure 3 show the average, minimum and maximum values of the scheduling size obtained by the GPHY and LDF mechanisms into a grid with 6x6 nodes (36 nodes), where five source nodes located at the edges diametrically opposed to the gateway are used. The vertical bars in

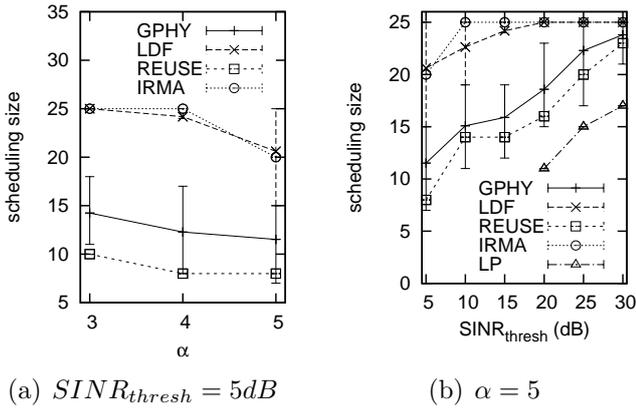


Figure 3: Average, min. and max. scheduling size (6x6 grid and $H = 0$).

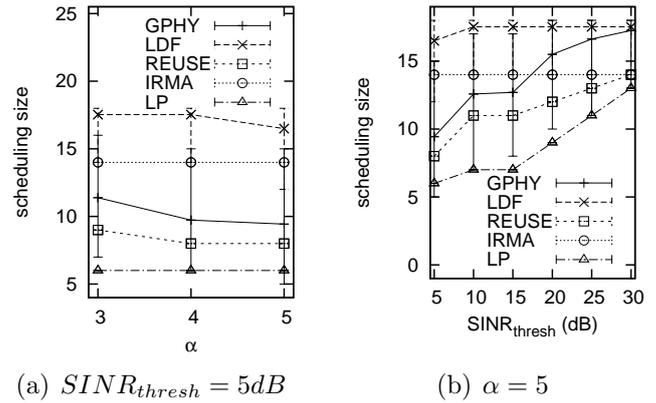


Figure 4: Average, min. and max. scheduling size (4x5 grid and $H = 1$).

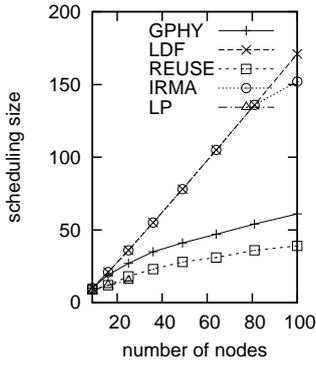
the graph are plotted between the minimum and maximum values while the curve indicates the average scheduling size for the whole combination of routes. These results are compared to the ones obtained by the REUSE and IRMA mechanisms, which choose specific routes according to their specialized link weight metrics. Moreover, the optimal results provided by the LP formulation are included for benchmarking purposes. For the GPHY and LDF simulations, all routes with number of hops equal to the shortest path route (SPF with $H = 0$) are obtained and combined leading to 176400 route combinations. The number of possible combinations grows with the number of source nodes and routes of the same size from these sources to the gateway. In a grid topology, there are an increasing number of routes of same cost as function of the distance between the source and the gateway. For example, by using $H = 1$ for this same topology and the same number of sources, the number of combination of routes is superior to 6×10^9 .

Figure 3 shows that LDF and GPHY exhibit a varying performance according to the route combination used since minimum and maximum values for the scheduling size present a wide divergence. This shows the impact of routes choice on the link scheduling performance. Moreover, one can notice that the REUSE mechanism presents smaller average scheduling sizes than the other solutions, giving results close to the minimum ob-

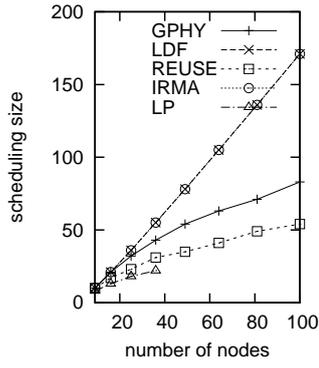
tained by the GPHY algorithm. This indicates that the proposed mechanism is able to find routes that allow better link scheduling. Otherwise, the IRMA mechanism provides poor results, with average scheduling sizes close to LDF. It occurs mainly because differently from REUSE the link scheduling algorithm of IRMA incrementally schedules the links of each discovered route, without rescheduling the links of the previously discovered ones. Therefore, IRMA has difficulties to improve global performance, even by using a specialized link weight metric.

Figure 4 shows the same results for a twenty node grid (4x5) with five source nodes by combining all routes up to one hop larger than the shortest path route ($H = 1$). In this case, there are more than three millions of combinations. The average scheduling size for GPHY and LDF algorithms were obtained for all route combinations, and are compared to the result of the proposed mechanism. Here again, the REUSE provides better results, lower than the average and close to the minimum values obtained with GPHY, and IRMA is unable to achieve the same performance of REUSE in most scenarios.

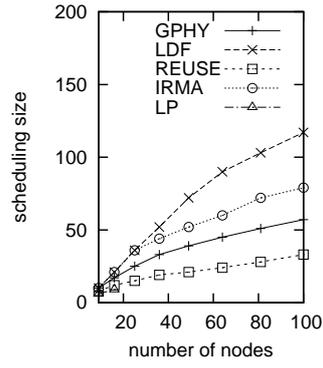
Also in Figures 3 and 4, one can see that the performance of the REUSE mechanism is still away from the optimal results obtained by the LP formulation in many scenarios. It is also interesting to notice that in the 4x5 grid, some minimum values of the scheduling size obtained by the GPHY



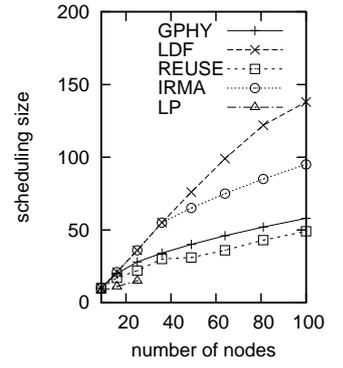
(a) $SINR_{thresh} = 5dB$



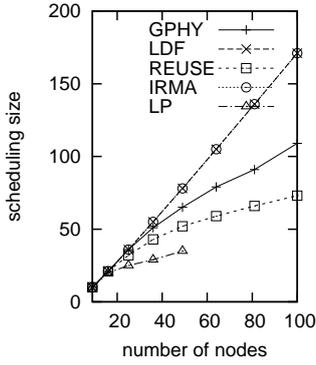
(b) $SINR_{thresh} = 10dB$



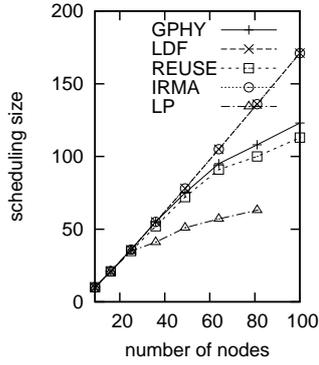
(a) $SINR_{thresh} = 5dB$



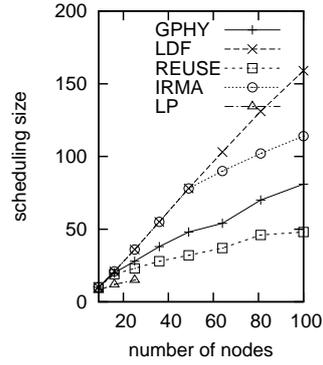
(b) $SINR_{thresh} = 10dB$



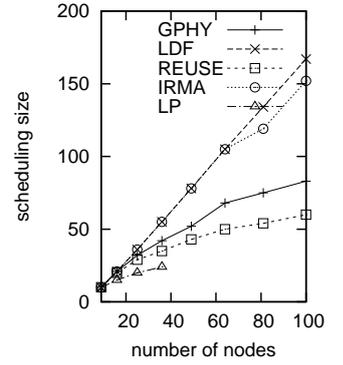
(c) $SINR_{thresh} = 15dB$



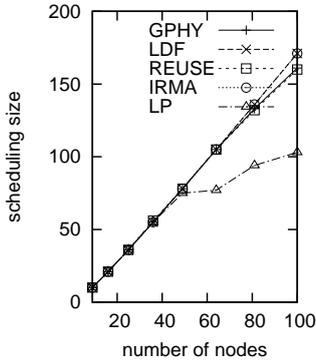
(d) $SINR_{thresh} = 20dB$



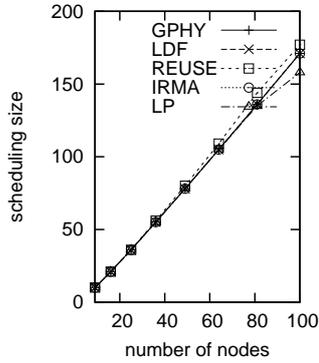
(c) $SINR_{thresh} = 15dB$



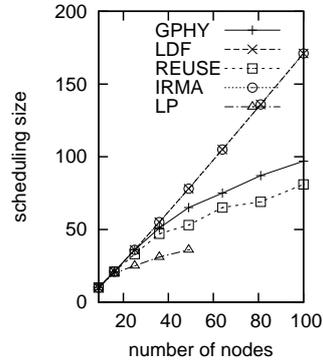
(d) $SINR_{thresh} = 20dB$



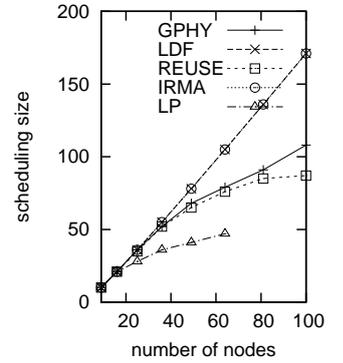
(e) $SINR_{thresh} = 25dB$



(f) $SINR_{thresh} = 30dB$



(e) $SINR_{thresh} = 25dB$



(f) $SINR_{thresh} = 30dB$

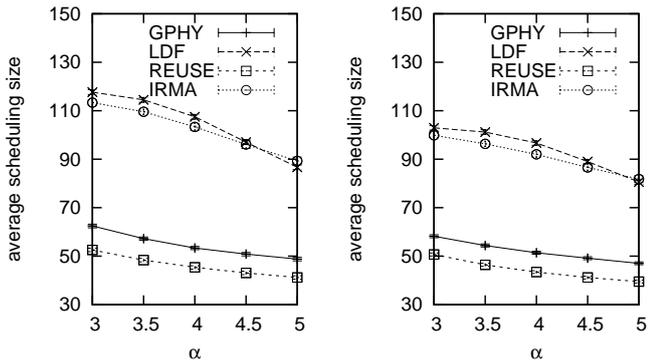
Figure 5: Scheduling size as a function of the number of nodes for $\alpha = 3$.

algorithm are equal to the optimum. It proves the importance of the careful selection of routes, which is able to provide optimal performance to the network.

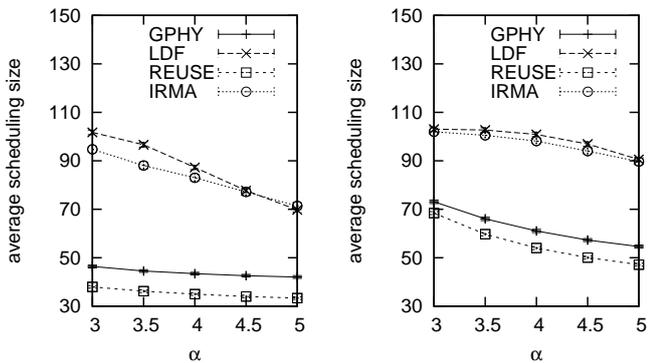
In what follows, we present the results obtained by increasing the number of nodes in grid topologies. We used $N \times N$ grids, where N nodes are placed in each line and column. In these experiments, all nodes located at the borders dia-

Figure 6: Scheduling size as a function of the number of nodes for $\alpha = 5$.

metrically opposed to the gateway are traffic sources. Therefore, the number of sources is given by $2 \times N - 1$, where N is the number of rows/columns of the grid. The graphs of Figures 5 and 6 show the scheduling size as a function of the grid size, in number of nodes (N^2), for different values of $SINR_{thresh}$ and α . At these simulations, the route used by the GPHY and LDF algorithms is the shortest path route ($H = -1$), while the IRMA and



(a) $SINR_{thresh} = 10dB$, max. degree 6, (b) $SINR_{thresh} = 10dB$, max. degree 9



(c) $SINR_{thresh} = 5dB$, max. degree 9, (d) $SINR_{thresh} = 15dB$, max. degree 9

Figure 7: Average scheduling size as a function of α for 15 sources.

REUSE mechanisms use the routes selected according to their specialized weight metrics.

We may observe that the scheduling size increases with the grid size and $SINR_{thresh}$, and decreases with α . For smaller values of α and higher values of $SINR_{thresh}$, it is increasingly difficult to the link scheduling mechanisms to schedule links simultaneously. Therefore, for $SINR_{thresh}$ greater than $20dB$ and $\alpha = 3$ (Figure 5(d)), the performance of all scheduling mechanisms is similar. As the grid grows, the lengths of the routes between the sources and the gateway get bigger, increasing the scheduling size.

In all simulation scenarios, our proposed mechanism requires a smaller number of slots to schedule all links in the network according to their traffic demands. This guarantees a higher throughput, proportional to the reduction in the num-

ber of slots required to schedule all links. As the size of the routes increases, the performance gain of the proposed mechanism is higher. On the other hand, the LDF and IRMA algorithms present the worst performances, with the scheduling size increasing almost linearly with the grid size for the LDF algorithm. Moreover, similar to the first results presented in this section, the results obtained by the LP formulation show that the performance of REUSE stay away from the optimal, especially for larger grid sizes. It shows that despite of being able to provide better results than other solutions in the literature by the use of the proposed link weigh metric, the REUSE mechanism is still unable to achieve the optimal performance.

The curves presented in Figures 7 and 8 show the results obtained in 300 pseudo-random topologies composed of 50 nodes. Those results present the average scheduling size with their respective confidence interval calculated with a confidence level of 95%, which are very small to appear in the curves. The routes used by GPHY and LDF mechanisms are the shortest path routes ($H = -1$). In these simulations, we vary the number of sources and the maximum degree as a function of α and $SINR_{thresh}$, respectively. We also do not solve the LP formulation for those random scenarios due to the high number of scenarios and the high computational costs involved.

By the analysis of all results presented in these pseudo-random scenarios, some considerations may be taken related to the performance of the simulated mechanisms. In all random topology scenarios, the REUSE provides better scheduling, 20% to 30% smaller than GPHY algorithm. On the other hand, the LDF and IRMA mechanisms perform far worse than the others in all scenarios. The modification of maximum degree used in the topology generation has low impact on the results, and the scheduling size decreases for higher values of maximum degree. This occurs because network connectivity increases, reducing the number of hops of the routes. The impact of $SINR_{thresh}$ and α is the same as discussed in the previous scenarios.

Simulations were performed to evaluate how much of the performance gains of the proposed mechanism were obtained due to the modifica-

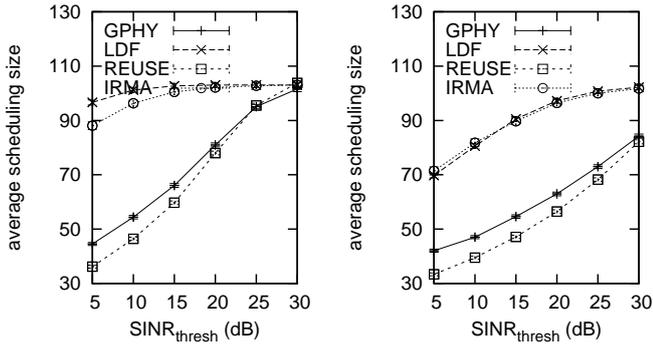
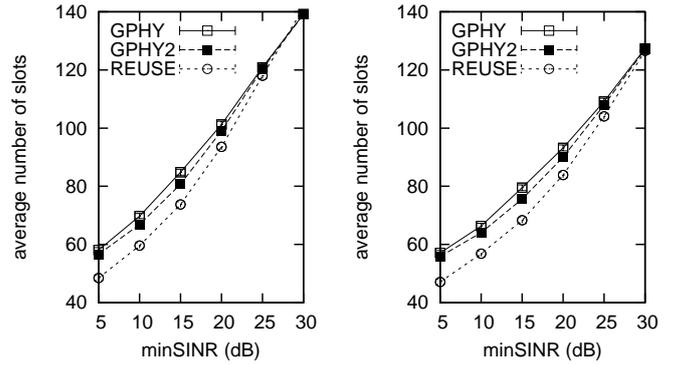
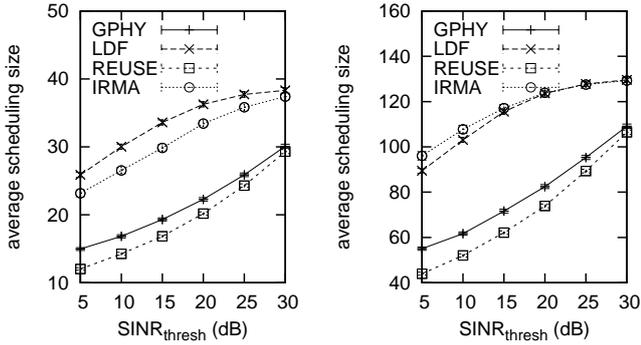
(a) 15 sources, $\alpha = 3.5$ (b) 15 sources, $\alpha = 5$ (a) 20 sources, $\alpha = 4$ (b) 20 sources, $\alpha = 4.5$ (c) 5 sources, $\alpha = 5$ (d) 20 sources, $\alpha = 5$

Figure 8: Average scheduling size as a function of $SINR_{thresh}$ for maximum degree 9.

tions in the GPHY algorithm or due to the proposed weight metric. The graphs in Figure 9 show the average scheduling size as a function of $SINR_{thresh}$ for the GPHY and REUSE algorithms and the GPHY algorithm without the route selection with the proposed weight metric (GPHY2). These results show that each separate part of the proposal (modification of GPHY and proposed weight metric) produces significant performance gains.

6. Acknowledgments

This research was supported by FAPERJ, CAPES, CNPq, UFAM, FAPEAM, SECT/AM and FINEP.

7. Conclusions

The appropriate selection of a link scheduling algorithm allows increasing the capacity of wireless mesh networks. However, the performance

Figure 9: Comparison of REUSE performance gains (max. degree 6).

achieved by those mechanisms is strictly related to the routes used to forward the flows in the network. Therefore, the isolated use of a link scheduling algorithm does not guarantee a significant capacity increase. Therefore, this work investigates the performance losses caused by the lack of interaction between these two techniques.

This work proposes a new joint mechanism for routing and link scheduling, named REUSE, which takes advantage of spatial reuse, favoring simultaneous activation of links. Moreover, the proposed mechanism considers the traffic demands and the allowed interference levels in the link scheduling. The performance evaluation shows the superiority of the proposed mechanism when compared to three other algorithms presented in the literature. However, its performance is still far from the optimal results obtained through Linear Programming formulation. It shows that the joint routing and link scheduling problem in wireless mesh network is not a trivial task, and indicates that there is still space for more performance improvements.

References

- [1] P. Gupta, P. R. Kumar, The Capacity of Wireless Networks, *IEEE Transactions on Information Theory* 46 (2) (2000) 388–404.
- [2] L. Tassiulas, A. Ephremides, Jointly optimal routing and scheduling in packet ratio networks, *IEEE Transactions on Information Theory* 38 (1) (1992) 165–168.

- [3] K. Jain, J. Padhye, V. N. Padmanabhan, L. Qiu, Impact of Interference on Multi-hop Wireless Network Performance, *Wireless Networks* 11 (4) (2005) 471–487.
- [4] G. Brar, D. M. Blough, P. Santi, Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks, in: *ACM MobiCom'06*, 2006.
- [5] L. Badia, A. Ertas, L. Lenzini, M. Zorzi, A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks, *IEEE Network* 22 (1) (2008) 32–38.
- [6] O. Goussevskaia, R. Wattenhofer, M. M. Halldorsson, E. Welzl, Capacity of Arbitrary Wireless Networks, in: *IEEE Infocom'09*, 2009.
- [7] J. Li, C. Blake, D. S. De Couto, H. I. Lee, R. Morris, Capacity of ad hoc wireless networks, in: *ACM MobiCom'01*, 2001.
- [8] M. Grossglauser, D. N. C. Tse, Mobility increases the capacity of ad hoc wireless networks, *IEEE/ACM Trans. Netw.* 10 (4) (2002) 477–486.
- [9] A. Behzad, I. Rubin, On the Performance of Graph-based Scheduling Algorithms for Packet Radio Networks, in: *IEEE Globecom'03*, 2003.
- [10] M. Kodialam, T. Nandagopal, Characterizing Achievable Rates in Multi-hop Wireless Mesh Networks with Orthogonal Channels, *IEEE/ACM Transactions on Networking* 13 (4) (2005) 868–880.
- [11] L. Badia, A. Botta, Evaluation of Various Interference Models for Joint Routing and Scheduling in Wireless Mesh Networks, in: *IEEE Globecom'08*, 2008.
- [12] Y. Wang, W. Wang, X.-Y. Li, W.-Z. Song, Interference-Aware Joint Routing and TDMA Link Scheduling for Static Wireless Networks, *IEEE Transactions on Parallel and Distributed Systems* 19 (12) (2008) 1709–1726.
- [13] Z. Wu, D. Raychaudhuri, Integrated routing and MAC scheduling for single-channel wireless mesh networks, in: *IEEE WoWMoM'08*, 2008.
- [14] X. Wang, J. Garcia-Luna-Aceves, Embracing Interference in Ad Hoc Networks using Joint Routing and Scheduling with Multiple Packet Reception, *Ad Hoc Networks* 7 (2) (2009) 460–471.
- [15] Z. Wu, S. Ganu, D. Raychaudhuri, IRMA: Integrated Routing and MAC Scheduling in Multi-hop Wireless Mesh Networks, in: *IEEE WiMesh'06*, 2006.
- [16] C. Barnhart, J. Wieselthier, A. Ephremides, Neural Network Techniques for Scheduling and Routing Problems in Multihop Radio Networks, in: *IEEE Milcom'91*, 1991.
- [17] S. Borbash, A. Ephremides, Wireless Link Scheduling With Power Control and SINR Constraints, *IEEE Transactions on Information Theory* 52 (11) (2006) 5106–5111.
- [18] Y. Shi, Y. T. Hou, J. Liu, S. Kompella, How to Correctly Use the Protocol Interference Model for Multi-hop Wireless Networks, in: *ACM MobiHoc'09*, 2009.
- [19] S. Gandham, M. Dawande, R. Prakash, Link Scheduling in Sensor Networks: Distributed Edge Coloring Revisited, in: *IEEE Infocom'05*, 2005.
- [20] P. Chaudhuri, H. Thompson, A self-stabilizing algorithm for the st-order problem, *Int. J. Parallel Emerg. Distrib. Syst.* 23 (3) (2008) 219–234.
- [21] GNU Project, GLPK (GNU Linear Programming Kit), <http://www.gnu.org/software/glpk/> - last access in 9/1/2010 (2010).