

The Impact of Joint Routing and Link Scheduling on the Performance of Wireless Mesh Networks

Carlos H. P. Augusto and Celso B. Carvalho and Marcel W. R. da Silva and José F. de Rezende

Grupo de Teleinformática e Automação

Universidade Federal do Rio de Janeiro (UFRJ)

P.O. Box 68.504 – Rio de Janeiro, RJ 21.945–970

Emails: {chenrique,celso,marcel,rezende}@gta.ufrj.br

Abstract—Increasing the capacity of wireless mesh networks has motivated numerous studies. In this context, link scheduling has been one of the most investigated mechanisms. However, the isolated use of scheduling mechanisms does not guarantee a significant increase in capacity independently of the routes followed by traffic. For this reason, the use of cross-layer optimization techniques involving joint use of scheduling and routing has been recommended in the literature. This paper demonstrates the loss of efficiency caused by the lack of interaction between these two mechanisms. Then, a joint mechanism is proposed, evaluated and compared through simulations. Results show the gains obtained by using this approach.

I. INTRODUCTION

The study of the capacity of wireless mesh networks (WMNs) has received a great deal of attention in the last years [1]–[4]. These previous works show that the capacity of WMNs depends on several factors such as the number of nodes, the degree of interference among links and the demand of traffic. A methodology commonly used to estimate an upper-bound for the network capacity assumes that the link scheduling occurs in slots of time, where links that do not present mutual interference may be activated in 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 cycle, this minimum sequence of slots corresponds to the maximum capacity of network under asymptotic traffic conditions. Thus, the goal is to minimize TDMA cycle length in order to maximize capacity. On the other hand, this same TDMA cycle 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 idea of coupling routing and link scheduling mechanisms comes from the fact that the 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 demand by assigning weights to links that are traversed by traffic flows. However, despite of having knowledge of the demand associated to each link, most

algorithms do not take into consideration the sequentiality in activation of the links that compose a multi-hop route. Thus, links are scheduled a certain number of times in each TDMA cycle, according to their demand, and regardless of the order in which they are scheduled.

Few works jointly define how links should be scheduled and which routes should be chosen to meet a certain traffic demand in a minimum time. Those works show that the combined use of routing and link scheduling mechanisms provides a better performance than the independent use of them [3], [5], [6]. However, this joint optimization, named cross-layer, uses most of the time linear programming optimization approaches to obtain these results. Therefore, there are few works that propose combined link scheduling and routing algorithms [7].

At first, this paper evaluates the impact of the routes used by multi-hop flows in WMNs in the performance of link scheduling algorithms. After that, a joint mechanism of routing and scheduling 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 obtained results show the efficiency of the mechanism in comparison to other two algorithms [2], [4] that schedule links traversed by the shortest paths provided by an SPF algorithm.

This paper is organized as follows. The next section presents related works. Section III describes the proposed mechanism. In Section IV, the simulation environment and models used in the performance evaluation of the algorithms are detailed. Section V shows the numerical results obtained through simulation as well as a discussion about them. Finally, Section VI presents the conclusions.

II. RELATED WORKS

The problem of link scheduling in wireless networks has been tackled for a long time [2]–[12]. These works address the problem in an isolated fashion [2], [5], [9] or jointly with routing and topology/power control [3], [6], [8], [10]. According to the used methodology, existing works can be classified as follows: i) optimization by linear/integer programming [10]; ii) graphs coloring [6], [13] iii) approximation algorithms [2], [4]; iv) other (neural networks, genetic algorithms, etc.) [8],

[11]. Concerning the interference model, these studies use the protocol (or UDG) [6], [9] or physical (or SINR) models [2]. 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 [6], [13] based on graph coloring.

A work similar to ours is presented in [7]. 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 [6], [13], [14]. 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 [14] 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 [6] proposes several algorithms, both centralized and distributed, to the coloring problem.

The next two subsections detail the two algorithms that have been used as comparison basis to the proposed one. The first one, the LDF (Lesser Distance First) algorithm [4], 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, the GreedyPhy, deals with heterogeneous traffic demands, presents good performance and was used as basis for our proposal.

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

A. LDF Algorithm

The work in [4] 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.

B. GreedyPhy Algorithm

The work in [2] proposes a centralized algorithm, called GreedyPhysical, which employs the SINR interference model and a conflict graph that models each node 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.

III. 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.

A. 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 [2] with the purpose of coupling it with the routing.

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

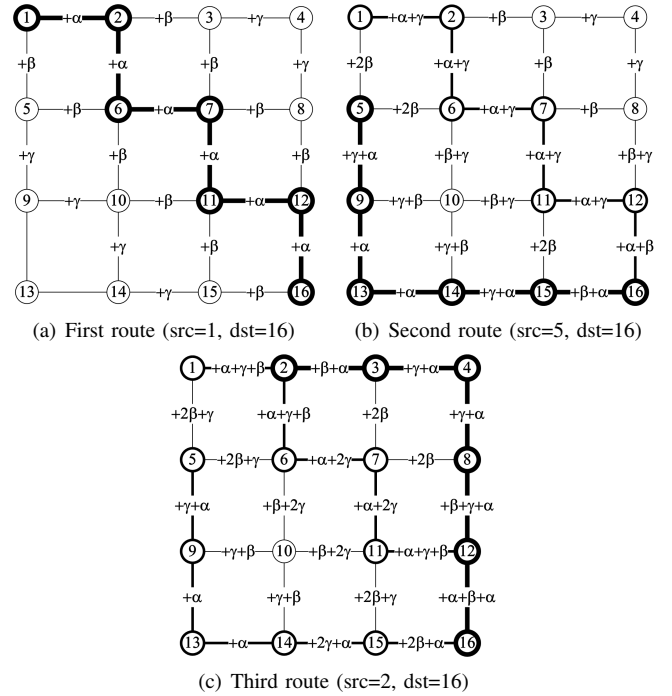


Fig. 1. Three consecutive routes created by the proposed routing metric.

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 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 by α , β and γ the weights of links closest to the selected path. 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.

B. 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 $V = \{v_1, v_2, \dots, v_n\}$ that represents wireless nodes and a edge set $E = \{e_{1,2}, \dots, e_{i,j}\}$ 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) [15], 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 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.

IV. MODELS AND SIMULATION ENVIRONMENT

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) [12]. 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 [12] 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, we have developed our own simulator in which we considered the physical interference model (or SINR) associated with the log-distance propagation model to determine the feasibility in the concurrent activation of links. The physical model establishes that the probability 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 nodes k . 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 $\frac{P_t * k}{d^\alpha}$, 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 k 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 [3]. 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

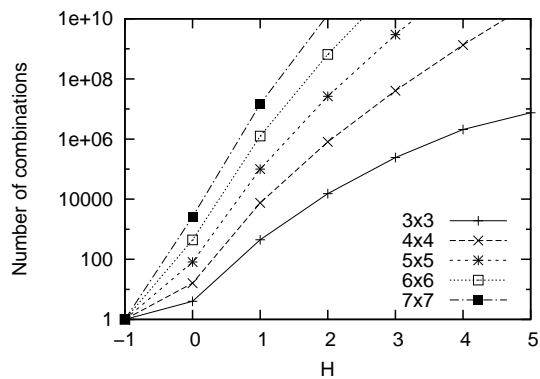


Fig. 2. Number of route combinations as a function of H .

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 mechanism is compared to two algorithms detailed in the subsections II-A and II-B. As these two algorithms do not work together with a specific route selection algorithm, their performance is calculated 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.

V. NUMERIC RESULTS

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 whether 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 parameterization 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. The three evaluated mechanisms and their respective acronyms are: GreedyPhy [2] (GPHY), the algorithm presented in [4] (LDF - Lesser Distance First), and our proposal REUSE.

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 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 mechanism, which chooses a specific route according to the proposed metric presented in Section III. 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

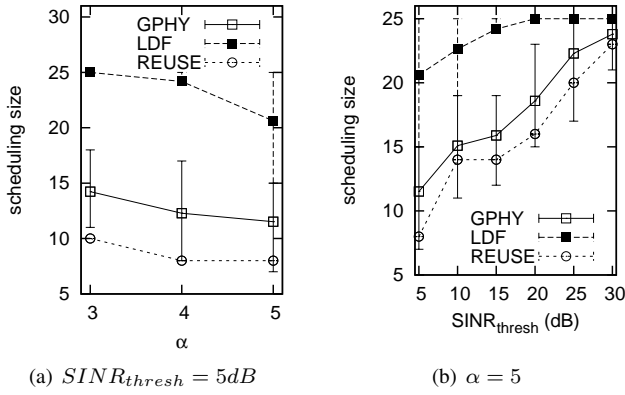


Fig. 3. Average, min. and max. scheduling size (6x6 grid and $H = 0$).

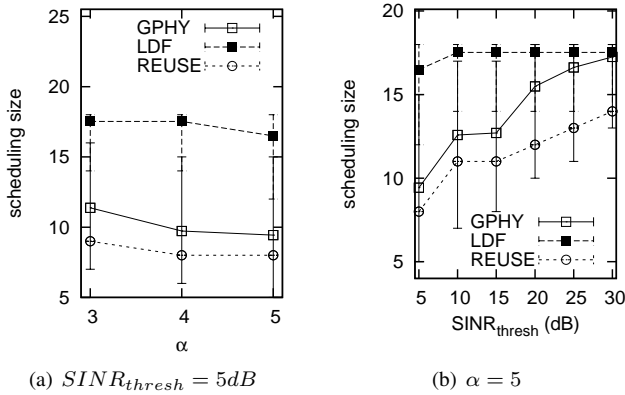


Fig. 4. Average, min. and max. scheduling size (4x5 grid and $H = 1$).

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 obtained by the GPHY algorithm. This indicates that the proposed mechanism is able to find routes that allow better link scheduling.

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.

In what follows, we present the results obtained by increas-

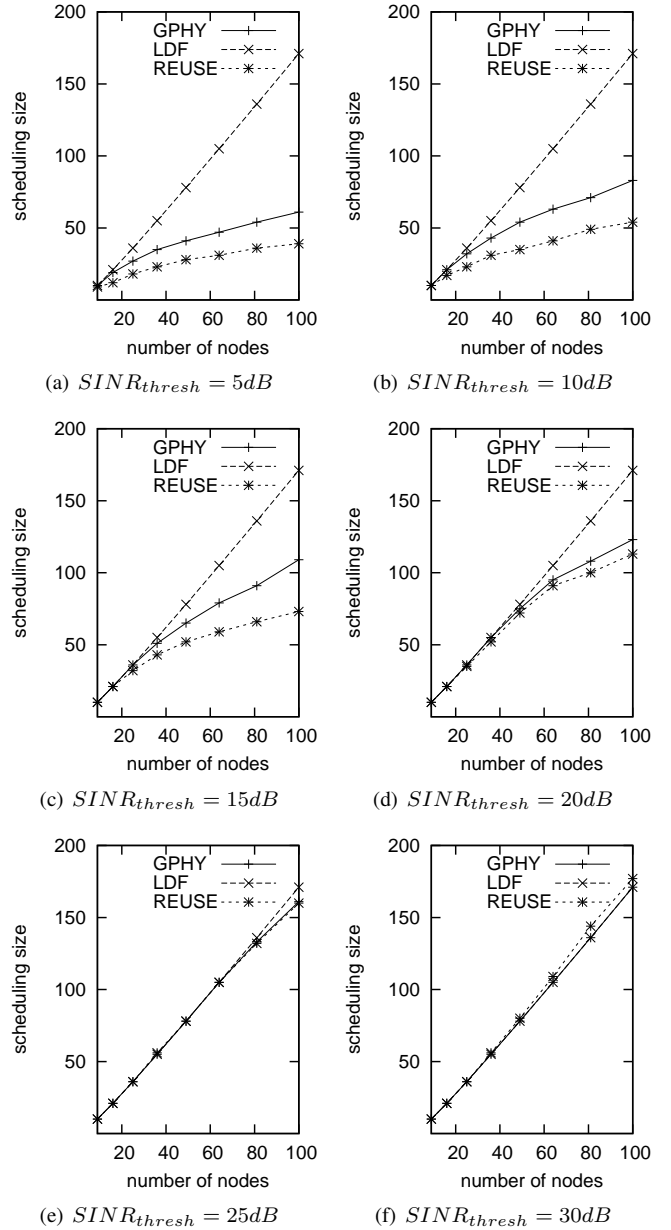


Fig. 5. Scheduling size as a function of the number of nodes for $\alpha = 3$.

ing 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 diametrically 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 REUSE mechanism uses the routes selected according to the weight metric proposed.

We may observe that the scheduling size increases with

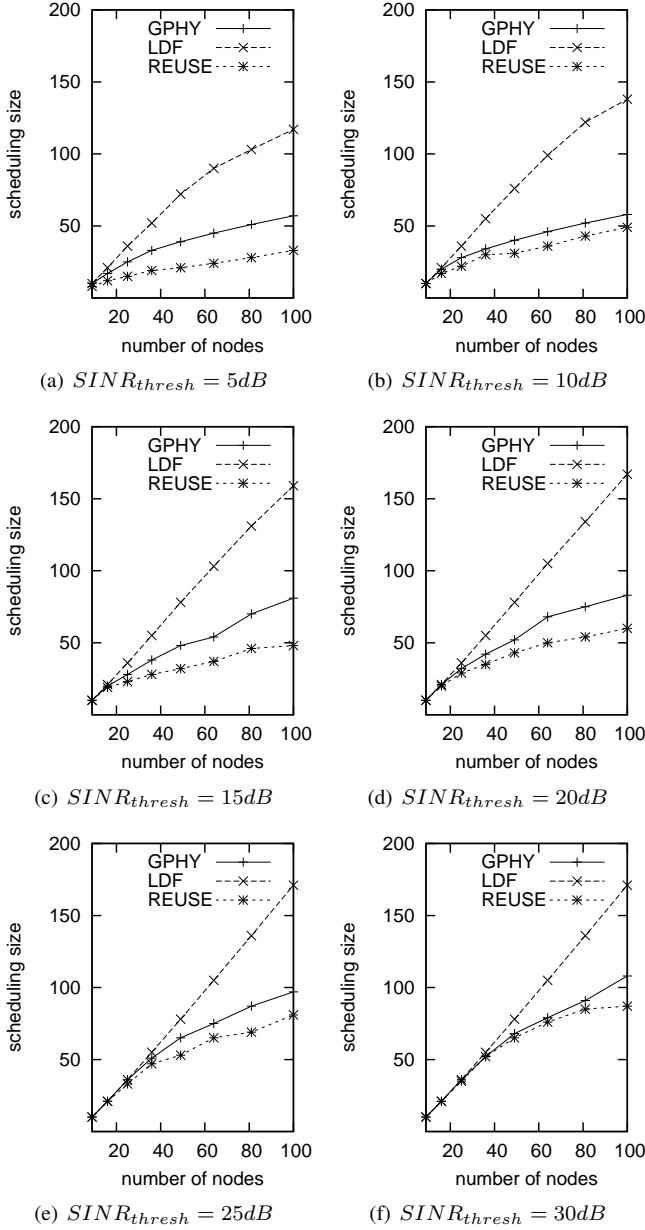


Fig. 6. Scheduling size as a function of the number of nodes for $\alpha = 5$.

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 number of slots required to schedule all links. As the size of the routes

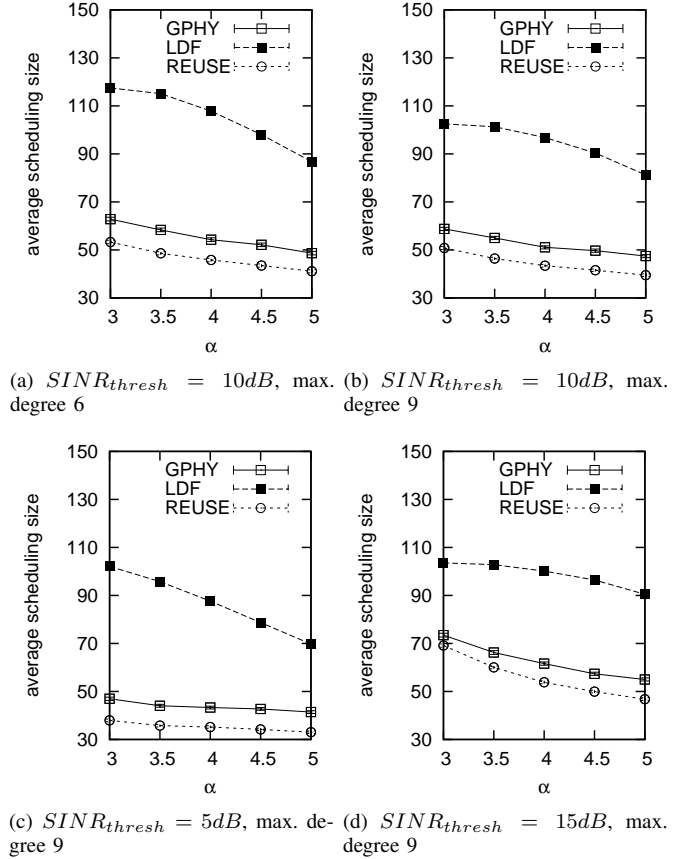


Fig. 7. Average scheduling size as a function of α for 15 sources.

increases, the performance gain of the proposed mechanism is higher. On the other hand, the LDF algorithm presents the worst performance, with the scheduling size increasing almost linearly with the grid size.

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.

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 mechanism performs 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

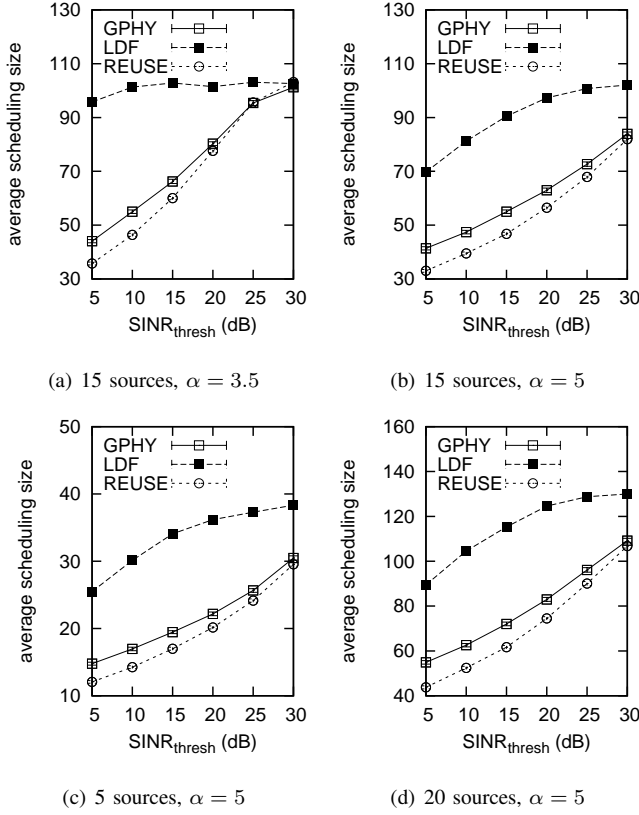


Fig. 8. Average scheduling size as a function of $SINR_{thresh}$ for maximum degree 9.

scenarios.

Simulations were performed to evaluate how much of the performance gains of the proposed mechanism were obtained due to the modifications 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.

VI. CONCLUSIONS

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

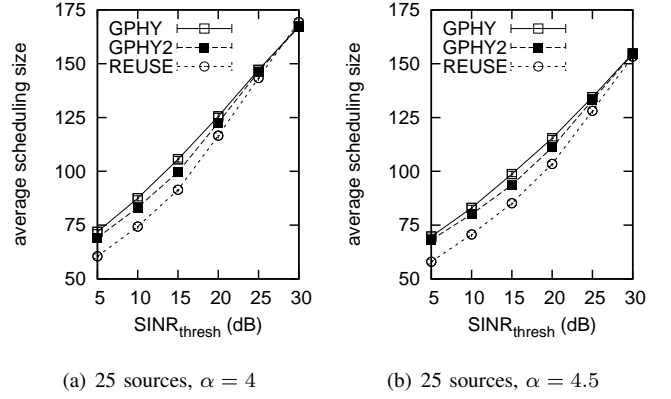


Fig. 9. Comparison of REUSE performance gains (max. degree 6).

performance evaluation shows the superiority of the proposed mechanism when compared to two other link scheduling algorithms presented in the literature.

REFERENCES

- [1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, 2000.
- [2] G. Brar, D. M. Blough, and P. Santi, "Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks," in *ACM MobiCom'06*, 2006.
- [3] L. Badia, A. Erta, L. Lenzi, and M. Zorzi, "A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks," *IEEE Network*, vol. 22, no. 1, Jan. 2008.
- [4] O. Goussevskaia, R. Wattenhofer, M. M. Halldorsson, and E. Welzl, "Capacity of Arbitrary Wireless Networks," in *IEEE Infocom'09*, 2009.
- [5] M. Kodialam and T. Nandagopal, "Characterizing Achievable Rates in Multi-hop Wireless Mesh Networks with Orthogonal Channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, 2005.
- [6] Y. Wang, W. Wang, X.-Y. Li, and W.-Z. Song, "Interference-Aware Joint Routing and TDMA Link Scheduling for Static Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 12, 2008.
- [7] X. Wang and J. Garcia-Luna-Aceves, "Embracing Interference in Ad Hoc Networks using Joint Routing and Scheduling with Multiple Packet Reception," *Ad Hoc Networks*, vol. 7, no. 2, 2009.
- [8] C. Barnhart, J. Wieselthier, and A. Ephremides, "Neural Network Techniques for Scheduling and Routing Problems in Multihop Radio Networks," in *IEEE Milcom'91*, 1991.
- [9] A. Behzad and I. Rubin, "On the Performance of Graph-based Scheduling Algorithms for Packet Radio Networks," in *IEEE Globecom'03*, 2003.
- [10] S. Borbash and A. Ephremides, "Wireless Link Scheduling With Power Control and SINR Constraints," *IEEE Transactions on Information Theory*, vol. 52, no. 11, 2006.
- [11] L. Badia and A. Botta, "Evaluation of Various Interference Models for Joint Routing and Scheduling in Wireless Mesh Networks," in *IEEE Globecom'08*, 2008.
- [12] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, "How to Correctly Use the Protocol Interference Model for Multi-hop Wireless Networks," in *ACM MobiHoc'09*, 2009.
- [13] S. Gandham, M. Dawande, and R. Prakash, "Link Scheduling in Sensor Networks: Distributed Edge Coloring Revisited," in *IEEE Infocom'05*, 2005.
- [14] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance," *Wireless Networks*, vol. 11, no. 4, 2005.
- [15] P. Chaudhuri and H. Thompson, "A self-stabilizing algorithm for the st-order problem," *Int. J. Parallel Emerg. Distrib. Syst.*, vol. 23, no. 3, pp. 219–234, 2008.