A Physical Model Scheduler for Multi-Hop Wireless Networks Based on Local Information

Leonardo Badia*, Alessandro Erta*, Luciano Lenzini†, Francesco Rossetto§, and Michele Zorzi§

* IMT Lucca Institute for Advanced Studies, piazza San Ponziano 6, 55100 Lucca, Italy

† Dept. of Information Engineering, University of Pisa, via Diotisalvi 2, 56122 Pisa, Italy

§ Dept. of Information Engineering, University of Padova, via Gradenigo 6B, 35131 Padova, Italy

email: {*l.badia,a.erta*}@*imtlucca.it*, *l.lenzini@iet.unipi.it*, {*rossetfm,zorzi*}@*dei.unipd.it*

Abstract

There is wide consensus that properly taking into account wireless interference is necessary to design high performance link scheduling algorithms for multi-hop networks. However, most approaches in the literature use simplified models, which significantly abstract from the physical behavior of wireless links. Indeed, the main problem in representing the wireless propagation conditions with a proper level of detail is the very large amount of information needed, that includes the wireless link gains between all node pairs. In this paper, we propose a greedy, centralized scheduler which is based on the physical interference model but aims at exploiting local information available at each node, in order to reduce global information exchange and therefore the overhead as well as the computational complexity of the algorithm. We prove the effectiveness of our approach by extensive simulation results. We also show that our system outperforms the most up-to-date benchmark in realistic interference aware schedulers for wireless multi-hop networks.

1 Introduction

In multi-hop wireless networks, routing strategies are used to identify the best paths to deliver information from source to destination terminals. Moreover, since the radio channel is shared, access to it must be managed through proper scheduling algorithms. These two problems can be combined in order to realize a joint routing and scheduling framework, whose theoretical basis has been posed by the pioneering work reported in [11, 20].

In this context, the network is represented as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where nodes in $\mathcal{N}$ are the network terminals and edges in $\mathcal{E} \subseteq \mathcal{N}^2$ are the communication links. In the following these terms (nodes and terminals, or edges and links, respectively) will be used interchangeably. Scheduling and routing are addressed by considering a transmission over a given link $e \in \mathcal{E}$ as corresponding to *activating*, i.e., "turning on," e, which is conversely inactive / turned off, when the link is not used for transmission. By looking at the activations of links in a sequential manner over time, one can determine scheduling for a Time Division Multiple Access (TDMA). The subsequent activation of links from a source node to a destination also implies routing. This framework can address, e.g., the minimization of the time required for the information delivery; this corresponds to an efficient utilization of the network capacity, i.e., allowing the simultaneous activation of a large number of transmissions, while checking at the same time that the active links bring information toward the desired destination.

Even though these theoretical principles are well studied, the network descriptions commonly used often abstract from a detailed characterization of the wireless medium. The only requirements used to determine the admissibility of a scheduling pattern relate to flow conservation and to avoiding the simultaneous utilization of a node for transmission when it is receiving a packet, or vice versa, a condition which in most of the literature [6, 16] is referred to as *primary interference constraint*. Note that this terminology is somehow improper, since this condition does not really depend on wireless interference, but rather on the half-duplex transceiver capability, which limits the number of simultaneous operations which can be performed at a time [2].

In fact, multi-hop radio networks may suffer a severe capacity limitation due to *wireless interference* phenomena. The majority of the approaches to model wireless interference beyond the primary constraint, e.g., involving at least two disjoint pairs of nodes, follow the classification made in [10], which distinguishes between the so-called *protocol* and *physical interference models*. Other variations have been proposed [12], e.g., to take into account additional aspects such as the capture effect, but these proposals can also be related to the aforementioned distinction.

The protocol model describes wireless interference by means of *conflict sets* (sometimes this representation is also

1-4244-2575-4/08/$20.00

translated into graphs called *conflict graphs*). For every edge $e \in \mathcal{E}$ the conflict set $\mathcal{I}(e) \subseteq \mathcal{E} \setminus \{e\}$ is defined as containing all links whose simultaneous activation with e is forbidden due to interference. This is simply modeled as a binary relationship: given a link $f \in \mathcal{E} \setminus \{e\}$, it can either interfere with e, thus is put into $\mathcal{I}(e)$, or not. However, this representation fails to capture that interference has a cumulative effect, since the simultaneous activation of multiple links may cause too high an interfering power for link e, even though none of these links alone is to be considered as interfering with e, and thus does not belong to $\mathcal{I}(e)$.

In spite of this problem, such a model is adopted in most of the literature which deals with routing and scheduling issues to capture wireless interference [4, 8, 15, 17]. However, the more realistic physical interference model should be preferred, as pointed out in [5]. The main problem in applying the physical interference model is instead on the complexity side, because it requires to check the Signal-to-Interference Ratio (SIR) at the receiver's side of all active communication links and to evaluate if it is above a given threshold. This requires a large amount of information, namely the link gains between any pair in $\mathcal{N}^2$.

To overcome this problem, in the present paper we propose to exploit an alternative model proposed in [19] to reduce the complexity of the physical model and enable its use when certain interference terms are difficult to quantify exactly, especially because they consist of many small contributions to the overall interfering power. The rationale behind this model is to define, for each node, a number K of *dominant interferers*, which are, typically, the K closest neighbors. For each node, the channel gains towards its dominant interferers must be known precisely, whereas the rest of the network is simply described in statistical terms. If K is properly chosen, the approximation introduced by this representation is almost negligible, while the complexity of the description can be significantly reduced.

To validate our model, we evaluate the scheduling time on graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ having different kinds of topologies with single-path routing and where the destination set contains only one node. All edges in $\mathcal{E}$ are also directed toward the root, possibly via multi-hop relaying, without multi-path. This choice can be justified by several reasons. On the one hand, this allows us to focus on scheduling only, without being involved in considerations about routing optimality, since in such a topology there is only one possible route from any node to the root. This also allows a simpler implementation of the scheduler, since it is reasonable to use, as will be argued in the following, greedy scheduling strategies which maximize the number of packets forwarded toward the sink. On the other hand, this kind of topology is still realistic, and can actually be envisioned in many implementations of wireless multi-hop networks, such as the IEEE 802.16 Mesh mode operating with centralized scheduling [6].

We will compare our interference model with alternative techniques, using either the protocol model or a different implementation of the physical model [5] and we also use, as a benchmark, the optimal scheduling computed through exhaustive search. We perform several evaluations with the Network Simulator 2 (ns-2) [1]. Numerical results show a very good agreement between the performance limit and our proposed strategy. At the same time, our interference model obtains a significantly better scheduling with respect to the protocol model and we are also able to improve the results obtained in [5], where a physical model is used. This justifies our model as a practical strategy to use the physical interference model in wireless multi-hop networks.

The rest of this paper is organized as follows: after discussing in Section 2 other related papers, we outline in Section 3 the model for the evaluation of interference based on the K dominant interferers. In Section 4 we instantiate the problem for single-path routing topologies and a greedy scheduler, and we discuss how this choice could be extended in a more general context. In Section 5 we discuss implementation details for the greedy schedulers, emphasizing parameter tuning and sensitivities. We present a detailed report of our performance evaluation campaign in Section 6 and finally we conclude in Section 7.

2 Related Work

The general approach used in the literature to describe TDMA scheduling in multi-hop wireless networks can be found in [11, 20]. In [11] the scheduling problem is studied through linear programming, and a polynomial complexity algorithm which solves the pure scheduling problem is given. In [20], and all the extensions to this framework proposed in other works by the same authors, the problems of routing and scheduling for wireless packet networks are framed in the more general context of identifying a suitable *link activation pattern* which satisfies certain optimality criteria and is subject to certain constraints, so that a linear programming framework can be derived. The main goal of this approach is to minimize the delivery time from all sources to all destinations; to this end, the network capacity must be efficiently exploited.

The contribution we give in the present paper can be viewed as an extension of this approach to a more realistic wireless environment. In fact, in [20] a more accurate characterization of wireless interference is left open; in most of the developments derived from this framework, simplified approaches such as the protocol interference model are used. For example, different theoretical aspects of scheduling in multi-hop networks are investigated in [4,8,15,17] by means of link activation schemes which rely on the protocol interference model only.

In more detail, in [8] the problem of delay guarantees in wireless multi-hop networks is studied. Differently from our approach, which closely follows [20], in this paper the per-flow delay is considered instead of the overall time to deliver all the packets to their respective destinations. In [15], the authors outline and investigate from a high level perspective certain bottleneck problems which arise in joint routing and scheduling scenarios. In this way, related performance bounds are highlighted. However, the analysis heavily relies on the protocol interference model, so it is unclear which conclusions can be extended to general wireless scenarios where a different interference model is to be adopted. The contribution of [4] is an analysis of optimal scheduling conditions, again based on the protocol interference model. With this background, a fair scheduling mechanism is proposed and discussed to activate wireless links, based on the maximal clique search over a graph. Finally, the contribution of [16] is to discuss a linear programming approach in order to solve routing and scheduling and to introduce practical algorithms based on efficient heuristics. Similarly to the analysis reported in [20], the only limitation imposed by wireless interference is that nodes can not be active in more than one operation (which can be either a packet transmission or a packet reception). This rationale is extended by the authors in [17] to a case where wireless interference is considered in a broader sense, but again this involves the protocol model only. In this paper instead we consider the problem of obtaining efficient scheduling heuristics when more realistic wireless interference models are considered. Note also that this poses an issue in terms of computational complexity and information exchange, which we will address in the following.

In this sense, our approach is similar to [5], which also focuses on the physical interference model, even though it introduces a simplification to prioritize the links in the scheduler. However, in that paper the challenge of multi-hop transmission is mitigated since intermediate nodes have a backlog equal to the aggregate backlog of all previous nodes. This simplification makes explicit relaying unnecessary (i.e., a packet can be forwarded even if it has not been received by the relay, because the relay backlog has been suitably increased to take these forwarded packets into account). Thus, the algorithm was run on a set of single-hop flows, whose traffic demands were sized so as to take routing into account. In our analysis, we consider multi-hop more explicitly, as nodes are also required to relay packets and therefore the status of the information delivery has to be constantly monitored also at intermediate nodes.

Finally, paper [6] also presents some similarity with our analysis, mostly in the fact that a similar application scenario is considered. In fact, in that paper the specific case of IEEE 802.16 Mesh mode operating with centralized scheduling is addressed, which might result in node placements similar to the topologies considered in the present paper. However, we remark that in [6] again only primary interference constraints are taken into account. For this reason, our investigations can be seen as an extension of this work to a more realistic interference description.

3 Overview of the Interference Model

In this section, we give a brief overview of the interference model reported in [19], which will be used throughout this paper. The interested reader may refer to [19] for further details. In the model, we focus on a specific link (i, j), for which we perform an approximate computation of the interference at receiver j caused by all nodes which transmit packets at the same time as the intended transmitter i. We assume that all nodes transmit with the same power P_T. This assumption is utilized only for the sake of simplicity, but it can easily be removed by a proper scaling of link gains between nodes. If $g(i, j)$ is the channel gain between i and j, we denote with $P_R(i, j) = g(i, j)P_T$ the power received at j when i is sending a packet, with $1[i](t)$ the indicator function that node i is transmitting at time t, and with π_i the so-called *activity factor* of node i, i.e., the long-run fraction of time that the i-th node spends transmitting a packet. Hence, $\mathrm{E}[1[i](t)] = \pi_i$, where $\mathrm{E}[\cdot]$ is the expectation operator. Therefore, the power received by node j because of the i-th terminal is $P_R(i, j)1[i](t)$. Consider now a link $(i, j) \in \mathcal{E}$. The interference at receiver j is caused by all other transmitting nodes in the network, except i. Thus, the aggregate interference, called $\Xi_{i,j}(t)$, is the sum of the contributions coming from these nodes:

$$\Xi_{i,j}(t) = \sum_{k \neq i} P_R(k, j)1[k](t) \tag{1}$$

and its mean and variance are:

$$\overline{\Xi}_{i,j} = \sum_{k \neq i} P_R(k, j)\pi_k \tag{2}$$

$$\sigma^2_{\Xi_{i,j}} = \sum_{k \neq i} P_R^2(k, j)\pi_k(1 - \pi_k) \tag{3}$$

In the second equation, the interferers are assumed to be uncorrelated, so that the variance of the sum of the received powers is the sum of the variances.

Since the aggregate interference power is the sum of many interferers, we postulate that the power statistics can be described by a multimodal Gaussian random variable, also called a Gaussian mixture [18]. At each node we select the K closest nodes which generate the largest amount of interference. These nodes will be referred to as the K *dominant interferers* (whose index w ranges between 0 and $K - 1$) of node j. If it is known which of these K nodes are transmitting, it is also possible to compute the level of interference generated by them. Since the model assumes to know whether the dominant interferers are transmitting or not, their activity factor is either 1 or 0. Thus, they do not contribute to the interference variance but only to the mean, see (3). On the other hand, all other interferers are assumed to create a Gaussian background interference. The overall statistics can be computed by (2) and (3). It is clear that the interference is a Gaussian mixture random variable, where 2^K modes are present, each corresponding to a possible combination of the dominant interferers states (active or inactive). The key purpose of this model is to estimate the probability of error at the tagged node. We assume that the interference statistics has been computed, and that $p(I)$ is the value of its probability density function evaluated at a given interference level I. Also, suppose that the packet error rate (PER) vs. SIR curve of the employed modulation/coding scheme is known. Then, the PER over link $i \to j$ can be computed as:

$$PER_{i,j} = \int_0^{+\infty} PER_{i,j}(I)p(I)dI \qquad (4)$$

This expression can be simplified if a capacity achieving code is employed. For these codes, the PER is 0 if the received SIR is above a certain threshold, and is 1 otherwise. In this case, if the received signal power is P and the threshold SIR is Λ_0, then $PER(I \leq P/\Lambda_0) = 0$ and $PER(I > P/\Lambda_0) = 1$, and (4) reduces to $1 - F(P/\Lambda_0)$. If we assume that the interference is a Gaussian mixture, each Gaussian mode implies a PER equal to $Q[(P/\Lambda_0 - \overline{\Xi}_{i,j}^{(m)})/\sigma_{\Xi_{i,j}}]$, where $\overline{\Xi}_{i,j}^{(m)}$ is the average value of the interference for that mode and $\sigma_{\Xi_{i,j}}$ is its standard deviation. Note that $\sigma_{\Xi_{i,j}}$ does not depend on the specific mode but on the statistics of the background interference, equal for all modes. Finally, the overall PER is:

$$PER_{i,j} = \sum_{m=0}^{2^K-1} \rho(m) Q[(P/\Lambda_0 - \overline{\Xi}_{i,j}^{(m)})/\sigma_{\Xi_{i,j}}] \qquad (5)$$

where $\rho(m)$ is the probability associated to the m-th mode, determined by the activity of the K dominant interferers, which is assumed to be known.

The evaluation of this expression is much simpler than the exact PER evaluation based on all received gains, because the model for the background interference is well defined as soon as its mean and standard deviation are known. This means to estimate the statistics of the aggregate interference, while the gains of the single links are not needed. Moreover, apart from the dominant interferers activity, the model is simple as it only requires to compute complementary Gaussian functions. For this reason, it can be easily incorporated within a scheduling algorithm, which will be performed in the following.

4 Scheduler Design

In designing the scheduler, we have considered different kinds of node placements, both deterministic and random, in an assigned area. A realistic propagation model (e.g., also including fading effects) has been considered, both in the analytical model presented in Section 3 and in the numerical evaluations in the following. Thanks to this accurate radio propagation description, even when the node deployment is regular, channel impairments occur in an unpredictable manner. For what concerns the random placement, the obtained topologies are even more variable as they are rather different in terms of node degree and length of the links. We also remark that we have kept a quite general approach for what concerns lower layers than the scheduler.

About higher layers instead, and more specifically routing, we focused in this paper on graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with only one destination node and where a single path is available from any node to the destination. The main reason to consider single-path topologies is in order to abstract the evaluation of the scheduling performance from routing. In fact, as shown in [13], there are many considerations which would advise for a cross-layer approach where routing and scheduling are performed jointly. If TDMA scheduling is coupled with a sub-optimal routing, it can achieve very low efficiency; even worse, a comparison of scheduling strategies obtained in this case may not reflect reality, since it is biased by routing inefficiencies. In our scenario this problem does not occur, as the routes are uniquely determined, and every non-root node can only transmit over a single edge, i.e., the one toward its parent node. This allows us to decouple the scheduling from the routing problem and to investigate in a more direct manner the application to a scheduling problem of the interference evaluation framework presented in Section 3. In our opinion, studying single-path networks is a first necessary step in order to gain valuable insight on the problems related to scheduling, but our future work will extend these findings to multi-path networks where a routing algorithm is also included.

This scenario corresponds to having a tree topology (either inherently derived from the radio propagation or superimposed by a Minimum-Spanning-Tree algorithm as is done by many routing algorithms [7]), where the root can be seen as a gateway node, reachable via multi-hop by all nodes, and in charge of collecting the information from them. Exchange of packets is allowed only from a node to its parent node in the tree hierarchy; however, all nodes can generate some interference at a receiver node, depending on their physical placement and not on the logical position in the tree. This actually happens in many scenarios, such as Wireless Mesh Networks operating with centralized scheduling in IEEE 802.16 Mesh mode [6], or Wireless Sensor Networks for distributed measurements [3]. In the former case, the gateway (i.e., the root node) is the access point of the wireless network backbone to a cabled connection to the Internet, in the latter it is the central data collecting unit; in general, it is sensible to think of it as the location where the centralized scheduling algorithm is run.

As a consequence of focusing on single-path topologies, we can simplify the computational complexity of the search for the minimal time schedule. In particular, even simple greedy scheduling strategies can be utilized as efficient schedulers, as justified by the following discussion. If $\boldsymbol{\ell} = (\ell_1, \ell_2, \dots, \ell_N)$ describes the queue lengths at all non-root nodes in $\mathcal{N}$, we denote with $\tau(\boldsymbol{\ell})$ the minimum time to deliver all the packets of $\boldsymbol{\ell}$ to the tree root. Also, we call $\mathbf{e}_k$ the canonical base vector equal to 1 at the kth entry and 0 otherwise. It is easy to prove that, if $i \in \mathcal{N}$ is the parent node of $j \in \mathcal{N}$, for any fixed vector $\boldsymbol{\ell}$ the value of $\tau(\boldsymbol{\ell}+\mathbf{e}_i)$ is not greater than $\tau(\boldsymbol{\ell}+\mathbf{e}_j)$. In fact, the transmission of $\boldsymbol{\ell}+\mathbf{e}_i$ can be achieved with the same optimal activation pattern for $\boldsymbol{\ell}+\mathbf{e}_j$ by turning off link $j \to i$ in one of the slots where it is active.

In order to reduce the computational complexity of the scheduling, we might want to define a slot-wise activation criterion. Of course different criteria are possible for the specific choice of which links to activate. However, the reasoning above implies that, within an already existing activation set, turning on another link which is compatible with interference conditions is always beneficial on single-path routing topologies. We can generalize this, when determining the link activation pattern for a given time-slot, stating a practical rule of thumb that the more active links, the better. Therefore, greedy schedulers appear to be appropriate for the scenario under study, since they offer very good performance and also seem to be better implementable in practical scenarios. Also, they allow comparisons to test the goodness of our proposed interference model against existing solutions present in the literature, which also rely on heuristic schedulers often with a greedy rationale. However, we stress that the choice of focusing on heuristic greedy scheduling does not give any advantage to our proposed model, and we may reasonably infer that a similar comparison would also hold for more detailed theoretical investigations performed within an optimization framework, which are however out of the scope of the present paper, being far more difficult to implement and compare.

Finally, we remark that our scheduler is a centralized one. However, nodes employ accurate local information (about the dominant interferers) and the rest of the network is modeled in statistical terms. Thus, our scheduler makes some important steps toward the goal of a distributed system based on the physical model, rather than the oversimplified protocol model.

5 Implementation Issues

In this section, we describe a low-complexity, centralized scheduler for transmitting data in the uplink of the tree topology (i.e., from all nodes to the root). In our setting, time is slotted and in each slot a centralized controller, e.g., located at the tree root, activates some links to transmit data. The key concept in our scheduler is that the nodes are selected in a greedy manner according to their chances of successfully transmitting a packet. This probability is estimated by means of the interference model reported in Section 3.

The scheduler knows the queue status at all nodes. We remark that this is the only information for which a shared knowledge is required: this can be important if a distributed implementation is sought. We also point out that it is possible to achieve good performance also with rather coarse information about queue status; it is often enough to know if the queue is empty, or its length is below or above a certain congestion threshold. Simulation traces show that such a strategy often leads to satisfactory results. Each terminal i is associated with a weight ψ_i, which is the sum of two factors: the probability of successful packet transmission and a function of the queue status. The success probability at a node is computed by means of (5) and multiplied by a constant suppression factor $\alpha \in [0, 1]$ if a node transmitted a packet in the previous slot, in order to improve fairness for nodes who are in disadvantaged positions, which otherwise would have fewer transmission opportunities. The queue status obviously influences the scheduler in the sense that nodes with empty queues are not eligible for transmission. Additionally, in order to keep the queue lengths under control, we assign a bonus b to the nodes with long queues. The weight ψ_i is equal to this bonus b plus the success probability, multiplied (if necessary) by α. In the following numerical evaluations, for a given queue length ℓ, this bonus b follows a linear piecewise function:

$$b = \begin{cases} 0 & \ell < \ell_A \\ \frac{\ell - \ell_A}{\ell_B - \ell_A} & \ell_A \le \ell \le \ell_B \\ 1 & \ell > \ell_B \end{cases} \tag{6}$$

The thresholds ℓ_A, ℓ_B have been empirically set to, respectively, 150% and 250% of the value of the initial node backlog. Note that this additional term is particularly useful in a tree topology since the nodes closer to the root have a higher traffic to deliver, since they also act as relays, but the leaves generally have a greater chance of being scheduled, because they have fewer neighbors and thus less interference.

This is the main loop the scheduler performs in each slot:

1. **select** the candidates
2. **compute** the queue size bonuses
3. **while** (the candidate queue is not empty)
   for (nodeIndex = 1:allNodes)
   (a) update the interference statistics and the packet success probability: for every node j, compute (5) by estimating the probability that each dominant interferers w of j transmits, as equal to its weight ψ_w. Add to all weights the queue bonus.
   (b) pick the best node
   (c) is this link compatible with the existing communications?
   yes) include it in the list of scheduled nodes. For the rest of the time-slot, its activity factor is 1 (i will surely transmit) and the activity factor and weight of every node k which is a neighbor of

i will be set to 0 (k will not transmit). All the neighbors are removed from the candidate list.
no) remove it from the candidate queue. Set its activity factor and weight to 0.

end_for
end_while

In phase 1, the scheduler selects which nodes may be eligible to be scheduled. In the default implementation of the scheduler, only nodes whose weight ψ_i is larger than 0 and whose queue is not empty continue in the following steps.

Step 3 (the *while* loop) can be fully understood realizing that the interference model is completely specified when the number of dominant interferers K and all the received powers and activity factors are defined. The first term is a constant and the second does not change within a slot. However, the activity factors must be given suitable values. If a node cannot be scheduled in this slot (e.g., it has an empty queue or there is a direct link with a node which has been scheduled to transmit), then it is assigned the value 0. Otherwise, the activity factor can be equal to 1 if the node has been scheduled to transmit. In all other cases, when it is not yet defined whether the node may or may not transmit, the activity factor is set to 1/2 if the node has not sent a packet in the previous slot or $\alpha/2$ if it has. Please note the inclusion of the suppression factor α. This reduces, in the interference model, the transmission probability (and therefore, the weight ψ_i) of the nodes which transmitted in the previous slot. Thus, the other terminals will predict a higher SIR and will compute a higher success probability. Finally, if $\alpha = 0$, then weight ψ_i for certain nodes will be 0 and they will not have a chance to transmit in the next slot. We shall discuss the impact of α in Section 6.

In step 3a, all the nodes update their interference model, which means to set the activity factors of the dominant interferers. In step 3c, if the new link does not decrease the SIR of the other nodes below the target level, it will be scheduled in the incoming slot. However, its father and all its children in the tree must have their weight ψ_k set to 0 because they will not transmit due to the half-duplex constraint. If instead the candidate link is incompatible with the links already scheduled, it is discarded and it will not be considered for the rest of the slot as a possible candidate. Thus its weight ψ_i is set to 0.

We point out that step 3c is carried out taking into account all ongoing transmissions in the network. This step is not distributed and requires global knowledge. However, [5] is subject to the same requirement, while in the other steps our scheduler requires less knowledge than the algorithm in [5].

The computational complexity of the scheduler with respect to the network size N depends on the interference statistics model, which affects step 3a. Its complexity is $O(K)$, since the weights of the dominant interferers are updated. Steps 3b and 3c have a constant complexity with respect to N. These operations must be repeated until the queue becomes empty, thus the outer loop is run $O(N)$ times, leading to a total of $O(NK)$. We state that K weakly depends on N, and this statement will be proved in the results section. Thus we approximate in this analysis K as a constant factor, independent of N. Finally, the scheduler must compute the queue size bonus at the beginning of the slot, and this is an $O(N)$ operation. Therefore the computational complexity is linear in the network size N.

Since we assume that the scheduler has perfect channel state information, scheduling errors (that is to say, some links turn out to have an insufficient SIR) cannot happen in our settings.

6 Performance Evaluation

We have quantified the performance of our scheduler in a number of different situations. Our goal was to compare the absolute performance of our proposed scheduler against some recognized benchmarks and to explore which factors impact its performance.

Scenario Description — All our tests have been run on tree topologies composed by a number of nodes ranging from 16 to 31. For all links, we assume a path loss proportional to $d^{-3.5}$, where d is the distance between transmitter and receiver. Additionally, we superimpose a correlated shadowing term modeled as in [9], with a variance of 5 dB and a correlation at 100 m equal to 0.6. Moreover, two classes of topologies have been created. The first type corresponds to the regular node deployment depicted in Fig. 1a. In such a case, each node is 300 m away from its next hop, and the tree is binary and balanced (the difference between the depth of any two leaves is at most one). Observe that, even though the nodes' positions are fixed and regular, the presence of the correlated shadowing, which is introduced in all the investigated topologies, allows us to obtain different values of the path gain for each topology instance. In the second class of topologies, nodes are randomly placed in a 1000 m $\times$ 1000 m square. A tree topology is generated by means of a spanning tree algorithm which chooses the closest node to the center of the square as the root, and allocates children nodes to the already built tree, with a limit on the node degrees set to 3. An example is shown in Fig. 1b. Differently from the previous scenario, the tree is no longer binary and balanced. In spite of these differences, most of the conclusions we derive for these two scenarios are quite similar, so we infer that they are likely to hold true for other cases as well.

Given an N node topology, an $N-1$ node topology is created by removing a leaf picked at random. According to this procedure, given a base tree consisting of 31 nodes (i.e., a full tree where all leaf nodes have depth 4), we generate a sequence of smaller topologies by successively removing

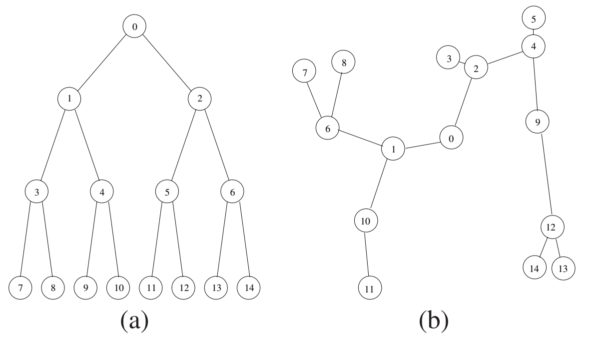

Figure 1. A sample regular (a) and random (b) tree topology

one leaf, until as few as 16 nodes are left (i.e., exactly one leaf node has depth 4). All the curves reported in the numerical results are averaged over 30 different samples, which ensures adequate statistical confidence. Where meaningful, 95% confidence intervals are reported.

The performance of our link scheduling algorithm is assessed by means of the following indices:

- *schedule length*: the duration of the schedule produced by the algorithm in number of slots.
- *end-to-end system throughput* (or simply *throughput*): the overall amount of net user data delivered by the system per unit time.
- *fairness index*: $(\sum_{i=0}^{n} x_i)^2 / \left(n \sum_{i=0}^{n} x_i^2\right)$ [14], where n denotes the number of data flows to the gateway, and x_i the throughput of the *i-th* flow. By definition, the fairness index is bounded in [0, 1] and for equal partitioning of bandwidth is equal to 1.

To investigate the performance of our algorithm in terms of schedule length, we assume that nodes have an integer number of packets in their queues (whose initial sizes need not be equal) and the link rates are normalized to 1. All nodes transmit at a fixed power of 10 dBm. The goal of the scheduler is to transfer all data as quickly as possible from the nodes to the tree root/gateway. If not stated otherwise, the interference model employs $K = 2$ dominant interferers, the suppression factor α is 0 and each node initially has 8 packets in its buffer. The number of dominant interferers K was chosen to be 2 because for higher K performance improvement was found to be negligible. Such a low K strikes a good balance between computational complexity and performance. Moreover, especially in random topologies, it is hard to find many strong interferers which generate comparable interference so that they should all be regarded as dominant. Hence a higher K does not lead to significant performance improvement in this setting. The set of results will explore the scheduler performance as a function of network size, SIR threshold and K.

In addition, the system throughput is analyzed under realistic traffic conditions. Two types of data traffic are used in the simulations, namely Web and Constant Bit Rate (CBR). In the former case, traffic is modeled as a Web source generating variable size packets at variable inter-arrival times. The packet size is distributed as a truncated Pareto random variable with location 10.3 kB, shape 1.1, and cut off 1500 kB. Packet inter-arrival time is exponentially distributed. In the latter case, the source produces packets with length equal to 1000 B at a constant average rate of 50 kB/s. The analysis was carried out by means of Network Simulator 2 (ns-2) [1]. Note that, since we deal with physical realism of the interference models, we utilized, within ns-2 simulation, a more detailed implementation of the physical level, including in particular the additive behavior of the interference. This means that, according to Section 3, the packets need an SIR above the threshold Λ_0 at the receiver's side to be correctly decoded.

Numerical Results — The first test compared the time to empty the node queues for four different systems: our proposed scheduler, the optimal link activation, the protocol model and the physical-model based scheduler by Brar *et al.* [5]. The scheduler by Brar *et al.* is the present benchmark for scheduling based on the physical model. The protocol model is implemented as in [5]; that is to say, whenever a tagged link is activated, it silences all other links whose transmitter or receiver lie inside the interference radius to the tagged link receiver. Observe that, whereas our proposed scheduler and also the one by Brar *et al.* mandatorily verify the feasibility condition for the SIR being above Λ_0 for each activated link, the protocol model, which performs just an approximate computation of the interference, may instead obtain infeasible link activation patterns. When this happens, we assume that an ideal ARQ recovery mechanism is available, which means that the erroneous packets are always detected and immediately (i.e., without delay) notified at the transmitter, which can retransmit them already in the next time slot. This is clearly an optimistic assumption, so the behavior of the protocol model is overestimated; actually, in practical environments, a realistic error recovery mechanism would imply an even worse performance.

The optimal link activation sequence is found by means of an exhaustive search over all possible schedules that are feasible under the physical model.

Figs. 2 and 3 report the results for the regular and random topologies, respectively. They show the ratio between the lengths of the schedules computed by the different approaches and the optimal schedule length. Note that all of the approaches achieve an approximately linearly increasing schedule length in the number of nodes, but with different slopes, that is to say the heights of the curves in Figs. 2-3 (which is the most interesting aspect, as it tells us also how the scheduling algorithms scale with the network size). First of all, the curve relative to our scheduler with queue bonus is usually within 1.1. This means that our schedule is about

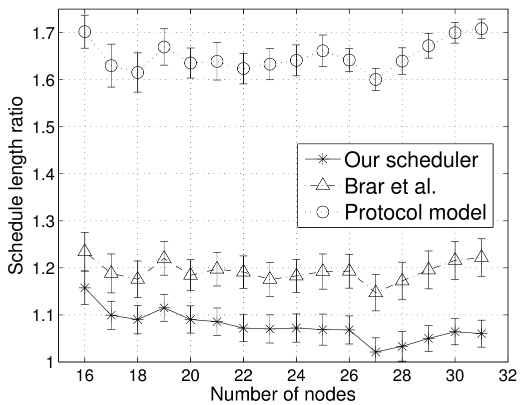

Figure 2. Performance comparison for the regular topologies, $\Lambda_0 = 2.5$ **dB,** $K = 2$

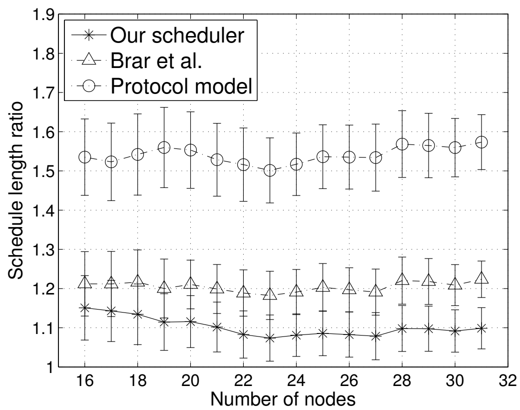

Figure 3. Performance comparison for the random topologies, $\Lambda_0 = 2.5$ **dB,** $K = 2$**,** $\alpha = 0$

10% longer than the optimal one (and often less than that). This is a non-trivial result, since there is no easy way to predict the performance of our scheduler, which could have been anywhere between the lower bound and the protocol model. This fact points out that our algorithm can harness the potential spatial reuse and achieve results which are very close to the optimal scheduling. Moreover, the performance of the algorithm by Brar *et al.* is 20% worse than the optimal schedule. Therefore, we are able to halve the gap with the lower bound, and we can often do better. Incidentally, Brar *et al.* proved that the performance of their algorithm was within a constant multiplicative factor from the optimal schedule. This is confirmed by our graphs. Finally, the protocol model performs rather poorly, because of the low degree of spatial reuse.

Similar reasonings can be applied to Fig. 3, where the random topology case is considered. Again, our scheduler's curve is very close to the optimal policy and thus confirms the adaptability of our method to realistic topologies. In this situation, the algorithm by Brar *et al.* does not significantly improve its performance over the previous case, while we

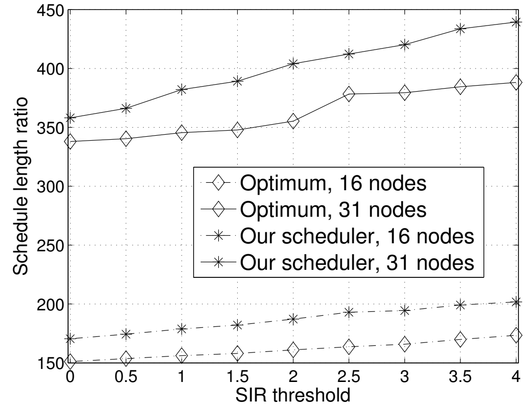

Figure 4. Performance dependence on the decoding threshold, $K = 2$**,** $\alpha = 0$

approach the lower bound more tightly.

Fig. 4 compares the performance of the optimum link activation sequence and our scheduler when the target SIR is changed, in the regular topology case. We note that the curve corresponding to our scheme remains close to the optimal one for all the SIR values. This shows that our approach is robust to the SIR choice. The schedule length increases with a higher SIR because the lower tolerable level of interference decreases the spatial reuse.

A key issue for our scheduler is the determination of the minimum number of dominant interferers used by the interference model necessary for satisfactory performance. It is reasonable to expect that the more the dominant interferers, the better the scheduler performance because the interference model becomes more accurate. However, the computational complexity increases. Fig. 5 explores this tradeoff for the regular topology when the queue size bonus is set to zero. This bonus has been removed in this context because we want to study the influence of the interference model accuracy in isolation. The queue size bonus can mask the different impacts and so it has been turned off for these experiments. Therefore, the curves here do not extend those in the previous graphs (e.g., Fig. 2 or 3). First of all, the mixture model ($K > 0$) yields non negligible improvement over the single mode model ($K = 0$) only for rather large networks (at least 20 nodes). The reason is the following: as pointed out in [19], the Gaussian mixture model works well when there are a few nodes whose power received by a certain terminal is larger than all the rest of the combined interference. When the nodes are few (less than 20) they are usually confined in a small area, and the range of powers received at any point in the tree from all the nodes is within one order of magnitude. Thus, the final interference is not multimodal, but can be already predicted fairly well by a simpler single-mode Gaussian estimator. However, for larger networks dispersed in a wider region, the ratio of the powers between two interferers may become significantly large and result in some noticeable performance difference. This is

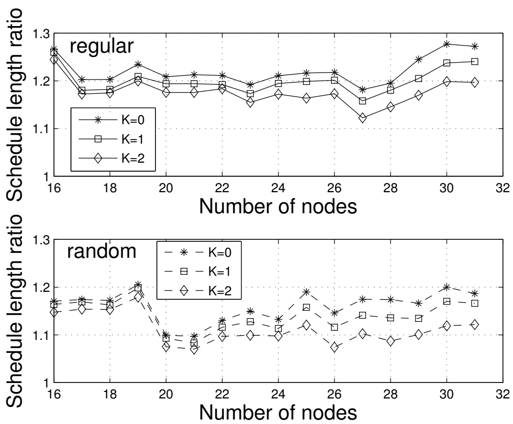

Figure 5. Dependence on the number of dominant interferers, for regular (above) and random (below) topologies, $\Lambda = 2.5$ dB, $\alpha = 0$

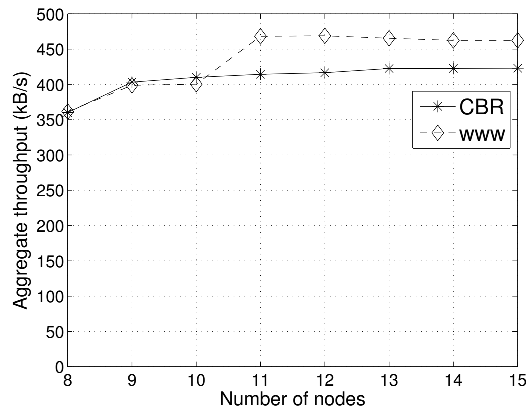

Figure 6. Performance of our scheduler with web-browsing and CBR flows, $\alpha = 0.5$.

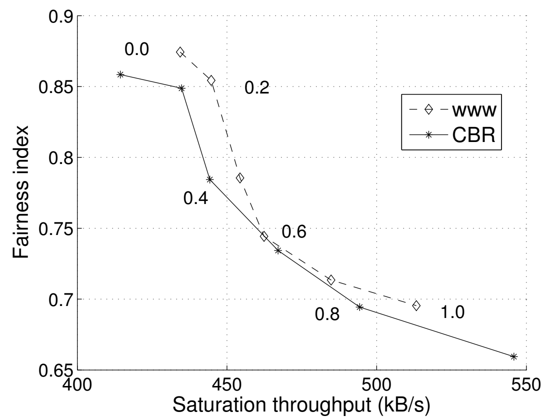

Figure 7. Performance of our scheduler for different values of the suppression factor α.

only partially captured by the average schedule length. In fact, due to the choice of a tree topology, the main bottleneck of the delivery is the tree root, which is independent of the interference evaluation. Thus, also other quantities such as the second order moments should be considered. In any case, the reported difference is about 5–10%. Also note that K is quite low, because no significant performance improvements can be achieved with higher K. Actually, the curves for $K \geq 3$ are not plotted because they are almost indistinguishable from the case $K = 2$. The graphs suggest that the dependence of K on the network size is weak. We conjecture that it is in fact sublinear, but further investigation is still needed in the area. Finally, observe also that the slope of the curves changes, thus we infer that for larger networks the gap would increase, which is also confirmed by preliminary results.

We have also explored how the scheduler performance changes when the queue size is modified. In particular, all our previous simulations considered all nodes to be equally backlogged. We have tested two more scenarios: in the former case, the nodes closer to the root have a longer queue than the leaves, and vice versa for the latter. In the first case the scheduler length is on average shorter (the packets are closer to the root) and the opposite happens in the latter case. But no matter what the load distribution could be, the gap between the optimal scheduling and our system is always in the order of 10% as in the previous cases. Therefore our scheduler is robust to the backlog location.

So far we have proved that our system offers excellent performance compared to the optimal scheduler and [5]. We complete our study by an analysis of our scheduler performance with realistic traffic sources, CBR and Web by means of ns-2.[1] In addition, further insight about the dependence of the scheduler behavior with respect to its fairness parameter α has been sought. In Fig. 6 we have studied the system throughput as the tree size is increased from 8 to 15 nodes. We note that the throughput already saturates at 15 nodes, so we have not analyzed larger networks. Each link has a data rate of 1 MB/s, and the maximum possible capacity in the tree topology is exactly 1 MB/s, because only one of all the links that go into the root can be active per slot. Our scheduler achieves around 45% of this value, which is a significant result considering the interference and topology constraints. As can be seen, when the network becomes overloaded, i.e., the number of nodes is greater than or equal to 11, the overall throughput achieved with the Web source is slightly higher than that with CBR. This is mainly due to the fact that flows experience random bursty arrivals of packets followed by periods of inactivity. Therefore, it may happen that not all flows are active at the same time and traffic experiences a better statistical multiplexing. Fig. 7 shows the throughput in overload conditions versus the fairness index while changing the suppression factor α. We note that there is a tradeoff between the two. This is due to the fact that when α is large (close to 1) all nodes are eligi-

[1] We point out that for web-browsing, UDP has been used as transport protocol, because TCP excessively influences the system performance and its impact on protocol activity would cancel many of the phenomena we are interested in.

ble to be scheduled. This implies that those link activations that enjoy a high spatial reuse may be used very often, and thus the throughput will eventually benefit. However, this also favors those nodes whose interference is inherently low because of their position. Therefore the fairness index will drop. It is also evident that CBR traffic is more affected by α. We believe that this fact is due to the time distribution of the packet arrivals: Web traffic is bursty, and thus terminals in unfavorable positions just have to wait for some traffic to be delivered before having their chance to transmit. On the other hand, CBR will keep busy those nodes in low interference locations, and thus the fairness-throughput curve will be shifted toward the low-right corner of Fig. 7. Incidentally, we observe that when no constraints on node selection are imposed ($\alpha = 1$) the system fairness is nonetheless acceptable (0.65). On the other hand, the ratio between the maximum and minimum saturation throughput is 0.78. This means that even when the candidate selection is strict ($\alpha = 0$) the achieved throughput is still a significant fraction of its best possible value. Hence the scheduler achieves simultaneously high fairness and throughput. Moreover, it is really possible to trade off the two quantities (the curve is smooth and there is no sudden change as the suppression factor changes) and thus α is a design parameter that can be tuned to achieve a desired point in the tradeoff curve. All these observations and findings lead us to concluding that the proposed scheduler is flexible and can perform well in a wide range of scenarios.

7 Conclusions

We have proposed a high performance centralized scheduler for wireless multi-hop networks based on the physical interference model, rather than the protocol model. The scheduler is based on a low complexity model for aggregate mutual interference between nodes, whose complexity is linear with the network size. We have evaluated the robustness of our algorithm with respect to some important system parameters (detection SIR and backlog distribution in the network) and we have shed some light on the dependence of the scheduler's performance on some of its parameters (number of dominant interferers or suppression factor). We are also able to outperform other proposed models which represent the benchmark for computationally efficient, physical-model based schedulers, achieving a gain larger than 50% in approaching the theoretical optimum.

Our future work will study the relaxation of some assumptions (for instance the perfect channel state information assumption). Moreover we are working toward a distributed version of our algorithm, for which problems like disseminating information about queue sizes, obtaining the status of the dominant interferers or coping with imperfect channel state estimates in the previous slot must be solved.

References

[1] Network simulator 2. http://www.isi.edu/nsnam/ns/.

[2] L. Badia, A. Erta, L. Lenzini, and M. Zorzi. A general framework of interference models for joint routing and link scheduling in wireless mesh networks. *IEEE Netw.*, 22(1):32–38, Jan.-Feb. 2008.

[3] S. Basagni, M. Conti, S. Giordano, and I. Stojmenović, editors. *Mobile Ad Hoc Networking*. IEEE press and John Wiley & Sons, New York, 2004.

[4] N. Ben Salem and J.-P. Hubaux. A fair scheduling for wireless mesh networks. In *Proc. WiMesh*, Santa Clara, CA, USA, Sept. 2005. invited paper.

[5] G. Brar, D. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *Proc. ACM MobiCom*, pages 2–13, Los Angeles, CA, USA, Sept. 2006.

[6] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu. Modelling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode. In *Proc. ACM MobiHoc*, pages 78–89, Urbana-Champaign, IL, USA, May 2005.

[7] J. Chang and L. Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *Proceedings of Allerton Conference*, Monticello, IL, Sept. 1999.

[8] P. Djukic and S. Valaee. Distributed link scheduling for TDMA mesh networks. In *Proc. IEEE ICC 07*, Glasgow, Scotland, June 2007.

[9] M. Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electronics Letters*, 27(23):2145–2146, Nov. 1991.

[10] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Trans. Inf. Theory*, 46(2):388–404, Mar. 2000.

[11] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Trans. Inf. Theory*, 34(5):910–917, Sept. 1988.

[12] A. Iyer, C. Rosenberg, and A. Karnik. What is the right model for wireless channel interference? In *Proc. Qshine*, Waterloo, ON, Canada, Aug. 2006. invited paper.

[13] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *Wireless Networks*, 11(4):471–487, July 2005.

[14] R. Jain, W. Hawe, and D. Chiu. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Tech. Rep. DEC-TR-301, Digital Equipment Corporation, 1984.

[15] J. Jun and M. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wirel. Comm. Mag.*, 10(5):8-14, 2003.

[16] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *Proc. ACM MobiCom*, pages 42–54, San Diego, CA, USA, Sept. 2003.

[17] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Proc. ACM MobiCom*, pages 73–87, Cologne, Germany, 2005.

[18] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, UK, 2003.

[19] F. Rossetto and M. Zorzi. Gaussian approximations for carrier sense modeling in wireless ad hoc networks. In *Proc. IEEE Globecom 07*, Washington, DC, USA, Nov. 2007.

[20] L. Tassiulas and A. Ephremides. Jointly optimal routing and scheduling in packet radio networks. *IEEE Trans. Inf. Theory*, 38(1):165–168, Jan. 1992.