Computer Networks 99 (2016) 125–133

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Scheduling wireless links by vertex multicoloring in the physical interference model

Fabio R.J. Vieira*, José F. de Rezende, Valmir C. Barbosa

Programa de Engenharia de Sistemas e Computação, COPPE, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21941-972 Rio de Janeiro, RJ, Brazil

ARTICLE INFO

Article history:
Received 21 October 2015
Revised 3 February 2016
Accepted 4 February 2016
Available online 17 February 2016

Keywords:
Wireless mesh networks
Link scheduling
Physical interference model
Vertex coloring
Vertex multicoloring

ABSTRACT

Scheduling wireless links for simultaneous activation in such a way that all transmissions are successfully decoded at the receivers and moreover network capacity is maximized is a computationally hard problem. Usually it is tackled by heuristics whose output is a sequence of time slots in which every link appears in exactly one time slot. Such approaches can be interpreted as the coloring of a graph's vertices so that every vertex gets exactly one color. Here we introduce a new approach that can be viewed as assigning multiple colors to each vertex, so that, in the resulting schedule, every link may appear more than once (though the same number of times for all links). We report on extensive computational experiments, under the physical interference model, revealing substantial gains for a variety of randomly generated networks.

1. Introduction

Let L be a set of wireless links, each link $i \in L$ characterized by a sender node s_i and a receiver node r_i. Depending on the spatial disposition of such nodes, activating more than one link simultaneously creates interference that may hamper the receivers' ability to decode what they receive. In the physical interference model [1], the chief quantity governing receiver r_i's ability to decode what it receives from s_i when all links of a set S containing link i are active is the signal-to-interference-and-noise ratio (SINR), given by

$$\mathrm{SINR}(i, S) = \frac{P/d^{\alpha}_{s_i r_i}}{N + \sum_{j \in S \setminus \{i\}} P/d^{\alpha}_{s_j r_i}}, \tag{1}$$

where P is a sender's transmission power (assumed the same for all senders), N is the noise floor, d_{ab} is the Euclidean distance between nodes a and b, and $\alpha > 2$ determines the law of power decay with Euclidean distance. We say that a nonempty subset S of L is *feasible* if no two of its members share a node (in case $|S| > 1$) and moreover $\mathrm{SINR}(i, S) \geq \beta$ for all $i \in S$, where β is a parameter related to a receiver's decoding capabilities (assumed the same for all receivers) and is chosen so that $\beta > 1$.

Several strategies have been devised to maximize network capacity, either through the self-contained scheduling of the links in L for activation [2–20] or by combining link scheduling with other techniques [21–29]. All these strategies revolve around formulations as NP-hard optimization problems, so all rely on some form of heuristic procedure drawing inspiration from various sources, some more of an intuitive nature [3–7,9,10,12–17,19–24,26,28], others more formally grounded on graph-theoretic notions [2,8,11,18,29]. Often the problem is formulated in a spatial time-division multiple access (STDMA) framework, that is, assuming essentially that time is divided into time slots, each one accommodating a certain number of simultaneous link activations. In this case, the problem is to find T feasible subsets of L, here denoted by $S_1, S_2, \ldots, S_T$, minimizing T while ensuring that every link appears in exactly one of the T subsets.

* Corresponding author.
E-mail address: fabiorjvieira@gmail.com (F.R.J. Vieira).

http://dx.doi.org/10.1016/j.comnet.2016.02.008
1389-1286/

There is a sense in which this formulation can be interpreted in the context of coloring a graph's vertices. To see this, first recall that to color a graph's vertices is to partition them into independent sets (that is, into vertex subsets that contain no two neighbors in the graph), each of these sets corresponding to a color. An optimal vertex coloring is obtained when no partition into fewer independent sets is possible. Depending on the application at hand, and letting $\mathcal{I}$ denote the set of all the graph's independent sets, it may be necessary to rule out some of the members of $\mathcal{I}$, i.e., to forbid their appearance in any partition. In the specific case of scheduling the links in L for simultaneous activation, we begin by defining a basic conflict graph, denoted by C, whose set of vertices is the set of links L and whose set of edges consists of all pairs of links that are not feasible sets. That is, the edge set of C contains all link pairs (x, y) such that $\{s_x, r_x\} \cap \{s_y, r_y\} \neq \emptyset$ (links x and y have nodes in common), or SINR$(x, \{x, y\}) < \beta$, or SINR$(y, \{x, y\}) < \beta$. If we attempted to schedule the links in L by coloring the vertices of graph C and using each of the resulting independent sets as the set of links to be scheduled in each time slot, clearly some independent set S with $|S| > 2$ might turn up as part of the solution such that SINR$(x, S) < \beta$ for some link $x \in S$. Such a schedule would not do, so we must further restrict the independent sets that the partitioning for vertex coloring can choose from, specifically by forbidding any independent set that is not feasible.

With this notion of a generalized form of graph coloring in place, the schedule given by the sequence $S_1, S_2, \ldots, S_T$ of feasible link sets can be regarded as the product of coloring the vertices of graph C with T colors in such a way that all vertices in S_k get color k. This interpretation suggests a further generalization, now allowing every link to appear not in exactly one of the T subsets but in any number of them, provided this number is the same for all links. What we have now is no longer simply our generalized form of vertex coloring, but a generalized form of vertex multicoloring. To multicolor the vertices of a graph in this generalized sense, and assuming that $q \geq 1$ is the number of distinct colors to be assigned to each vertex, is to identify a certain number of independent sets of graph C (avoiding the forbidden ones) such that every vertex belongs to exactly q of them. To do so optimally is no longer to minimize the total number of colors, but rather to minimize the ratio of such a number to q. Returning to the scheduling context, we no longer look to minimize the number T of time slots, but look instead for the values of T and q that minimize T/q. Now the schedule $S_1, S_2, \ldots, S_T$ of feasible link sets can be regarded as resulting from multicoloring the vertices of C with T colors in such a way that color k is assigned to all vertices in S_k and that every vertex receives exactly q distinct colors.[1]

[1] The reader familiar with the theory of hypergraphs will notice that forbidding independent sets of graph C while coloring or multicoloring its vertices is equivalent to coloring or multicoloring, respectively, the vertices of a hypergraph [30]. In this hypergraph, the vertices are the same as in graph C and the hyperedges are the nonempty subsets of links that are not feasible (and therefore include those pairs of links that are edges in C).

The potential advantages of this multicoloring-based formulation are tantalizing. If the original formulation leads to a number T of time slots while the new one leads to $T' > T$ time slots for some $q > 1$, the latter schedule is preferable to the former, even though it requires more time slots, provided only that $T'/q < T$ (or $qT > T'$). To see that this is so, first note that the longer schedule promotes an overall number of link activations given by $q|L|$ in T' time slots. In order for the shorter schedule to achieve this same number of activations, it would have to be repeated q times in a row, taking up $qT > T'$ time slots.

The possibility of multicoloring-based link scheduling in the physical interference model seems to have been overlooked so far, despite the recent demonstration of its success in the protocol-based interference model [29]. Here we introduce a heuristic framework to obtain multicoloring-based schedules from the single-color schedules produced by any rank-based heuristic (i.e., one that decides the time slot in which to activate a given link based on how it ranks relative to the others with respect to some criterion; cf., e.g., [3,5,9,11,13,17,20]). We use two iconic single-color heuristics (GreedyPhysical [3], for its simplicity, and ApproxLogN [9,17], for its role in establishing new bounds on network capacity), as well as a third one that we introduce in response to improvement opportunities that we perceived in the former two. Incidentally, the latter heuristic, called MaxCRank, is found to perform best both as a stand-alone, single-color strategy and as a base for the multicoloring scheme. All three single-color heuristics run in time polynomial in $|L|$.

Before continuing, we note that the problem we address, that of maximizing network capacity by link scheduling in the physical interference model, though the same as the one considered in [1,3,9,17], is only one of a great variety of problems that likewise must face the many constraints imposed by the need to circumvent the effects of electromagnetic interference in wireless networks. Such problems relate to various aspects of network design, such as node placement [31–34] and frequency assignment [35,36], to name two prominent ones. Some of them take into account some form of end-to-end communication demand [31–35], while others, as in our case in this paper, do not [36]. In a similar vein, the adoption of vertex-coloring and -multicoloring notions to inform our approach is by no means exclusive. In fact, often the proposed solutions to those related problems are closely based on some form of vertex coloring [34–36], including in the case of [35] – and, incidentally, of GreedyPhysical with non-unit demands [3] as well – the possibility of assigning more than one color to the same vertex (though not in as strict a meaning of vertex multicoloring as the one we adopt, since in those cases the number of distinct colors to be assigned to each vertex is fixed beforehand, as opposed to being part of the solution).

We proceed by first discussing single-color schedules in Section 2, where the three heuristics mentioned above are explained in relation to a single overarching template. Then we move to multicoloring-based schedules in Section 3, introducing our heuristic framework for single-color schedules to be automatically turned into multicoloring-based ones. Our computational results are