



Bottleneck Analysis for Routing and Call Scheduling in Multi-hop Wireless Networks

Cristiana Gomes, Stéphane Pérennes, Hervé Rivano

► **To cite this version:**

Cristiana Gomes, Stéphane Pérennes, Hervé Rivano. Bottleneck Analysis for Routing and Call Scheduling in Multi-hop Wireless Networks. 4th IEEE Workshop on Broadband Wireless Access (BWA), Dec 2008, New-Orleans, US, France. 2008.

HAL Id: hal-00371144

<https://hal.archives-ouvertes.fr/hal-00371144>

Submitted on 26 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bottleneck Analysis for Routing and Call Scheduling in Multi-hop Wireless Networks

Cristiana Gomes, Stéphane Pérennes, Hervé Rivano
INRIA/I3S(CNRS-UNIVERSITÉ DE NICE SOPHIA ANTIPOLIS)
2004 route des lucioles - BP 93 FR-06902 Sophia Antipolis, France
{cristiana.gomes,stephane.perennes,herve.rivano}@sophia.inria.fr

Abstract—In this paper, we address the routing and call scheduling problem in which one has to find a minimum-length schedule of selected links in a TDMA (Time Division Multiple Access) based wireless network. As we deal with multi-hop networks, these selected links represent a routing solution (paths) providing enough capacity to achieve the routers requirements of bandwidth. We present a cross-layer formulation of the problem that computes joint routing and scheduling.

We use a branch-and-price algorithm to solve optimally the problem. A column generation algorithm is used to cope with the exponential set of rounds. The branch-and-bound algorithm provides mono-routing. We run experiments on networks from the literature, with different number of gateways. Experimental results as well as theoretical insights let us conjecture that the bottleneck region analysis is enough to find the optimal solution. The *Integer Round-Up Property* (IRUP) seems to hold for our problem.

I. INTRODUCTION

In wireless networks, the communication channels are shared among the terminals. Thus, one of the major problems faced is the reduction of capacity due to interferences caused by simultaneous transmissions [1]. In this work, we call a *round* a collection of links that can be simultaneously activated in the network. We address the problem called *Round Weighting Problem* (RWP) [2] that consider joint routing and scheduling. We present a cross-layer formulation of the problem. We have to find a minimum-length schedule of selected links in a TDMA (Time Division Multiple Access) based wireless network. As we deal with multi-hop networks, these selected links represent a routing solution (paths) providing enough capacity to achieve the routers requirements of bandwidth. Scheduling methods such TDMA can guarantee achieving higher capacities by allowing time slots to be shared by simultaneous transmissions.

A *communication graph* $G = (V, E)$ represents the network topology, where the nodes are the routers and the edges are the links. Interferences between links are given as a *conflict graph* G_c . We consider the RWP as a flow routing problem. Therefore, the flows in the edges of the problem solution represent the allocated bandwidth.

We work with a special case of RWP where data are exchanged only between the routers in V_r and the gateways in V_g such as in Wireless Mesh networks (WMNs) or sensor networks. The input of the RWP corresponds to the communication graph $G(V_r \cup V_g, E)$, the conflict graph G_c representing the edge interferences, and the network bandwidth b_v

required by each router $v \in V_r$. In the output, each edge $e \in E$ receives a positive value b_e that represents the edge bandwidth from the routing problem solution. Simultaneously, we have to find a set of *rounds* R with their weights $w(R_i)$ achieving the routers bandwidth ($\sum_{R_i \ni e} w(R_i) \geq b_e$), such that the total time slots $W = \sum_i w(R_i)$ is minimized.

We use a branch-and-price (BnP) [3] algorithm to solve optimally the problem with integer flows. A column generation algorithm is used to cope with the exponential set of rounds. The branch-and-bound algorithm provides mono-routing by turning integer the flow variables. The mono-routing is interesting to avoid dealing with the packet-reordering problem.

We run experiments on networks from the literature, with different number of gateways. Experimental results as well as theoretical insights let us conjecture that the bottleneck region analysis is enough to find the optimal solution. The bottleneck is usually the gateway considering almost uniform traffic. In particular, the *Integer Round-Up Property* (IRUP) seems to hold for our problem. As defined in [4], an integer programming problem has the IRUP if its optimal value is given by the smallest integer greater than or equal to the optimal value of its LP relaxation.

In this work we study the problem of Routing and Call scheduling in multi-hop wireless networks. It can be considered as an upperbound for networks with distributed links scheduling, like in IEEE 802.11. It can also be useful in a context where centralized links scheduling is possible (e.g. IEEE 802.16) that can directly take advantage of our analysis.

This paper is organized as follows. In the next section, we discuss the related works. In section III we define the problem. In section IV, we give a description of the Branch and Price algorithm adopted. Experimental results and analysis of the bottleneck importance, mono-routing and IRUP are presented in section V. We conclude the paper and give the future directions in section VI.

II. RELATED WORKS

A key issue in wireless networks is the interferences produced between neighboring routers. It is important to know what are the sets of transmission links that can be active at the same time, the rounds. The Round Weighting Problem RWP was introduced in [2]. The authors make a primal/dual analysis and propose approximation algorithms for some specific network topologies. They show the NP-hardness of this

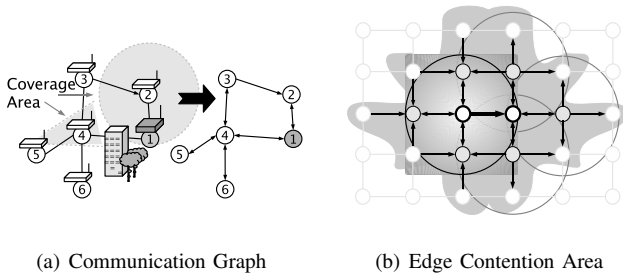


Fig. 1. Network modeling

problem by proving that the well-known NP-hard problem of finding the *Fractional Coloring* on unit graphs reduces to it. In a previous work [5], we gave exact bounds for the problem on grids graphs. We mainly prove that the throughput is determined by the bottleneck around the base station.

In [6], a similar problem, the *Round Scheduling Problem* was treated. The relation with the round weighting problem is the following: if one must repeat rounds scheduling many times then the problem is equivalent to the *RWP*. The authors prove lower bounds on the number of rounds required for any two-dimensional grids and describe protocols for $n \times n$ grids with n odd that are optimal. Notice that, in [6] the interference model is not symmetric because they deal with the exact case of gathering (directed interference).

An algorithm enumerating a tractably large subset of simultaneous transmission rounds has been developed in order to compute an approximated solution for maximum throughput using linear programming (LP) in [7]. Solving the full LP problem means generating an exponential set of scenarios which is intractable even for small networks as seen in [8]. To cope with this issue, column generation methods have been introduced, e.g. [9], [10], [11] and [12]. The work in [10] presents a round generation taking into account the SINR (Signal to Interference plus Noise Ratio) constraints. In [9], a multi-objective study about the Round Weighting Problem was presented that can be used to find bottlenecks position. To solve integer programs with a huge number of variables, the present work use a Branch-and-Price (BnP) [3] algorithm that combines Branch-and-bound with Column Generation. It is an extension of the work in [13].

III. HYPOTHESES AND PROBLEM DEFINITION

In this section we give some definitions that will help to understand the problem. The *RWP* can be modeled as a graph problem. A wireless topology is represented as a digraph $G = (V, E)$, the *communication graph* as illustrated in Figure 1(a).

The set of routers is represented by the set of nodes V . The set of edges $E \subseteq V \times V$ corresponds to the communication links from the real network. If a router j is located within the transmission range of a router i , considering range distance, obstacles, directional antennas, etc, then there is an edge $(i, j) \in E$.

The definition of round is flexible and permit the utilization of either binary models or physical models. The work in [10]

presents a round generation taking into consideration the SINR constraints. As our analysis is based on flow contention (bottleneck) similar results should be obtained independently of the adopted model.

In the present work, a round is simply defined by any set of edges pairwise at distance at least d . It defines a symmetric interference model that permits the calls happen in both directions (download or upload). The most basic case is $d = 1$, where the set of rounds \mathcal{R} is simply the set of the matchings of G . In other words, an active link (i, j) interferes with another links located within the interference range defined by d of the router i . The set $I_{u,v}$ is composed by all links interfering with the link (u, v) that defines the *contention area*. In Figure 1(b), we can see an example in a grid graph.

We consider a symmetrical interference model (a link interferes all other links at distance d). It implies a symmetrical contention area in a way to permit communication in both directions. That is all edges in bold in Figure 1(b) will be interfered in both directions. This distance d is the same to all links. The conflict graph G_c is the line graph power d , $L^d(G)$. For example, if $d = 2$ a *round* is an *independent set* of G_c or an *induced matching* in G . In this case, the *RWP* is a *strong edge-coloring* (or a $L(1,1)$ *edge labeling*) of the G with edge weights defined by the flow going through them given by a routing. These weights are simultaneously defined representing the best routing possible to reach the minimum number of colors W .

The bandwidth should be allocated between the set of nodes V_r and the set of gateways V_g ($V_r \cup V_g = V$ and $V_r \cap V_g = \emptyset$). The input of the *RWP* corresponds to the communication graph $G(V_r \cup V_g, E)$, the conflict graph G_c representing the edge interferences, and the network bandwidth b_v to each router $v \in V_r$. Let $B = \sum_v b_v$. In the output, each edge $e \in E$ receives a positive value b_e that represents the edge bandwidth from the routing problem solution. Simultaneously to the routing problem, we have to find a set of *rounds* R with their weights $w(R_i)$ achieving the routers bandwidth ($\sum_{R_i \ni e} w(R_i) \geq b_e$), such that the total time slots $W = \sum_i w(R_i)$ is minimized. From this set of rounds can be deduced the paths followed by the data.

IV. BRANCH AND PRICE ALGORITHM

The BnP combines Branch-and-bound (BnB) with Column Generation to solve integer program with a large number of variables. We use a Branch-and-Price (BnP) algorithm to turn integer the flow variables x of the model presented in IV-A.1.

As the BnP algorithm is well known, we give only a brief description. Each node of the BnB tree corresponds a linear program to be solved, a *constraint stack* that should be considered in this linear program, a *list of variables* that should be integer and a position in the *execution stack*. Each time, the *execution stack* saves only a part of the BnB tree. As we use a Depth-First approach, these parts correspond to exactly paths that we follow deep into the tree. We stop when we find the best objective, otherwise, a backtracking is needed.

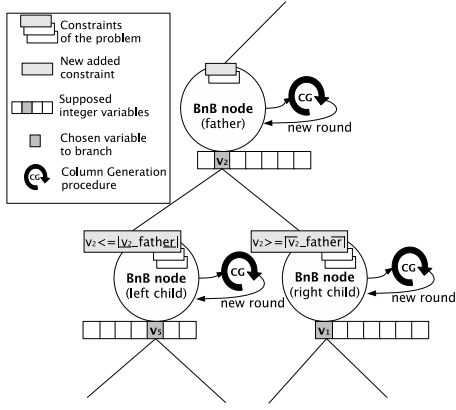


Fig. 2. The Branch and Price algorithm

At each node of the tree, we solve a linear relaxation of the problem with column generation. Therefore, each node is repeated until no further variables price out favorably. With the optimal solution of the column generation we have a *list of variables* and we can choose among them how to bound the children nodes. We choose a non-integer variable to branch from the list of variables. We respect a priority order, the first variables corresponds to the flow variables around the gateway and the next ones are chosen randomly. We explain in the section V why this priority order was adopted.

We start with a relaxed model at the root node with an empty *constraint stack*. For the execution of each node of the Branch-and-Bound tree we create two children with a copy of the father *constraints stack* and a new constraint each one. We push a child node of the BnB tree on the execution stack. This node is then on top of the *execution stack* and it will be executed next. If at any point the relaxed model of a BnB node becomes infeasible, this node is pruned. That is, it is popped out from the *execution stack*.

The best node is a BnB node with all its variables integer and the objective is better than the current best solution. So, this node will be used as current best solution. Otherwise, we continue splitting the problem in one or two new problems branching on a variable of the *list of variables* that is not integer yet. The figure 2 shows an example of a part of the BnB tree of the BnP algorithm.

Actually, we use the classical BnP algorithm with some adaptations to our problem, e.g. the branch process (priorities), which is sufficient since we observed this problem has particular characteristics that can be explored. It permits the algorithm to cut very early the BnB tree (as explained in section V), and the optimal solution is found efficiently.

A. Column Generation Method

The problem considered takes into account the complete set of rounds. Since the number of rounds is exponential, the number of columns of the constraint matrix of our model is also exponential. A column generation (CG) algorithm is used to avoid dealing with the complete set of rounds. Our CG algo-

rithm has an exponential complexity (in the number of links) as well. The difference is that the brute force enumerating algorithms have the average case exponential complexity while the CG algorithm has only worst case exponential complexity.

The linear programming model to this problem is decomposed into a master problem and a sub-problem models as presented in [9]. We show this model in this section for the sake of completeness. Notice that in [9] the flow variables x are not integer that allows splitting the packet generating multiples paths. The present work makes analysis considering also these variables as integer.

We solve the master problem with a small subset of columns, which serves as an initial basis. The sub-problem is then solved to check the optimality of the solution under the current master basis and to generate new columns for the master problem. This procedure repeats until the master problem contains all columns necessary to find the optimal solution of the original problem. Considering the *RWP*, each column corresponds to one round.

In each iteration, if the sub-problem can find a new column that may improve the master solution, this column is inserted in the master basis and a new master solution is computed. If the sub-problem can not find, linear programming and duality theory ensure that the solution of the problem is optimal.

If we does not consider the flow variables x as integer, the master problem is a linear programming problem and therefore can be solved in polynomial time using the *simplex* algorithm. The complexity is solving the sub-problem which is combinatorial, and finding the optimal is NP-hard. We will see in section V that this complexity is limited to only a small part of the graph, the bottleneck region.

1) *Master problem formulation:* We define the following variables: Let the variable $x_{i,j}^v$ be the flow from the router v over link i,j . The variable $x_{i,j}^v$ is turned integer with the BnB algorithm or it remains relaxed when only column generation is employed. The integer demand from each router v is represented by the parameter b_v . Let the binary parameter $a_{i,j}^r$ be 1 if link (i,j) is active in the round r , and 0 otherwise.

Recall that set $I_{u,v}$ is composed by all links interfering with (u,v) . We define $F_{(u,v)}^{(i,j)} = 0$ if $(i,j) \in I_{u,v}$ and 1, otherwise. We define w_r as the fraction of time that round $r \in R$ is scheduled. Consequently, there is an induced edges capacity $c_{i,j} = \sum_{r \in R} a_{i,j}^r w_r, \forall (i,j) \in E$.

The master problem can be defined as follow: Given a graph $G(V_r \cup V_g, E)$, a set of routers demand b_v with $v \in V_r$ and a set of rounds R , the problem is to assign a weight w_r to each round $r \in R$. The weights represent the amount of time a round will be activated. The total amount of time needed to satisfy all demand will be $w_R = \sum_{r \in R} w(r)$. From the edges of the rounds can be deduced the paths followed by the data. The constraints of the master problem expressed as a linear programming model are the following:

$$\sum_{i \in V_r / (v,i) \in E} x_{v,i}^v = b_v, \forall v \in V_r \quad (1)$$

$$\sum_{j \in V_g} \sum_{i \in V_r / (i,j) \in E} x_{i,j}^v = b_v, \forall v \in V_r \quad (2)$$

$$\sum_{i \in V_r / (i,j) \in E} x_{i,j}^v - \sum_{k \in V / (j,k) \in E} x_{j,k}^v = 0, \forall j, v \in V_r \quad (3)$$

$$\sum_{r \in R} a_{i,j}^r \cdot w_r - \sum_{v \in V_r} x_{i,j}^v \geq 0, \forall i, j \in E \quad (4)$$

Constraints (1-3) correspond to the flow constraints. Constraints (1) define the flow leaving its source router and constraints (2) define the flow arriving in a gateway. Constraints (3) represent the flow conservation, that is, the flow entering an intermediate router equals the flow leaving that router. Constraints (4) assign weights to the rounds to satisfy the flow in the edges.

2) *Sub-problem formulation:* The sub-problem generates a round r with the minimal *reduced cost* $\left(1 - \sum_{(i,j) \in E} p_{(i,j)} \cdot a_{i,j}^r\right)$ to enter the master basis. To express the sub-problem as a linear programming model, we have to define some additional notations. Let the parameter $p_{(i,j)}$ be given by the dual variable associated with the constraints (4) of the master problem. Consider the binary variable $u_{(i,j)} = 1$ indicating if the edge (i,j) enters the round to be added to R , $u_{(i,j)} = 0$ otherwise. The sub-problem can be seen as the *Maximum Weighted Independent Set Problem* which is NP-hard [14]. The parameter $p_{(i,j)}$ corresponds to the weight of the edges. The objective function is to maximize the sum of the weights of all active edges respecting interferences. The formulation of the sub-problem is the following:

$$\max \sum_{(i,j) \in E} p_{(i,j)} u_{(i,j)} \quad (5)$$

$$u_{(i,j)} + u_{(k,l)} \leq 1 + F_{(i,j)}^{(k,l)}, \forall (i,j) \in E, \forall (k,l) \in E \quad (6)$$

The objective function (5) searches the maximum weight, which is equivalent to the minimum reduced cost. The parameter $p_{(i,j)}$ guides the column generation to select the best round. Constraints (6) avoid interferences according to the interference model in F .

If the value of the objective function in the sub-problem is strictly greater than 1 (e.g. the reduced cost is negative), a new column $u_{(i,j)}$ is found and the master basis is expanded. Otherwise, the master problem already gives the optimal solution to the original problem.

V. RESULT ANALYSIS

The model¹ is implemented using the AMPL modeling language and the Branch-and-Price algorithm is implemented using ILOG Concert Technology. The instances are solved using the commercial software Cplex version 10, on a desktop PC with one gigabyte of RAM. We use the public graph representations of some mesh networks topologies from [15] in

¹The source code can be found at <http://www.sop.inria.fr/mascotte/personnel/Cristiana.Gomes/implementations.html>

a way to permit the reproduction of our tests. As our results are based on flow contention, they should be adequate for simplest graphs as these ones that represent an idealized view of wireless networks (e.g. unit disk graphs).

In our tests, we consider a simple interference model where each edge interferes with another one if the distance between them in graph G is lower than 2, but the model accepts other interference models. We consider equal bandwidth requirements $b_v = 1$ for all routers, which can be interesting to provide fairness (uniform bandwidth).

We consider the *RWP* as a flow problem. Therefore, the flows in the edges represent the bandwidth. Table I gives the network topologies characteristics. The solutions W_i (W with integer flow) to *RWP* obtained by the BnP algorithm are shown. Table I also shows the solutions W_f (W with fractional flow) to *RWP* obtained using only Column Generation algorithm. The computation time to solve any instance was low, of the order of tenths of seconds.

The IRUP [4] seems to hold for the *RWP* in our tests results. Table I shows that $W_i = \lceil W_f \rceil$. It does not mean that we can obtain an integer routing rounding the edges flow of a fractional solution. The IRUP is due to the fact that there is a huge concentration of traffic around the gateway (critical region) because all the flow goes toward it. Therefore, we give priority to these variables in the BnB algorithm. The work in [16] presents an algorithm that solve polynomially an integer program respecting the IRUP.

A. A Lower Bound

If n routers send the same b_v units of flow during W slots of time, there is nb_v units of flow on the edges around the gateway. If we transform these flows (weights) in new nodes of the conflict graph G_c , we can see large cliques composed of the gateway and its neighborhood. The number of colors to cover the largest one is a lower bound to our problem solution. It is known that $\omega(G) \leq \chi_f(G) \leq \chi(G)$ for any graph G , where $\omega(G)$ is the clique number of G , $\chi_f(G)$ is the fractional chromatic number and $\chi(G)$ is the chromatic number of G .

We observe that the clique number represents already the optimal solution in our tests, as shown in Figure 3 (the letters represent the rounds). This fact is not usual. It happens only in special classes of graphs as the perfect graphs that is not the case of our graphs. Usually the clique number is derived from the bottleneck in the gateway. Otherwise, it will be derived from bottlenecks in other parts of the network. In [9] was presented an algorithm that can be used to find bottlenecks position. We can see in Table I that if there are 2 or more gateways, W is not exactly divided by the number of gateways because they have different absorption rates or because they are close and their critical regions make interference one with the other.

The *RWP* can be easily solved once we have covered the critical region. The links out of this region have *slacks* of activation. An edge have a *slack* when it has several possible options to get activated forming a *round* with edges on the critical region. It is easy to assign time slots (colors) to

TABLE I
NETWORKS TOPOLOGIES AND RESULTS

Network	Gateways	Nodes	Edges	W_f	W_i
A	1	11	34	16	16
A	2	11	34	9.5	10
B	1	12	18	15	15
C	1	15	22	17.666	18
C	3	15	22	7.71428	8
D	1	16	49	18.5	19
D	3	16	49	6.6666	7
E	1	25	45	54	54
E	3	25	45	14.5	15
F	1	28	41	38	38

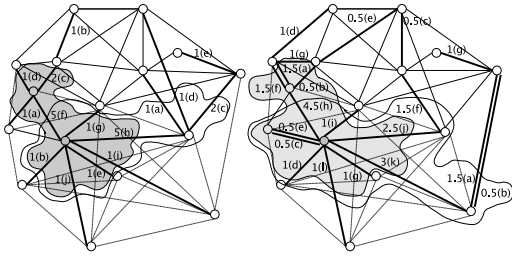


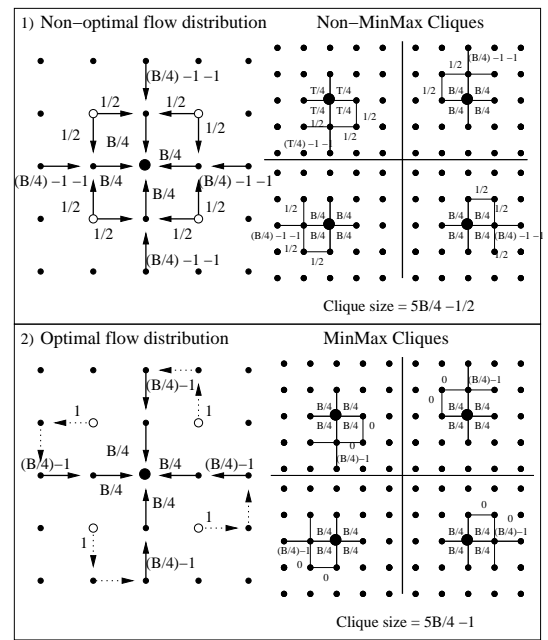
Fig. 3. Network D ($B = 15$) with solution $W_f = 18.5$ and $W_i = 19$

the edges out of this region because once we have covered the critical region they will have several possibilities to get activated without change the total routing time W .

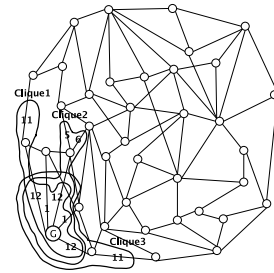
The traffic around the gateway is well distributed between the cliques because the model tries to minimize the maximum clique, it results in cliques of the same size for the fractional result and almost the same size for the integer result. This explains why the integer coloring solution is simply the round up of the fractional coloring ($W_i = \lceil W_f \rceil$). It allows to cut efficiently the BnB tree in the BnB algorithm IV putting the current best solution objective value equal to $\lceil W_f \rceil + 1$.

We notice that the colors (time slots) used in the weighted min-max cliques around the gateway is enough to color all the paths. The weighted min-max cliques can be determined by a distribution of the total flow B over the links of the critical region. The cliques are generated in a way to minimize the maximal one, consequently it will minimize the total number of colors needed (total routing time W).

We make some experiments in grid graphs with the gateway in the center. Figure 4.a.1 shows an example of a bad traffic distribution (the clique have not minimum weight) giving a worse result and Figure 4.a.2 gives the best configuration. The best configuration shows that the 4 white nodes close to the gateway have to go far from it and then come back through the central axes. In fact, this strange routing minimizes



(a) Grid Networks, solution $W = 5 \cdot \frac{B}{4} - 1$



(b) Network with 172 links and 39 nodes, solution $W = 49$

Fig. 4. Weighted Min-Max Cliques

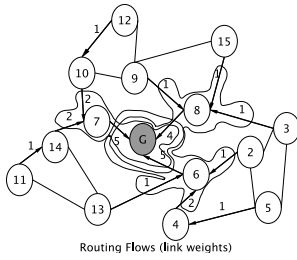


Fig. 5. MST backbone of the network C

the maximal cliques that have weight $W_f = 5 \cdot \frac{B}{4} - 1$ instead $W_f = 5 \cdot \frac{B}{4} - \frac{1}{2}$ (with the non-minmax cliques). So, we gain $\frac{1}{2}$ of time, and it corresponds to the optimal solution as proved by [5]. Another results can be seen in Figure 4, we test a larger network computing only W_f . The result is already integer and if we are right it is also the result using integer flow.

B. Mono-routing

As $b_v = 1$, the BnP algorithm give us a mono-routing that can be interesting to avoid dealing with the packet-reordering problem. With $b_v = 1$ the BnP give only one path from each v to the gateway because the flow is turned integer. Recall that the result W_f being integer does not necessarily mean the flow x is integer. In other words, the demand $b_v = 1$ does not implicate mono-routing if we permit x being fractional.

In our tests analysis, the path followed by the routers is close to a balanced spanning tree with the gateway being the root. The flows arriving at the gateway respect the min-max weighted cliques capacity. It means that distributed algorithms could solve optimally this problem considering a given optimal configuration of the critical region. Algorithms based in [17] may be an interesting option. There is an example in Figure 5 that shows the balanced spanning tree. We can also see, the complete routing with integer flows forming three cliques around of the gateway with the weights 18, 18 and 17.

VI. CONCLUSION AND PERSPECTIVES

In this article, we solve the Round Weighting Problem in order to satisfy a given demand subjected to the multiaccess interferences. We define a branch-and-price algorithm to solve optimally the non-relaxed version of this problem.

A column generation algorithm is used to avoid dealing with the whole exponential set of rounds. The branch-and-bound algorithm provides mono-routing. We make experiments with networks with different numbers of gateways.

Experimental results as well as theoretical insights let us conjecture that the bottleneck region analysis is enough to find the optimal solution. Therefore the RWP can be easily solved once we have covered this critical region. It is easy to assign time slots to the edges out of this region because, once we have covered the critical region, they will have several possibilities to get activated whitout change the total routing time W . Distributed algorithms could solve optimally this problem considering a given optimal configuration of the critical region.

The IRUP holds in our tests instances. Consequently the problem with integer flows is as easy as its relaxed version. We are getting further the proof of whether the integer round-up property holds for specific class of graphs.

ACKNOWLEDGMENTS

The authors wish to thank Jean-Claude Bermond and Joseph Yu for helpful discussions. C. Gomes is funded by CAPES, Brazil. This work has been partially funded by European project IST/FET AEOLUS, ANR-JC OSERA, and ARC CARMA.

REFERENCES

- [1] W. Wang, X. Li, O. Frieder, Y. Wang, and W. Song, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *International Conference on Mobile Computing and Networking (Mobicom)*. ACM Press, 2006, pp. 262–273.
- [2] R. Klasing, N. Morales, and S. Pérennes, "On the complexity of bandwidth allocation in radio networks," *Theoretical Computer Science*, 2008. [Online]. Available: <http://www.elsevier.com/locate/tcs>
- [3] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Oper. Res.*, vol. 46, no. 3, pp. 316–329, 1998.
- [4] S. Baum and J. L. E. Trotter, "Integer rounding for polymatroid and branching optimization problems," *SIAM J. Algebraic Discrete Methods*, vol. 2, no. 4, pp. 416–425, 1981.
- [5] C. Gomes, S. Pérennes, P. Reyes, and H. Rivano, "Bandwidth allocation in radio grid networks," in *10èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'08)*, Saint-Malo, May 2008.
- [6] J.-C. Bermond and J. Peters, "Efficient gathering in radio grids with interference," in *Septièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'05)*, Presqu'île de Giens, May 2005, pp. 103–106.
- [7] S. Mukherjee and H. Viswanathan, "Throughput-range tradeoff of wireless mesh backhaul networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 3, pp. 593–602, 2006.
- [8] C. Gomes, C. Molle, and P. Reyes, "Optimal design of wireless mesh networks," in *9èmes Journées Doctorales en Informatique et Réseaux (JDIR)*, 2008.
- [9] C. Gomes and G. Huiban, "Multiobjective analysis in wireless mesh networks," in *15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2007.
- [10] S. Kompella, J. Wieselthier, and A. Ephremides, "Multi-hop routing and scheduling in wireless networks subject to sinr constraints," in *46th IEEE Conference on Decision and Control*, December 2007, pp. 5690–5695.
- [11] J. Zhang, H. Wu, Q. Zhang, and B. Li, "Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks," in *International Conference on Broadband Networks (Broadnets)*, vol. 1. IEEE Press, October 2005, pp. 631–640.
- [12] P. Varbrand, D. Yuan, and P. Bjorklund, "Resource optimization of spatial tdma in ad hoc radio networks: a column generation approach," in *INFOCOM*, vol. 2. IEEE, March-April 2003, pp. 814–824.
- [13] C. Gomes, S. Pérennes, and H. Rivano, "Joint gathering and coloring problem in wireless mesh networks," INRIA, Tech. Rep. inria-00282200, May 2008. [Online]. Available: <http://hal.inria.fr/inria-00282200/fr/>
- [14] M. Garey and D. Johnson, *Computers and intractability*. New York: W. H. Freeman and Company, 1979.
- [15] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessaly, "SNDlib 1.0-Survivable Network Design Library," in *Proceedings of the Third International Network Optimization Conference (INOC 2007)*, Belgium, April 2007.
- [16] J. B. Orlin, "polynomial algorithm for integer programming covering problems satisfying the integer round-up property," *Journal Mathematical Programming*, vol. 22, no. 1, pp. 231–235, 1982.
- [17] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," in *Wireless Communications, IEEE Transactions on*, vol. 5, no. 8, Aug 2006, pp. 2185– 2193.