

Exact approach to reliability of wireless mesh networks with directional antennas

Mateusz Żotkiewicz · Michał Pióro

Published online: 30 August 2013

© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract The paper is devoted to modeling and optimization of reliable wireless mesh networks that employ directional antennas. We introduce two mixed-integer programming formulations that allow to simultaneously characterize routing patterns and transmission schedules. The first model allows for maximizing the minimal flow in a network. The second model involves reliability constraints and aims at minimizing the number of used directional antennas. In both cases locations of mesh routers are known. However, the number of installed radio interfaces and their directions are subject to optimization. We discuss a way of solving a cost minimization problem based on the introduced characterization, and present an extensive numerical study that illustrates the efficiency of the solution algorithm. We also provide an algorithm capable of verifying feasibility of obtained solutions. Moreover, in rare cases of failed verification, the algorithm provides additional constraints that should be added to the problem.

Keywords Wireless mesh networks · Directional antennas · Reliability · max-min fairness

This work was supported by the National Science Centre (Poland) under Grant: 2011/01/B/ST7/02967. M. Pióro was partly supported by ELLITT—Excellence Center in Information Technology (Sweden).

M. Żotkiewicz (✉) · M. Pióro
Institute of Telecommunications, Warsaw University of
Technology, Nowowiejska 15/19, 00-665 Warszawa, Poland
e-mail: mzotkiew@tele.pw.edu.pl

M. Pióro
e-mail: mpp@tele.pw.edu.pl

M. Pióro
Department of Electrical and Information Technology,
Lund University, Post Box 118, 221 00 Lund, Sweden

1 Introduction

Wireless mesh networks (WMN) offer common, affordable access to the Internet in metropolitan and residential areas (for popular surveys on WMN see [1, 9]). The core of a WMN consists of a set of fixed mesh nodes—routers and Internet gateways—interconnected by radio links that typically are based on the Wi-Fi IEEE 802.11-family standards. Other standards, such as Bluetooth IEEE 802.15.5, WiMAX IEEE 802.16a, and IEEE 802.20 can also support WMNs. In effect, aggregates of mesh clients (who are either fixed or mobile) connected to the routers obtain access to the Internet gateways either over direct links or over multi-hop mesh routes. WMN is a cost-efficient networking technology and provides a bandwidth in the range of hundreds Mbps. The WMN solution is competitive to the wired Internet access provided by cable network providers or offered by mobile operators. WMNs are decentralized, non-hierarchical networks, typically deployed by communities of users (see [4, 21]), and based on commonly available off-the-shelf wireless communication equipment (see [5, 18, 23]). The idea of WMN stems from the ad-hoc networking paradigm and, as such, fits very well the decentralized philosophy of the Internet.

Although WMNs are relatively cheap, efficient and fair resource allocation is not that simple to achieve in those networks. As discussed below, the issue is how to effectively divide the offered network capacity among the logical bandwidth assigned to the routes between the gateways and the routers. If not supported by some kind of tools (ideally simple, fast, and implemented in a distributed way in the routers) for effective management of transmission scheduling, channel assignment, transmission power adjustment, rate adaptation, and routing, a WMN can behave poorly, delivering significantly lower traffic throughput than can be

potentially achievable. For this reason, WMN optimization and traffic engineering form an important research area, situated in the mainstream of research in communication networking.

In WMN, it is not possible to carry point-to-point radio transmissions simultaneously on all links because of strong signal interferences at the receivers; typically, only a small proportion of the links can be active at the same time. Therefore, radio transmission scheduling algorithms are used to achieve a reasonable network traffic throughput. In fact, a WMN throughput can be significantly increased due to a properly designed transmission schedule implemented through an appropriate scheduling protocol (see for instance [24]). Implementing effective MAC scheduling algorithms, however, is not only difficult in practice, but also poses a considerable challenge in devising efficient algorithms for the traffic optimization problems.

In the paper we consider a problem of WMN throughput optimization through simultaneous design of the transmission schedule and routing (see for instance [2]). The throughput is a vector of bandwidths assigned to the downstreams from the Internet gateways to the mesh routers, while the routing is determined by the path (or the set of paths) by which each router is connected to the gateway(s). Traffic demands are assumed to be elastic and therefore they will consume any bandwidth they are assigned. In effect, the demands compete for link capacities, and the objective of the optimization becomes a fair assignment of bandwidth available to the routers. The transmission schedule is dealt with indirectly, by forbidding the interfering link transmissions to be held simultaneously. We assume that the transmission schedule specified (indirectly) in the result of the optimization can be effectively implemented by the scheduling protocol.

We significantly extend the considerations of [20, 27] by examining the reliability of wireless mesh networks when mesh routers are equipped with directional antennas. Using such equipment, it is possible to utilize available radio resources more efficiently by enabling more simultaneous transmissions in the network. On the other hand, the usage of directional antennas is more expensive (cost of equipment, cost of setting up the equipment). What is more, directional antennas decrease the robustness of the network by limiting its connectivity comparing to omnidirectional antennas. In the paper we verify the claim that using directional antennas is still beneficial because employing even a small number of additional directional antennas can significantly increase the reliability of a network.

The paper is organized as follows. Section 2 presents the related published work. In Sect. 3, the basic optimization models are introduced and discussed. Numerical results are given in Sect. 4. The paper is concluded in Sect. 5.

2 Related work

Wireless mesh networks have become an important research area, and consequently the number of positions in the literature that deal with WMN is remarkable. Some of the references treating WMNs in general can be found in the introduction. In this section we concentrate on two aspects of WMN, namely: reliability and directional antennas.

In the context of reliability, WMNs were usually considered as equipped with omnidirectional antennas, and the design tasks were usually reduced to finding a two-connected topology [7]. In [6], the authors analyze the impact of reliability on the cost of a network. An interesting position is [12] where the authors consider failures of optical units in the context of Hybrid Wireless-Optical Broadband-Access Network (WOBAN). In such a network a single failure of an optical unit can lead to a multiple failure from the point of view of the wireless component of the network.

In the context of directional antennas, the most interesting results are those dealing with theoretical increase in the capacity of a WMN when equipped with directional antennas. Those theoretical bounds were studied in different settings for instance in [25, 26]. In [13] authors also considered the possibility of multi-channel assignment that can significantly increase network capacity, while in [17] it is shown how directional antennas can influence multicast communication in wireless networks.

From the point of view of this paper the most important work is [11] describing a planning method for an urban WMN. The paper takes into account both reliability and possibility of using directional antennas. However, the authors assume that the interferences are negligible and thus are omitted in the considerations. This issue is directly considered in our investigations.

3 Optimization model

3.1 Notation

In this section we summarize the notation used in the paper. The considered network is composed of a set of nodes (gateways and mesh routers) connected with directed links (arcs). The network graph is bi-directed, i.e., if an arc between two nodes exists in one direction then there is also a symmetric arc in the opposite direction. This setting assumes the use of two directional antennas located at the two end nodes common to the two arcs. The two antennas realize the transmission in both directions, but only in one direction at a time. Throughout the paper we assume that each gateway is connected to the Internet using a fiber, thus it can deliver as much flow as needed.

Indices

- v, w nodes (mesh routers and gateways)
- e undirected link (link)
- a directed link (arc)
- s failure state
- 0 nominal state (no failures)
- i compatible set

Input

- \mathcal{R} list of mesh routers, $\mathcal{R} \subset \mathcal{V}$
- \mathcal{G} list of gateways, $\mathcal{G} \subset \mathcal{V}$
- \mathcal{V} list of all nodes, $\mathcal{V} = \mathcal{R} \cup \mathcal{G}, \mathcal{R} \cap \mathcal{G} = \emptyset$
- \mathcal{E} list of (undirected) links, $\mathcal{E} \subseteq \mathcal{V}^{|2|}$, where $\mathcal{V}^{|2|}$ denotes the set all two element subsets of \mathcal{V}
- \mathcal{A} list of (directed) arcs, $\mathcal{A} \subseteq \mathcal{V}^2 \setminus \{(v, v) : v \in \mathcal{V}\}$
- \mathcal{Q} conflict clique, $\mathcal{Q} \subset \mathcal{A}$
- \mathcal{C} set of all maximal conflict cliques, $\mathcal{Q} \in \mathcal{C}$
- \mathcal{S} set of failure states $\mathcal{S} = \mathcal{E} \cup \{0\}$; 0 denotes the nominal state with no failures, e denotes a failure of link $e \in \mathcal{E}$
- \mathcal{I} set of compatible sets, a compatible set is a set of links that can transmit simultaneously
- c_a capacity of arc a
- $e(a)$ link e that realizes arc a , i.e., if $a = (v, w)$ then $e(a) = \{v, w\}$; it is assumed that $a \in \mathcal{A} \Rightarrow e(a) \in \mathcal{E}$
- $\delta^+(v)$ list of arcs leaving node v
- $\delta^-(v)$ list of arcs entering node v
- P_a^v power received by node v when arc a is transmitting
- N noise
- δ_a SINR threshold at arc a (in the paper we assume that MCS used for arc a is set to the fastest MCS that will work on this arc when interferences are not present)
- M_a^s a big number fixed for arc a and failure state s
- α constant from interval $[0, 1]$ indicating the expected service quality after a failure; if $\alpha = 1$ then a failure should be unnoticeable by users, if $\alpha = 0$ then reliability is not concerned
- f^* maximal minimum flow
- f_a^* flow (load) on arc a
- f_a^{s*} flow (load) on arc a in failure state s
- x_e^* equal to 1 if at least one pair of antennas realizing link e is installed; 0 otherwise
- y_e^* equal to 1 if a backup (second) pair of antennas realizing link e is installed; 0 otherwise

Variables

- f minimum flow
- f_a flow (load) on arc a
- f^s minimum flow in failure state s
- f_a^s flow (load) on arc a in failure state s
- x_e binary variable equal to 1 if at least one pair of antennas realizing link e is installed; 0 otherwise
- y_e binary variable equal to 1 if a backup (second) pair of antennas realizing link e is installed; 0 otherwise ($y_e \leq x_e$)
- z_i fraction of available time used by compatible set i

Table 1 IEEE 802.11a MCS, FER $\leq 1\%$, 1500 Byte payload

MCS	Bit rate	SNR threshold δ
BPSK 1/2	6 Mbps	3.5 dB
BPSK 3/4	9 Mbps	6.5 dB
QPSK 1/2	12 Mbps	6.6 dB
QPSK 3/4	18 Mbps	9.5 dB
16-QAM 1/2	24 Mbps	12.8 dB
16-QAM 3/4	36 Mbps	16.2 dB
64-QAM 2/3	48 Mbps	20.3 dB
64-QAM 3/4	54 Mbps	22.1 dB

3.2 Basics

In WMN networks the notion of a link is vague, as any node (a mesh router or a gateway) can transmit to any other node in its range. In the paper we assume that a link potentially exists between every pair of mesh routers that can communicate with each other. In radio networks two nodes v and w can communicate, if the power of the signal received from node v at node w in comparison to the noise at node w is greater than an acceptable *signal-to-noise* (SNR) threshold for the applied modulation and coding scheme (MCS). We consider eight different MCSs. They are characterized by different bit rates and different SNR thresholds presented in Table 1 (reproduced from [22]). We assume that each link uses the fastest MCS that does not violate an SNR constraint, i.e., a signal received at a sink node of an arc is greater than the noise by a factor specified by an SNR threshold. In order to calculate the volume of the received signal we use a generic path model presented in [14] with a path loss exponent of 4. Other details, like transmitting powers of nodes and distances between nodes, can be found in the descriptions of Sect. 4.

In WMN networks radio resources are shared between all nodes. Thus, typically all nodes cannot transmit simultaneously but, of course, simultaneous transmissions by subsets of nodes are still possible (and desired). In the case of simultaneous transmissions, however, not only the noise can disrupt a transmission but also interferences created by other nodes. Therefore, in the context of WMN we use the notion of the *signal to interference plus noise ratio* (SINR) instead of SNR. SINR is understood as a proportion of a received power to a sum of the noise and received interferences. In this paper we assume a common (simplified) interference model that utilizes the notion of a *conflict graph* (see for instance [15]).

A conflict graph represents pairs of arcs that cannot transmit simultaneously because of interferences. Each arc in a WMN network is represented by a vertex in a conflict graph, and an edge in a conflict graph represents a pair of arcs that are forbidden to transmit at the same time. Consider four

mesh routers v , w , v' , and w' , and two arcs: a (from node v to node w) and a' (from v' to w'). Those two arcs correspond to two vertices denoted by A and A' , respectively, in the conflict graph. Finally, an edge between A and A' exists only if at least one of the following constraints is not satisfied:

$$\frac{P_a^w}{N + P_a^w} \geq \delta_a \quad (1a)$$

$$\frac{P_{a'}^{w'}}{N + P_{a'}^{w'}} \geq \delta_{a'}, \quad (1b)$$

where constraint (1a) assures that a transmission on arc a' does not jam a transmission on arc a , while (1b) assures the opposite—a transmission on arc a does not jam a transmission on arc a' .

Following [15], we find all maximal cliques in a conflict graph and use them in our formulations (2a)–(2d) and (3a)–(3j). A clique of a conflict graph represents a set of arcs out of which only one can transmit at a time. Notice that the simplified interference model does not take into account a possibility that a number (two or more) of mesh routers can together create interferences that will make communication on another arc impossible. Such a situation is considered in the full interference model considered for instance in [10] or [20], where interferences created by different mesh routers are cumulated. However, due to the fact that in our paper we assume WMNs with directional antennas (which limits the interferences) we have decided to use the simplified model, as it makes the optimization process more efficient.

We also assume that a scheduling algorithm is run above the MAC level, and each node is transmitting only in time slots that are allocated to it. Sizes of the time slots are not equal and are subject to optimization. Our task is to find an optimal routing for each demand. The routing must be designed so that the minimum flow in a network is maximized. We consider only the downlink traffic (from gateways to mesh routers), as it is usually much greater than the uplink traffic in the considered networks. We note that the uplink traffic could be added to the considerations through straightforward extensions of the introduced models.

3.3 Directional antennas

Considerations presented in the previous section apply regardless of the type of antennas used in a network. In this section we discuss an influence of using directional antennas on modeling. In fact, for any bi-directional link we need two dedicated directional antennas, one in each of the two end nodes of this particular link. In other words, the number of available arcs $|\mathcal{A}|$ is equal to the number of antennas, as a pair of antennas creates two oppositely directed arcs between them.

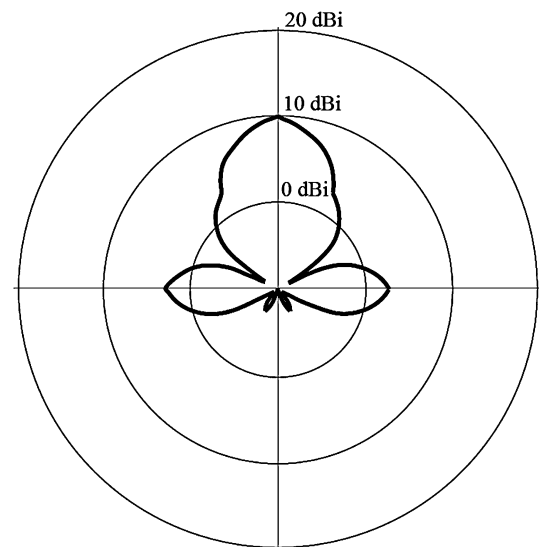


Fig. 1 Radiation pattern

An impact of directional antennas on our considerations is twofold. First of all, the power received by mesh routers (and gateways, for that matter), when other routers are transmitting, depends not only on a location of a transmitting router but also on a location of a router that the transmission is aimed to. Secondly, a link between two nodes (in fact, two oppositely directed arcs) exists only if a pair of directional antennas is dedicated to it. Notice that in case of omnidirectional antennas a link exists between each router and all other routers in its range.

The first issue is addressed in the following way. We assume that locations of links are given (they are not subject to optimization), and all the potential antennas use the same transmission power and the radiation pattern shown in Fig. 1 that represents a typical radiation pattern of a directional antenna [3]. Therefore, the values of P_a^w can be computed beforehand, and thus treated as constants in (1a)–(1b).

The second issue requires variables x_e that are responsible for limiting the number of installed antennas. Notice that in case of omnidirectional antennas the number of used links does not influence the number of antennas that have to be installed, as there is just one antenna at every node.

3.4 Reliability

In our model we consider all single failures of network interfaces, i.e., antennas. We say that a network is *reliable*, if in case of any failure the minimum flow f among total flows assigned to mesh routers is greater or equal to αf^* . The quantity f^* is equal to the maximal minimum flow when failures are not present, and α is a fixed given number from interval $[0, 1]$. Notice that in the scenario with omnidirectional antennas reliability implies that the number of additional interfaces has to be equal at least to $|\mathcal{R}| = |\mathcal{V}| - |\mathcal{G}|$

for any $\alpha > 0$. In case of the scenario with directional antennas, the number of additional interfaces is much smaller as shown by numerical results in Sect. 4.

3.5 Problem

We are now able to present mathematical models that encompass the specific features of WMN, the impact of directional antennas, and the notion of reliability. In our studies we use two tightly connected formulations. The first (simpler) is called the flow maximization model. It deals only with the nominal state and does not consider reliability. The second (more complicated) is called the resource minimization model. It uses the optimal objective value of the first model and takes into account reliability.

Flow maximization model

$$\max f \tag{2a}$$

$$\sum_{a \in \delta^-(v)} f_a - \sum_{a \in \delta^+(v)} f_a = f, \quad v \in \mathcal{R} \tag{2b}$$

$$\sum_{a \in \mathcal{Q}} \frac{f_a}{c_a} \leq 1, \quad \mathcal{Q} \in \mathcal{C} \tag{2c}$$

$$f_a \geq 0, \quad a \in \mathcal{A} \tag{2d}$$

Problem (2a)–(2d) finds the maximal minimum flow that can be supported by the network. Constraint (2b) expresses the flow conservation law, where the right hand side (i.e., variable f) is the total bandwidth that is designated to the router $v \in \mathcal{R}$ associated with this particular constraint. Note that this is a common value f that is maximized through objective (2a). Recall that we assume that each gateway can deliver as much flow as needed and that gateways do not transit traffic. Constraint (2c) assures that a capacity of any clique in a conflict graph is not exceeded. More precisely, since $\frac{f_a}{c_a}$ is in fact a fraction of time of network operation that is required to send the flow f_a (expressed in Mbps) over a link of a constant capacity c_a (also expressed in Mbps), constraint (2c) makes it possible, for each clique, to realize the clique’s link loads one by one. Observe that this constraint also indirectly assures that for no link its capacity is exceeded by its load, i.e., $f_a \leq c_a$, for all $a \in \mathcal{A}$. Constraint (2d) assures nonnegativity of variables.

The value of the optimal objective (2a) is denoted by f^* and is used as a constant in the next model.

Resource minimization model

$$\min \sum_{e \in \mathcal{E}} (x_e + y_e) \tag{3a}$$

$$\sum_{a \in \delta^-(v)} f_a^s - \sum_{a \in \delta^+(v)} f_a^s = f^s, \quad v \in \mathcal{R}, s \in \mathcal{S} \tag{3b}$$

$$\sum_{a \in \mathcal{Q}} \frac{f_a^s}{c_a} \leq 1, \quad \mathcal{Q} \in \mathcal{C}, s \in \mathcal{S} \tag{3c}$$

$$f_a^0 \leq M_a^0 \cdot x_{e(a)}, \quad a \in \mathcal{A} \tag{3d}$$

$$f_a^s \leq M_a^s \cdot \begin{cases} x_{e(a)} & \text{if } e(a) \neq s \\ y_{e(a)} & \text{if } e(a) = s, \end{cases} \quad a \in \mathcal{A}, s \in \mathcal{E} \tag{3e}$$

$$f^s \geq \alpha \cdot f^*, \quad s \in \mathcal{E} \tag{3f}$$

$$f^0 = f^* \tag{3g}$$

$$y_e \leq x_e, \quad e \in \mathcal{E} \tag{3h}$$

$$x_e, y_e \in \{0, 1\}, \quad a \in \mathcal{E} \tag{3i}$$

$$f_a^s \geq 0, \quad a \in \mathcal{A}, s \in \mathcal{S} \tag{3j}$$

Formulation (3a)–(3j) minimizes the number $\sum_{e \in \mathcal{E}} 2 \cdot (x_e + y_e)$ of installed antennas required to fulfil the reliability requirements for a network flow maximized through model (2a)–(2d). Constraint (3b) expresses the flow conservation law for each state in \mathcal{S} (i.e., for all single link failures and the nominal state). The right hand side is a total flow in state s that is designated to router $v \in \mathcal{R}$ associated with a particular constraint. A total flow designated to routers can differ between failure states. Constraint (3c) assures that a capacity of any clique in a conflict graph is not exceeded in any failure state. Constraints (3d) and (3e) set the values of x_e and y_e for each link. Recall that a pair of antennas creates two oppositely directed arcs. Note that for each state $s \in \mathcal{S}$ in an optimal solution each link can be used in only one direction, as otherwise we would have a loop that could be removed from the solution. However, the direction of a flow on a link can certainly change between failure states. Constraint (3g) ensures that the minimum flow in the nominal state f^0 is equal to maximal minimum flow f^* computed by (2a)–(2d), while constraint (3f) assures that a network is reliable, i.e., forces the minimum flow in all failure states to be at least $\alpha \cdot f^*$. Constraint (3h) assures consistency of variables x and y , while constraints (3i) and (3j) are binarity and nonnegativity constraints, respectively.

The problem we consider is to minimize the number of used directional radio interfaces that allow for providing a reliable WMN with fair share of resources (maximizing the minimum flow). We notice that the maximal minimum flow can always be made the same for any $\alpha \in [0, 1]$ by putting $y_e = x_e, e \in \mathcal{E}$, i.e., by providing 1:1 backup for each link. Such a modification is feasible and uses the same flows in all states. Therefore, for a given topology we first solve (2a)–(2d) and treat its objective value f^* as a constant while solving (3a)–(3j). Obviously, solving (3a)–(3j) gives us the solution—the number of directional antennas needed to provide a WMN maximizing the minimum flow and characterized by a given reliability constant α .

It has to be emphasized that neither (2a)–(2d) nor (3a)–(3j) guarantees that a feasible scheduling exists. However, in great majority of practical scenarios constraints (2c) and (3c) assure the existence of a feasible scheduling [15]. In order to assure 100 % certainty, models based on [20, 27] can be used to verify a solution.

3.6 Validation

First of all, it has to be verified if a solution provided by (2a)–(2d) is implementable, i.e., there exists a scheduling that provides it. In order to explain the validation process, the notion of compatible sets has to be elaborated. A compatible set is a set of links that can simultaneously transmit without jamming each other. In other words, if all links of a compatible set are transmitting simultaneously, SINR constraints are satisfied for all of them, i.e., constraints (1a)–(1b) are satisfied for all pair of arcs in a compatible set.

Using the notion of compatible sets the verification can be easily done by solving the following formulation.

Flow validation model

$$\sum_{i \in \mathcal{I}} z_i = 1 \tag{4a}$$

$$\sum_{i \in \mathcal{I}: i \ni a} c_a z_i \geq f_a^*, \quad a \in \mathcal{A} \tag{4b}$$

$$z_i \geq 0, \quad i \in \mathcal{I} \tag{4c}$$

Formulation (4a)–(4c) does not have an objective function, as its sole goal is to verify if a solution provided by (2a)–(2d) can be implemented. First, constraint (4a) divides available time resources between compatible sets. It assures that two links which, due to interferences, cannot transmit simultaneously will not be active at the same time. Second, constraint (4b) assures that selected compatible sets provide enough capacity at links. Note that f_a^* in (4b) is a constant equal to an optimal value of variable f_a from (2a)–(2d), thus the only variable in the formulation are z_i (notice that c_a are constant and depend on a topology of a network). The problem is that the number of those variables grows exponentially with the size of a network. Fortunately the problem can still be solved efficiently using column generation [20]. It is worth to emphasize that the model does not optimize routing and its running times are significantly smaller, thus it can be successfully used for verification.

If model (4a)–(4c) has a feasible solution then the obtained maximal minimum flow can be achieved, and a solution to (3a)–(3j) should be verified now. However, if (4a)–(4c) does not have a feasible solution then the following formulation merging (2a)–(2d) and (4a)–(4c) should be solved.

Extended flow maximization model

$$\max f \tag{5a}$$

$$\sum_{a \in \delta^-(v)} f_a - \sum_{a \in \delta^+(v)} f_a = f, \quad v \in \mathcal{R} \tag{5b}$$

$$\sum_{i \in \mathcal{I}} z_i = 1 \tag{5c}$$

$$\sum_{i \in \mathcal{I}: i \ni a} c_a z_i \geq f_a, \quad a \in \mathcal{A} \tag{5d}$$

$$z_i \geq 0, \quad i \in \mathcal{I} \tag{5e}$$

$$f_a \geq 0, \quad a \in \mathcal{A} \tag{5f}$$

Formulation (5a)–(5f) merges routing issues from (2a)–(2d) with scheduling issues of (4a)–(4c). It is used instead of (2a)–(2d), when the validation performed by (4a)–(4c) fails.

If either validation test (4a)–(4c) is successful or extended formulation (5a)–(5f) is solved and also a solution to (3a)–(3j) is obtained then the second phase of the validation process, which confirms existence of a feasible scheduling for an obtained solution to (3a)–(3j), should take place. This process is divided into steps. First of all, it has to be verified, if there exists a feasible solution to (3a)–(3j) that uses the same set of directional links but divides traffic in a different (feasible from the (6a)–(6c) viewpoint) way. It can be done solving the following model.

Resource validation model

$$\sum_{i \in \mathcal{I}} z_i^s = 1, \quad s \in \mathcal{S} \tag{6a}$$

$$\sum_{i \in \mathcal{I}: i \ni a} c_a z_i^s \geq f_a^{s*}, \quad s \in \mathcal{S}, a \in \mathcal{A} \tag{6b}$$

$$z_i^s \geq 0, \quad s \in \mathcal{S}, i \in \mathcal{I} \tag{6c}$$

Formulation (6a)–(6c) is similar to (4a)–(4c). The only difference is that in the former all failure states has to be taken into account. However, notice that it can be decomposed on s indices, thus from the mathematical point of view they are the same and all the technics used to solve (4a)–(4c) can be also applied here.

If model (6a)–(6c) has a feasible solution then the considered problem is solved and a solution to (3a)–(3j) can be implemented. However, if (6a)–(6c) does not have a feasible solution then appropriate steps have to be taken in order to solve the problem. First of all, it has to be verified, if there exists a feasible solution to (3a)–(3j) that uses the same set of directional links but divides traffic in a different (feasible from the (6a)–(6c) viewpoint) way. It can be done solving the following model.

Routing model

$$\sum_{a \in \delta^-(v)} f_a^s - \sum_{a \in \delta^+(v)} f_a^s = f^s, \quad v \in \mathcal{R}, s \in \mathcal{S} \tag{7a}$$

$$\sum_{i \in \mathcal{I}} z_i^s = 1, \quad s \in \mathcal{S} \tag{7b}$$

$$\sum_{i \in \mathcal{I}: i \ni a} c_a z_i^s \geq f_a^s, \quad s \in \mathcal{S}, a \in \mathcal{A} \tag{7c}$$

$$f_a^0 \leq M_a^0 \cdot x_{e(a)}^*, \quad a \in \mathcal{A} \tag{7d}$$

$$f_a^s \leq M_a^s \cdot \begin{cases} x_{e(a)}^* & \text{if } e(a) \neq s \\ y_{e(a)}^* & \text{if } e(a) = s, \end{cases} \quad a \in \mathcal{A}, s \in \mathcal{E} \tag{7e}$$

$$f^s \geq \alpha \cdot f^*, \quad s \in \mathcal{E} \tag{7f}$$

$$f^0 = f^* \tag{7g}$$

$$f_a^s \geq 0, \quad a \in \mathcal{A}, s \in \mathcal{S} \tag{7h}$$

$$z_i^s \geq 0, \quad s \in \mathcal{S}, i \in \mathcal{I} \tag{7i}$$

Formulation (7a)–(7i) merges (3a)–(3j) and (6a)–(6c), but it is solved for only one configuration of antennas (replaces variables x_e and y_e by constants x_e^* and y_e^*). This operations make the problem linear. However, the number of variables in the formulation again grows exponentially with the size of the problem. Luckily, also in this case the column generation can be successfully used. If we decided to keep integer variables x_e and y_e in the formulation then the problem would have to be solved using more sophisticated technics like branch & price. Notice that, although formulation (7a)–(7i) is not a validation model, it does not have an objective function. The reason is that all the routing solutions it can provide will be equivalent in term of costs seen from the point of view of (3a)–(3j), because they all use the same set of directional antennas.

If formulation (7a)–(7i) is solvable then its solution is a feasible and implementable solution to the considered problem. However, when the formulation does not have a solution then (3a)–(3j) has to be resolved with additional constraints that assure a different solution. The goal can be obtained in the following way. Consider sets $\mathcal{X} = \{x_e : x_e^* = 1\}$ and $\mathcal{Y} = \{y_e : y_e^* = 1\}$ containing variables that were equal 1 in an optimal solution to (3a)–(3j). Then the constraints that has to be added to (3a)–(3j) are as follows.

$$\sum_{x_e \in \mathcal{X}} x_e + \sum_{y_e \in \mathcal{Y}} y_e \leq |\mathcal{X}| + |\mathcal{Y}| - 1 \tag{8a}$$

$$\sum_{e \in \mathcal{E}} (x_e + y_e) = |\mathcal{X}| + |\mathcal{Y}| \tag{8b}$$

The former assures that a set of installed antennas will be different from the previously obtained set. The latter imposes the same value of the objective function. The obtained

solution to the augmented (3a)–(3j) has to be verified using the above procedure. If (3a)–(3j) is unsolvable because of added inequalities then all the above constraints should be erased and the following constraint should be added instead.

$$\sum_{e \in \mathcal{E}} (x_e + y_e) > |\mathcal{X}| + |\mathcal{Y}| \tag{9}$$

Summarizing, the complete algorithm is as follows.

Algorithm 1 Network design problem

```

solve (2a)–(2d)
solve (4a)–(4c)
if (4a)–(4c) unsolvable then
    solve (5a)–(5f)
end if
while true do
    solve (3a)–(3j)
    if (3a)–(3j) unsolvable then
        erase (8a)–(8b) constraints
        add (9) constraint
        solve (3a)–(3j)
    end if
    solve (6a)–(6c)
    if (6a)–(6c) solvable then
        return (3a)–(3j) solution
    end if
    solve (7a)–(7i)
    if (7a)–(7i) solvable then
        return (7a)–(7i) solution
    end if
    add (8a)–(8b) constraints
end while
    
```

Algorithm 1 seems to be complex. However, in great majority of practical cases is reduces to solving (2a)–(2d), validating it using (4a)–(4c), solving (3a)–(3j), and validating it using (6a)–(6c).

3.7 Solution techniques

It has to be emphasized that both (2a)–(2d) and (3a)–(3j) are non-compact formulations, as the number of possible maximal cliques of a conflict graph is exponential in the size of a network, thus the number of constraints (2c) and (3c) also grows exponentially. A common strategy for solving such problems involves constraint generation techniques (adding violated constraints at each node of a B&C tree). However, in our case, first, the constraint generation problem is to find a maximum weighted clique, thus it is \mathcal{NP} -hard, second, the complexity of generating all maximal cliques is comparable to the complexity of finding the maximum clique. Finally,

our preliminary research has shown that the number of generated constraints is usually roughly equal to 30 % of the number of all possible constraints. Therefore, in our experiments we decided to generate all maximal cliques beforehand, and add them to the MILP model. Our implementation of an algorithm presented in [8] that lists all maximal cliques of a graph runs for less than one second for all test cases of Sect. 4.

Notice that both validating formulations, i.e., (4a)–(4c) and (6a)–(6c), are also non-compact. However, in this case the number of variables, not constraints, is an issue. The problem can be solved using column generation technics. However, in test cases considered in this paper it was enough to generate all feasible, maximum compatible sets, and use all of them in (4a)–(4c) and (6a)–(6c), as in the worst test case only 1856 compatible sets existed.

Formulation (3a)–(3j) uses M_a^s constants—fixed numbers that must be greater than or equal to the optimal flows $f_a^s, a \in \mathcal{A}, s \in \mathcal{S}$. It is well known (and will be illustrated in Sect. 4) that for efficiency of the branch-and-bound algorithms it is important that M_a^s are as small as possible, because they directly affect the quality of lower bounds provided by linear relaxations of (3a)–(3j). Certainly, $M_a^s = c_a$ can always be used. Still, much better values of M_a^s can be computed through a linear program using the following notation.

Input

A the fixed arc, $A \in \mathcal{A}$, v_A mesh router that ends A

Variables

- f_a flow on arc a that has not used arc A
- g_a flow on arc a that has used arc A before reaching a
- F_v flow destined to router v that has not used arc A
- G_v flow destined to router v that has used arc A before reaching router v .

The problem below is solved for each $A \in \mathcal{A}$.

Arc-usage maximization model

$$\max f_A \tag{10a}$$

$$\sum_{a \in \delta^-(v)} f_a = F_v + \sum_{a \in \delta^+(v)} f_a, \quad v \in \mathcal{R} \setminus \{v_A\} \tag{10b}$$

$$\sum_{a \in \delta^-(v)} g_a = G_v + \sum_{a \in \delta^+(v)} g_a, \quad v \in \mathcal{R} \setminus \{v_A\} \tag{10c}$$

$$\sum_{a \in \delta^-(v_A)} f_a - f_A = F_{v_A} + \sum_{a \in \delta^+(v_A)} f_a \tag{10d}$$

$$\sum_{a \in \delta^-(v_A)} g_a + f_A = G_{v_A} + \sum_{a \in \delta^+(v_A)} g_a \tag{10e}$$

$$\sum_{a \in \delta^-(v_A)} g_a = 0 \quad (\text{alternatively } g_A = 0) \tag{10f}$$

$$F_v + G_v = f^*, \quad v \in \mathcal{R} \tag{10g}$$

$$\sum_{a \in \mathcal{Q}} \frac{f_a + g_a}{c_a} \leq 1, \quad \mathcal{Q} \in \mathcal{C} \tag{10h}$$

$$f_a, g_a \geq 0, \quad a \in \mathcal{A} \tag{10i}$$

$$F_v, G_v \geq 0, \quad v \in \mathcal{R} \tag{10j}$$

Then we can set M_A^0 to $\max f_A$. In order to get M_A^s for the failure states we have to solve (10a)–(10j) with (10g) replaced by $F_v + G_v = \alpha f^*$.

Formulation (10a)–(10j) maximizes a flow that can be transmitted on arc A . One may think that the same result can be obtained by solving (2a)–(2d) with an objective $\max f_A$. However, in such a case, it is impossible to avoid loops on variables f_a that increase the value of f_A . In case of (10a)–(10j) such loops cannot be generated as flows that have used arc A (resp. have not used arc A) use different sets of variables, namely g_a (resp. f_a).

4 Numerical results

We tested numerically the models presented in Sect. 3 using 15 different network topologies, generated randomly in the following way. First, nodes were randomly located inside a square. Then, a value of transmitting power (the same for all gateways and routers) was set to the minimum value that keeps the network connected (we followed [14] and assumed the path loss exponent of 4). MCSs were set to the fastest among those that are able to operate with SNRs resulting from the selected transmission power and the locations of routers. The assumptions presented above may seem unpractical. However, their goal is to obtain test cases consisting of: set of nodes, conflict graph, and MCSs for each possible transmission. After obtaining test cases, the questionable assumptions, e.g., the same transmitting power for all gateways and routers, are not visible.

The models were solved using CPLEX 12.1 [16] on Intel Core 2 Quad CPU Q6600 2.40 GHz with 4.00 GB RAM running 64-bit Windows 7.

In Table 2 the test cases are presented together with the number of radio interfaces required. The first six columns of this table describe tested topologies, while the following six columns give results concerning the number of required antennas. The meaning of the first six columns is as follows.

- # identification number of a test case (the same numbers are used in Table 3)
- $|\mathcal{R}|$ number of mesh routers
- $|\mathcal{G}|$ number of gateways
- $|\mathcal{A}|$ total number of radio interfaces that can be installed
- CFL number of edges in a corresponding conflict graph
- $|\mathcal{C}|$ number of maximal cliques in a conflict graph

Table 2 Required number of antennas

Topology #	\mathcal{R}	\mathcal{G}	\mathcal{A}	CFL	\mathcal{C}	α					
						0.0	0.05	0.25 (LP)	0.25 (c_a)	0.5	1.0
1	8	2	24	176	30	18	24	24	24	24	32
2	8	2	34	390	32	24	26	26	26	28	46
3	8	2	54	1672	409	16	22	22	22	24	30
4	8	2	48	976	107	16	24	24	24	24	30
5	8	2	24	184	8	16	24	24	24	24	26
6	12	3	62	1220	76	28	38	38	38	38	50
7	12	3	84	2328	183	26	32	32	32	34	42-39.28
8	12	3	54	690	47	26	38	38	38	38	46
9	12	3	58	1046	161	26	32	32	32	32	38
10	12	3	52	716	76	24	34	34	34	36	42
11	17	3	96	2842	1369	34	44	68-42.74	44	68-44.00	68-47.04
12	17	3	118	3458	314	36	42	42-40.00	46-40.00	42	50-45.88
13	17	3	44	298	30	34	60	60	60	60	66
14	17	3	110	3386	268	44	58	58	58	58	88
15	17	3	128	3708	542	44	50	50	54-45.66	52-47.98	78

The following six columns contain the numbers of required antennas for different values of α given in the second row of the table. It is worth to notice that two columns among the six deal with the case $\alpha = 0.25$. The reason is that for this value of α we decided to evaluate the impact of the M_a^s generation strategy presented in Sect. 3.7. Results in the column marked as (LP) have been obtained using the LP formulation of Sect. 3.7 for generating M_a^s , while results in the column marked by (c_a) have been obtained by setting $M_a^s = c_a$. Note that in some cases the number of required antennas is given in the form a - b which means that a 2-hour time limit was reached for this test case and the best found upper bound is a , while the obtained lower bound is b .

The first conclusion that we draw from the results is that usually the minimum number of directional antennas that connects a network is not sufficient for maximizing the flow. The minimum number of antennas was enough to obtain the maximum flow only for test cases #: 3, 4, 5, 10, 11, and 13. The second conclusion is that in order to obtain even the basic reliability (by the basic reliability we understand $\alpha = 0.05$), a significant increase in the number of directional interfaces is required, ranging from 8 % in test case #2 to 76 % in test case # 13, with the mean value of 34 %. However, this additional investment usually results in much greater reliability than the basic one. When 100 % reliability is required, the increase in the number of antennas ranges from 46 % to 100 %, with the mean value of 80 %.

In Table 3 the running times are presented. The first column of this table contains the identification numbers of the test cases (they correspond to the values from Table 2), while the following columns contain the running times for differ-

ent values of α . Again, a number of different columns correspond to $\alpha = 0.25$. Three of them deal with the model enhanced by the M_a^s generation strategy presented in Sect. 3.7, and are marked by (LP). Another two columns, marked by (c_a), have been obtained by setting $M_a^s = c_a$. Those five columns have the following meaning: *prepare*—is listed only for (LP) and contains time needed to compute all values of M_a^s , note that for (c_a) we just take $M_a^s = c_a$; *solve*—contains the time needed by CPLEX to solve the problem to optimality; *prove*—contains time needed by CPLEX to prove optimality of a solution, when the solution is given beforehand. This time was obtained by running CPLEX with an upper bound set to the value of an optimal solution found while filing column *prove* minus one (as only integral objective values are feasible for this problem). Notice that for the test case #12 we were not able to find an optimal solution for $\alpha = 0.25$. However, we have found an optimal solution for $\alpha = 0.05$ whose objective value is equal to the upper bound for $\alpha = 0.25$ (LP). Therefore, we can treat this value as optimal for $\alpha = 0.25$ and fill the column *prove* for it. For all test cases a 2-hour time limit was set. The test cases that exhaust the time limit are marked with an asterisk. The column marked by *validate* presents time needed to validate a solution. For all test cases the validating algorithm reduced to its simplest form, i.e., solving (2a)–(2d), validating it using (4a)–(4c), solving (3a)–(3j), and validating it using (6a)–(6c).

First notice that the computational difficulty of the problem increases with α . The reason is that for bigger α the importance of (3c) increases, and it becomes more difficult to find a solution that satisfies this constraint. Also observe

Table 3 Running times

#	$\alpha = 0.0$ $\alpha = 0.05$		(LP)			(c_a)			$\alpha = 0.5$	$\alpha = 1.0$
			Prepare $\alpha = 0.25$	Solve	Prove	Solve	Prove	Validate		
1	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
2	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
3	<1 s	7 s	5 s	17 s	<1 s	1 m 30 s	<1 s	<1 s	15 s	6 m 57 s
4	<1 s	1 s	<1 s	2 s	2 s	3 s	3 s	<1 s	4 s	17 s
5	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
6	<1 s	<1 s	2 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	4 s
7	<1 s	15 s	7 s	48 s	21 s	13 m 10 s	5 m 26 s	<1 s	23 m 34 s	2 h*
8	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
9	<1 s	1 s	3 s	3 s	<1 s	1 s	1 s	<1 s	9 s	8 m 48 s
10	<1 s	<1 s	2 s	<1 s	<1 s	1 s	<1 s	<1 s	3 s	28 s
11	<1 s	30 m 52 s	2 m 44 s	2 h*	13 s	27 m 1 s	16 m 2 s	<1 s	2 h*	2 h*
12	<1 s	14 m 11 s	44 s	2 h*	2 h*	2 h*	2 h*	-	1 h 15 m	2 h*
13	<1 s	<1 s	2 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
14	<1 s	3 s	34 s	1 m 7 s	30 s	47 m 3 s	18 m 2 s	<1 s	29 s	19 s
15	<1 s	22 s	1 m 32 s	1 h 38 m	21 m 10 s	2 h*	2 h*	-	2 h*	26 m 56 s

the impact of M_a^s on the running times. In most cases the running time decreases significantly when M_a^s are precomputed using the LP method of Sect. 3.7. The last conclusion is the importance of finding an upper bound of good quality. It is often the case in network optimization that finding an optimal solution is relatively simple. However, proving its optimality becomes a challenge [19]. Our results show that in this problem finding an optimal solution is also very important, as having it from the beginning significantly reduces computational time.

5 Conclusion

In the paper we have introduced two mixed-integer programming formulations that allow to simultaneously characterize routing patterns and transmission schedules, one for the case where the max-min flow is of interest, and the other (involving reliability constraints) that minimizes the number of used antennas. We have discussed a way of solving the considered problems, and presented an extensive numerical study that illustrates the efficiency of the solution algorithm. We also provided an algorithm capable of verifying feasibility of obtained solutions. Moreover, in rare cases of failed verification, the algorithm provides additional constraints that should be added to the problem.

Further work in this field should analyze the impact of using antennas of different radiation patterns and/or different transmitting power. A problem where MCSs are not given in advance (are subject to optimization) is also challenging.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Akyildiz, I. F., & Wang, X. (2005). A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9), 23–30.
- Avallone, S., D’Elia, F. P., & Ventre, G. (2012). A new channel, power and rate assignment algorithm for multi-radio wireless mesh networks. *Telecommunication Systems*, 51(1), 73–80.
- Balanis, C. A. (2005). *Antenna theory: analysis and design*. New York: Wiley-Interscience.
- Bay area wireless users group. See <http://www.bawug.com>.
- Belair networks. See <http://www.belairnetworks.com>.
- Beljadid, A., Hafid, A. S., & Gendreau, M. (2008). Design of wireless mesh networks: expansion and reliability studies. In *Proceedings of IEEE Globecom*, New Orleans, LA, USA.
- Benyamina, D., Hafid, A., & Gendreau, M. (2009). On the design of bi-connected wireless mesh network infrastructure with QoS constraints. In *Proceedings of IEEE Globecom*, Honolulu, HI, USA.
- Bron, C., & Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of ACM*, 16, 575–577.
- Bruno, R., Conti, M., & Gregori, E. (2005). Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3), 123–131.
- Capone, A., Carello, G., Filippini, I., Gualandi, S., & Malucelli, F. (2010). Routing, scheduling and channel assignment in wmn networks: optimization, models and algorithms. *Ad Hoc Networks*, 8, 545–563.
- Chen, C., & Chekuri, C. (2007). *Urban wireless mesh network planning: the case of directional antennas*. Technical Re-

- port UIUCDCS-R-2007-2874, University of Illinois at Urbana-Champaign.
12. Correia, N., Coimbra, J., & Schütz, G. (2009). Fault-tolerance planning in multiradio hybrid wireless-optical broadband access networks. *Journal of Optical Communications and Networking*, 1(7), 645–654.
 13. Dai, H., Ng, K., Wong, R. C., & Wu, M. (2008). On the capacity of multi-channel wireless networks using directional antennas. In *Proceedings of IEEE Infocom*, Phoenix, AZ, USA.
 14. Goldsmith, A. (2005). *Wireless communications*. Cambridge: Cambridge University Press.
 15. Huang, X. L., & Bensaou, B. (2001). On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation. In *Proceedings of ACM MobiHoc*, Long Beach, CA, USA (pp. 221–231).
 16. ILOG CPLEX 12.1 User's Manual. ILOG. See ftp://ftp.software.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf.
 17. Jaikaeo, C., & Shen, C.-C. (2005). Multicast communication in ad hoc networks with directional antennas. *Telecommunication Systems*, 28(3–4), 297–316.
 18. Nortel. See <http://www.nortel.com>.
 19. Pióro, M., & Medhi, D. (2004). *Routing, flow, and capacity design in communication and computer networks*. San Mateo: Morgan-Kaufmann.
 20. Pióro, M., Żotkiewicz, M., Staehle, B., Staehle, D., & Yuan, D. (2011). On max-min fair flow optimization in wireless mesh networks. *Ad Hoc Networks*. doi:10.1016/j.adhoc.2011.05.003.
 21. Seattle wireless. See <http://www.seattlewireless.net>.
 22. Staehle, B., Staehle, D., & Pries, R. (2009). Effects of link rate assignment on the max-min fair throughput of wireless mesh networks. In *Proceedings of ITC*, Paris, France.
 23. Tropos networks. See <http://www.tropos.com>.
 24. Vergados, D. J., Sgora, A., Vergados, D. D., Vouyioukas, D., & Anagnostopoulos, I. (2010). Fair TDMA scheduling in wireless multihop networks. In *Telecommunication Systems*. doi:10.1007/s11235-010-9397-9.
 25. Yi, S., & Pei, Y. (2003). On the capacity improvement of ad hoc wireless networks using directional antennas. In *Proceedings of ACM MobiHoc*, Annapolis, MA, USA (pp. 108–116).
 26. Zhang, J., & Jia, X. (2009). Capacity analysis of wireless mesh networks with omni or directional antennas. In *Proceedings of IEEE Infocom*, Rio de Janeiro, Brasil (pp. 2881–2885).
 27. Żotkiewicz, M., & Pióro, M. (2011). Modeling wireless mesh networks with directional antennas. In *Proceedings of ETS*, Poznań, Poland.



Mateusz Żotkiewicz is an assistant professor at the Computer Networks and Switching Division at the Institute of Telecommunications, Warsaw University of Technology, Poland. He received the M.Sc. degree in telecommunications in 2007 from Warsaw University of Technology, and the Ph.D. degree in 2011 from Warsaw University of Technology and Telecom SudParis, France. His research interests concentrate on modeling and design of telecommunications networks.



Michał Pióro is a professor and head of the Computer Networks and Switching Division at the Institute of Telecommunications, Warsaw University of Technology, Poland, and a professor at Lund University, Sweden. He received the Ph.D. degree in telecommunications in 1979 and the D.Sc. degree in 1990, both from the Warsaw University of Technology. In 2002 he received a Polish State Professorship. His research interests concentrate on modeling, design and performance evaluation of telecommunication systems. He is an author of four books and more than 150 technical papers presented in the telecommunication journals and conference proceedings. He has led many research projects for telecom industry in the field of network modeling, design, and performance analysis. He is deeply involved in international research projects including FP7, Celtic and COST projects.