WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2015; **15**:924–946
Published online 19 July 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.2393

RESEARCH ARTICLE

Fair optimization of mesh-connected WLAN hotspots

Peter Dely[1], Fabio D'Andreagiovanni[2] and Andreas Kassler[1*]

[1] Karlstad University, 65188 Karlstad, Sweden
[2] Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 14195 Berlin, Germany

ABSTRACT

In Wireless mesh networks mesh access points (MAPs) forward traffic wirelessly towards users or Internet gateways. A user device usually connects to the MAP with the strongest signal, as such MAP should guarantee the best quality of service. However, this connection policy may lead to: (i) unfairness towards users that are distant from gateways; (ii) uneven distribution of users to MAPs; and (iii) inefficient use of network paths. We present a new model and solution approach to the problem of assigning users to MAPs and routing the data within the mesh network with the objective of providing max–min fair throughput. The problem is formulated as a mixed-integer linear programming problem (MILP). Because of the inherent complexity of the problem, real size instances cannot be solved to optimality within the time limits for online optimization. Therefore, we propose an original heuristic solution algorithm for the resulting MILP. Both numerical comparisons and network simulations demonstrate the effectiveness of the proposed heuristic. For random networks, the heuristic achieves 98% of the optimal solution. Network simulations show that in medium-sized networks, the number of users with at least 1 Mbit/s minimum end-to-end rate increases by 550% when compared with the classical signal-strength based association. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS
wireless mesh networks; fairness; flow optimization

***Correspondence**
Andreas Kassler, Karlstad University, Universitetsgatan 2, 65188 Karlstad, Sweden.
E-mail: andreas.kassler@kau.se

1. INTRODUCTION

Wireless local area networks (WLANs) have become extremely popular to provide Internet access in homes, schools, or public places. Such deployments typically cover a small geographic area and are called hotspots. To extend the area of hotspots in a cost-efficient way, network providers have increasingly used wireless mesh networks (WMNs) to interconnect WLAN access points (APs) [1]. In a WMN, mesh nodes (MNs) forward traffic by a wireless connection on behalf of mesh APs (MAPs) or other MNs. Only a subset of the MN, the mesh gateways (MGWs), is connected to a wired distribution network like the Internet. MAPs offer access to user stations (STAs), just like normal APs.

In many practical network deployments, an STA may access the Internet through a range of different (M)APs, routing paths, and MGWs. For example, Figure 1 shows the results of a site survey for the Karlstad University campus, which was made using the Cisco Wireless LAN Control System. With IEEE 802.11g and a PHY rate of 24 Mbit/s, roughly 55% of the users in the hotspot area can choose from only one AP. At 45% of the locations, users can choose between 2, 3, and even 4 APs. With 12 Mbit/s, even more choices are possible. In this example, the campus WLAN does not use a WMN. However, when a WMN is used as a backbone, the APs can be deployed even denser thus allowing even more choices.

When an STA wants to connect to a hotspot of a certain network provider (which might use a mesh network to connect the APs to the Internet), the WLAN driver normally searches for all APs with the service set identifier of the desired network provider. It typically connects to the AP with the highest received signal strength (RSS). Using the AP with the highest signal strength is often problematic and may lead to congestion because some APs may need to serve a large portion of the users, whereas others may be underutilized. Figure 2 illustrates this problem in a small network. The entire traffic of one STA needs to be routed via one AP and the mesh to a Network Operations Center (NOC), where access control and accounting functions are performed. The MGWs are connected to the NOC via xDSL and Generic Routing Encapsulation (GRE) tunnels. In such a setup, which is typical for commercial hotspot deployments, RSS-based STA/MAP association and minimum hop-count routing results in three

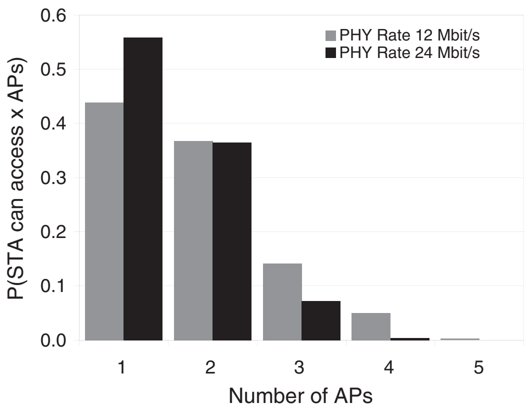


Figure 1. Likelihood being in coverage range from several APs in Karlstad University Campus WLAN.

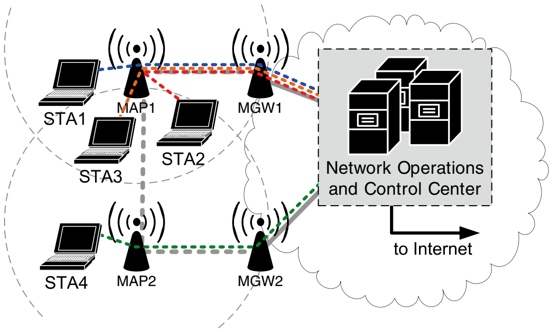


Figure 2. Example of RSS association/minimum hop count.

stations using MAP1 as AP and MGW1 as gateway, whereas MAP2 and MGW2 are only used by one station. This leads to unfairness: STA1-3 need to share resources, whereas STA4 exclusively uses one AP and MGW. By distributing the load better (Figure 3), STA1-3 now receive higher flow rates, as resources are only shared among two instead of three users.

The goal of this paper is to develop mechanisms that allow the network to better utilize available resources of mesh-connected hotspots. This is achieved by optimizing user/MAP associations, path selections, flow rates given fairness, and network resource constraints.

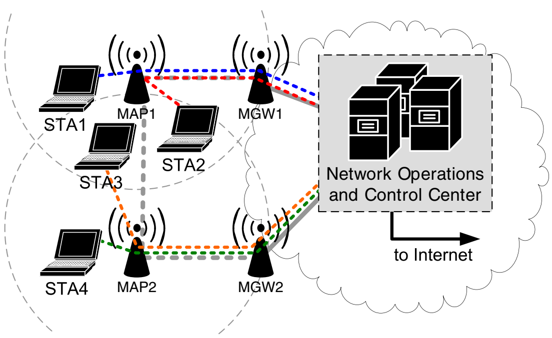


Figure 3. Example of optimized association/routing.

1.1. Related work

Related work falls into two areas: optimization of STA-AP selection in WLANs and optimization of STA-AP selection and routing in WMNs.

For normal WLANs (without mesh backhaul), the problem has been studied extensively from an optimization theory perspective as well as from a system architecture design perspective. Bejerano *et al.* [2] study fairness in STA-AP association problem. The authors propose an approximation algorithm to compute a max–min fair rate allocation. Li *et al.* [3] introduce a nonlinear formulation for the optimization problem. The authors propose a heuristic solution algorithm to compute a proportional fair rate allocation that guarantees to achieve at least 50% of the optimal solution. Bahl *et al.* [4] present a mixed integer linear programming (MILP) formulation for the problem and implement an optimization system using cell-breathing concepts known from cellular networks. To speed up the computation, Bejerano *et al.* [2], Li *et al.* [3], and Bahl *et al.* [4] use linear relaxations of integer constraints. Bahl *et al.* [4], in addition, decompose the initial problem, so that instead of solving the initial MILP, a minimum weight bipartite matching problem is solved. Bipartite matching problems can be solved to optimality in polynomial time. In [5], the re-optimization problem of WLAN associations is studied. With dynamic user arrivals and departures, it may be necessary to perform handovers of stations frequently and hence disrupt ongoing connections. The optimization model in [5] then provides a method to determine when such a handover should be performed, given that there is a cost associated with it.

Whereas Bejerano *et al.* [2], Li *et al.* [3], and Bahl *et al.* [4] focus on the optimization problem itself, Vasan *et al.* [6], Vasudevan *et al.* [7], Ahmed and Keshav [8], and Murty *et al.* [9] use simple heuristics to find an AP-STA selection, but propose system architectures for actually implementing such solutions in a real network. Such simple heuristics in general do not provide good performance. Recently, D'Andreagiovanni *et al.* [10] and D'Andreagiovanni [11] have demonstrated that in real network deployments, the adaptation of optimization techniques leads to a more effective and efficient use of the limited available resources.

Although the aforementioned works provide valuable insights and give inspiration for the efficient optimization of STA-AP associations, they focus on infrastructure WLANs only. In a WMN, those algorithms might still provide low performance: Connecting to a lightly loaded AP may result in low throughput, as the wireless mesh backbone from the AP to the MGW might be overloaded or the gateway has low available capacity. Hence, the joint optimization of routing and STA-AP associations can give large benefits, as shown in [12–14]. The STA-MAP association problem in the context of WMNs has been studied as a nonlinear optimization problem in [12,13]. Whereas Kim *et al.* [12] state that there is a trade-off between fairness and throughput, Luo *et al.* [13] claim *to improve throughput while enhancing fairness*. Määttä and Bräysy [15] propose a heuristic algorithm to solve the association and routing problem. The three latter works use simple fairness concepts. In [12] and [15], the bit rate of any two STA cannot differ by more than a fairness ratio. However, such ratio may limit the rate of one STA, even if it does not compete with a second STA that receives a lower rate. Luo *et al.* [13] claim to achieve max–min fairness and indeed do not compute the lexicographically maximum rate vector, as required by the classical textbook definition [16].

In [17–21], max–min fair rate allocations are computed according to [16]. For example, Dong *et al.* [17] jointly optimize routing and rate allocation but rely on *clients* to forward traffic on behalf of other clients. This is typically not possible in WMNs. Tang *et al.* [18] study max–min fair rate allocation and routing in WMNs. However, Tang *et al.* [18] might require multipath routing, which often leads to practical problems such as packet reordering and higher jitter.

Leith *et al.* [19] propose a convex optimization model for max–min fairness in WMNs on the basis of the IEEE 802.11e standard. Raniwala *et al.* [20] propose a congestion control algorithm to achieve max–min fairness in WMNs. Finally, Pioro *et al.* [21] present a mixed-integer linear programming model for max–min fair rate allocation in WMNs. The latter three papers do not optimize routing.

1.2. Contributions

To the best of our knowledge, no other optimization model and a corresponding fast heuristic for WMNs have been proposed previously that jointly consider single-path routing, AP selection, and rate allocation under max–min fairness. Under practical considerations, such a model would be desirable, as the single-path routing avoids packet reordering and max–min fairness does not require *a priori* knowledge of traffic demands.

Some previous works only consider a subset of those aspects and do not consider an integrated optimization problem. Other works jointly optimize all those aspects but rely on simple heuristics and do not compare them against the solutions obtained by the exact methods. It is thus not possible to evaluate the quality of the produced solutions.

The main objective of our work is to close the gaps in the literature that we highlighted, proposing a fast heuristic for the integrated planning of all the considered aspects of the mesh optimization problem. Additionally, we assess the quality of the obtained solutions w.r.t. those obtained by an exact approach.

The *key contributions* of this paper are as follows:

- A *compact mathematical formulation* (i.e., the number of constraints is polynomial in the size of the input) for the problem: Specifically, the set of solutions to the problem is defined through a mixed-integer linear programming (MILP) formulation. The resulting optimization problem can, in principle, be solved by an effective commercial solver, although in practice, this cannot be performed within the implicit time limit imposed by the application;
- A *fast heuristic algorithm*: Because of the long time required to find an optimal solution, the exact approach is not suitable for online optimization. We hence develop a novel heuristic solution algorithm, which iteratively solves the STA-MAP association and the routing and rate allocation problem. For medium to large size networks (up to 300 users), the run-time does not exceed 10 s, which makes it suitable and competitive for online network optimization. For a number of random networks, we show that the minimum per STA rate computed by our heuristic roughly matches 98% of the rate computed by the exact solution algorithm;
- *Extensive evaluation with network simulations*: We extensively evaluate the solution algorithms and compare them with two simple heuristics, which use the RSS and the distance of an MAP to the next MGW (similar to what is performed in [22]). We show the superiority of our heuristic over the other heuristics. In particular for TCP connections, fairness improves greatly.

We stress that, to the best of our knowledge, this is the first paper which (i) compares a heuristic to the exact solution in terms of algorithm run-time and solution quality *and* (ii) verifies the effectiveness of the approach using network simulations over a large amount of topologies and traffic scenarios.

The remaining paper is organized as follows: Sections 2 and 3 describe in detail the optimization problem and solution algorithms. Section 4 evaluates different solution algorithms. In Section 5, we compare the performance of the solution algorithms with two simple distributed heuristics using ns-2 simulations. Section 6 discusses how the optimization scheme could be implemented in real networks. Finally, in Section 7, we derive some closing conclusions.

2. SYSTEM MODEL

2.1. Notation

The considered mesh network is modeled as a directed graph $G(V, E)$. The set of nodes V is the union of three sets: (i) a set U that includes one element for each user station; (ii) a set M that includes one element for each MN; and (iii) a set W that includes one element for each MGW. An additional node t is also introduced to represent a wired connection to the NOC and the Internet. So, $V = U \cup M \cup W \cup \{t\}$. The set of edges E includes one element for each potential communication link between nodes in V, and each edge $(u, v) \in E$ is associated with

Wirel. Commun. Mob. Comput. 2015; **15**:924–946 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

Table I. Summary of notation.

Symbol	Description
U	Set of STA
M	Set of non-gateway mesh nodes
W	Set of mesh gateways
t	Virtual node representing the Internet
V	Set of nodes $V = U \cup M \cup W$
E	Set of links
G	Connectivity graph $G(V, E)$
K	Set of contenting links
H	Contention graph $H(E, K)$
v_u	Source of traffic demand for STA u
f_{ij}^u	Flow of STA u on link (i, j)
x_{ij}^u	Indicator variable if STA u uses link (i, j)
$F^u(j)$	Total flow of node j by STA u
$X^u(j)$	Total number of links at node j by STA u
b_{ij}	IP-layer capacity of link (i, j)
R	Set of feasible rate allocation vectors

a non-negative value $b_{(u,v)} \geq 0$ representing the maximum IP-layer bit rate (the rate usable by Internet Protocol (IP) connections, taking into account Medium Access Control (MAC) and Physical Layer (PHY) layer overheads) that link (u, v) can support. The achievable IP-layer bit rate depends on PHY rate and the interference model. In Appendix B of this paper, we show how to compute $b_{(u,v)}$ for the collision domain model.

The collision domain of a link (u, v) contains all other links that cannot be active at the same time as (u, v) and can be determined as shown in Appendix B. Once the collision domains are established, the *contention graph* $H(E, K)$ can be built in the following way: The set of nodes coincides with the set E of edges in the connectivity graph $G(V, E)$. An edge is added to K to connect two nodes of (i, j) and $(k, l) \in E$ if $(k, l) \in \mathcal{C}(i, j)$. In the following, if (i, j), (k, l) are edges of the graph G that share resources, we denote by $((i, j), (k, l))$ the edge connecting the corresponding nodes in G.

We remark that in this case, we are modeling a relaxed resource sharing condition, as we require *all* links of a collision domain to be inactive. A more refined modeling of the resource sharing condition can be obtained through the so-called *clique model* [23]. However, this model requires to compute maximal independent sets, which is Non-deterministic Polynomial-time (NP) hard. Moreover, the *collision domain model* has been shown to be reasonably accurate [24] and is thus widely used (e.g., [25–27]).

A summary of the notation used throughout the rest of the paper is given in Table I.

2.2. Feasible solution set

Each STA uses its corresponding station $u \in U$ to download data from node $v_u \in V$ via a hotspot that is connected to the mesh network. The data transmission request generated by each STA can thus be identified by a tuple (u, v_u) that indicates the destination and source nodes of the transmission in the graph G. We assume a single-path routing protocol, which imposes that the entire traffic flow of each request must be routed on a single path.

The problem that we study can be described as follows: Given the mesh network graph $G(V, E)$ including a set U of STA and a set M of MNs, the contention graph $H(E, K)$ and the vector of edge bit rate capacity $\boldsymbol{b} \in \mathbb{Z}_+^{|E|}$, the *Mesh Max-Routing Problem (MESHMAX)* is the one establishing a single routing path for each STA so as to maximize the flow rates sent on the network, while respecting the capacity of each link and the resource sharing relations imposed by the contention graph. Note that by computing a path for each STA, we implicitly compute the STA-MAP association, as the STA and the MAP are the first two nodes among the ones constituting the path.

The MESHMAX problem can be naturally modeled as a variant of the *unsplittable multicommodity flow problem* [28], where the flow models the transfer rate associated to each STA. In particular, we refer to a formulation based on edge flows, where we introduce a non-negative continuous variable f_{ij}^u $\forall u \in U, (i, j) \in E$ to represent the bit rate of STA $u \in U$ on link $(i, j) \in E$ and a binary variable x_{ij}^u $\forall u \in U, (i, j) \in E$ that is equal to 1 if the entire traffic of u is routed on edge (i, j) and 0 otherwise. The latter variable is needed to model the unsplittable flow requirement. The unsplittable multicommodity flow problem has no constraints on fairness and hence can lead to the starvation of users. Hence, later when solving MESHMAX, we impose additional fairness constraints.

In Equations (1)–(6), we define the set of feasible solutions for the MESHMAX problem. Equation (1) models the flow conservation. For each node j and each STA u, the amount of flow $F^u(j)$ associated with STA u and entering node j must be equal to the amount exiting the node, except in the origin node v_u and destination node u. Equation (2) ensures that each STA routes its entire traffic on a single path between u and v_u. The capacity constraint (3) ensures that the sum of flows sent on an link does not exceed the capacity of the collision domain of the corresponding link. Specifically, the overall flow that is considered in the left-hand side of each constraint (3) includes the following: (i) the sum of the flows of all STAs sent on edge $(i, j) \in E$ plus (ii) the sum of flows of all STAs sent on edges $(l, k) \in E$ that share the bandwidth of (i, j) according to the contention sharing graph $H(E, K)$ (this requires that we must add the flows of edges associated to nodes in Z that are adjacent to the node representing edge (i, j) in the graph H). η is a constant to model the efficiency of the channel access protocol. We normalize the flow rates by dividing by b_{ij}; that is, flows sent on faster links occupy the wireless channel less. Constraint (4) ensures that if edge (i, j) is not used to route the traffic of u, then no flow of u can be sent over it. Finally, Equations (5) and (6) define the decision variables of the problem.

$$\sum_{(i,j)\in E} f_{ij}^u - \sum_{(j,i)\in E} f_{ji}^u = F^u(j)$$

$$\text{with } F^u(j) = \begin{cases} \sum_{(v_u,i)\in E} f_{v_u i}^u & j = u \\ 0 & \text{otherwise} \\ -\sum_{(u,j)\in E} f_{uj}^u & j = v_u \end{cases} \qquad \forall u \in U, \forall j \in V \tag{1}$$

$$\sum_{(i,j)\in E} x_{ij}^u - \sum_{(j,i)\in E} x_{ji}^u = X^u(j)$$

$$\text{with } X^u(j) = \begin{cases} 1 & j = u \\ 0 & \text{otherwise} \\ -1 & j = v_u \end{cases} \qquad \forall u \in U, \forall j \in V \tag{2}$$

$$\sum_{u\in U} f_{ij}^u / b_{ij} + \sum_{(k,l)\in E:((i,j),(k,l))\in H} \sum_{u\in U} f_{kl}^u / b_{kl} \le \eta \qquad \forall (i,j) \in E \tag{3}$$

$$f_{ij}^u \le b_{ij} \cdot x_{ij}^u \qquad \forall u \in U(i,j) \in E \tag{4}$$

$$f_{ij}^u \ge 0 \qquad \forall u \in U(i,j) \in E \tag{5}$$

$$x_{ij}^u \in \{0,1\} \qquad \forall u \in U(i,j) \in E \tag{6}$$

2.3. Discussions

Constraint (3) is based on the assumption that contending links can share resources arbitrarily, for example, by time division. It further assumes that the efficiency of the channel access is independent of the traffic in the network. Kumar *et al.* [25] show how to find a schedule of finite length for arbitrary legal flow rates in Time Division Multiple Access (TDMA) networks under such assumptions.

Many deployed WMNs however adopt the IEEE 802.11 MAC layer, which uses random access. Unfortunately, computing η and the achievable throughput for IEEE 802.11 requires lengthy computations with no closed-form solution [29]. In particular, when the network load is close to the capacity of the network, there are many collisions, which require a complicated model. However, Daneshgaran *et al.* [30] show that one can precisely approximate the throughput of an IEEE 802.11 station in non-saturated conditions as a linear function, as long as the offered load is not too high. Garetto *et al.* [31] adopt a similar approach and model the IEEE 802.11 throughput as a piecewise linear function. Dely *et al.* [32] present measurements from a real network, which support this claim. The intuition behind [30] and [31] is simple: If the network load is low, there is a small chance for collisions, and hence, all offered load contributes to the throughput. By setting η to a value smaller than 1, one makes sure that the network always stays in the linear region, and a linear approximation is valid. Clearly, by using $\eta < 1$, one cannot fully utilize the network, but the throughput difference associated with, for example, $\eta = 0.9$ and $\eta = 1$, is minor, as most of the additionally sent packets create collisions and do not contribute to the throughput. Hence, constraint (3) is also a good approximation for IEEE 802.11 networks.

Another critical aspect is the time-varying and stochastic nature of the wireless channel. We model the channel gain factor $\mathcal{G}$ as a fixed value. However, because of variations in the propagation environment, $\mathcal{G}$ may change thus resulting in a new connectivity and contention graph. Collecting and conveying instantaneous channel feedback to a central entity, which solves the MESHMAX problem, is intrinsically difficult. Consequently, $\mathcal{G}$ should be interpreted as an average on a longer period, which is easier to obtain in practice.

The system model given can, in principle, also be applied to multi-radio and multichannel WMNs. In such networks, the interference can be reduced by using orthogonal channels and minimizing the number of interfering links. Assuming that channel assignment was computed in a preliminary phase, our model can be applied as is by just specifying the proper collision domains.

The focus of this paper is in the optimization of routing and associations. However, for the sake of a complete formulation, in Appendix A, we show how to extend our model and algorithms, for the case where one would like to optimize both channel assignment and routing.

2.4. Objective function and fairness considerations

It is reasonable to assume that user utility increases as the flow rate increases. Hence, we aim to maximize the flow rate of each STA. For notational convenience, we define a rate allocation vector $\boldsymbol{r} = (r_1, \ldots, r_{|U|})$, which denotes the flow rate of each STA, that is, $r_u = \sum_{(v,u)\in E} f_{vu}^u$. The set of feasible rate allocation vectors $\boldsymbol{R}$ is described by Equations (1)–(6). We would like to maximize $\boldsymbol{r}$ over all $\boldsymbol{r} \in \boldsymbol{R}$. Hence, the following optimization problem needs to be solved:

$$\textbf{maximize} \{\boldsymbol{r} : \boldsymbol{r} \in \boldsymbol{R}\} \tag{7}$$

Problem (7) is not a standard optimization problem, as it tries to optimize a vector of user rates and not a single real number. A standard approach to solve such an optimization problem is to define an aggregate objective function $f(\boldsymbol{r}) : \mathbb{R}^{|U|} \rightarrow \mathbb{R}$ and then maximize $f(\boldsymbol{r})$. One of the simplest aggregation functions is the so-called max–sum:

$$\textbf{maximize} \left\{\sum_{u \in U} r_u : \boldsymbol{r} \in \boldsymbol{R}\right\} \tag{8}$$

The drawback of max–sum optimization is the complete lack of fairness. Some STAs may receive high rates, whereas other STAs are completely deprived of bandwidth. Consider for example a simple network consisting of two MNs v_1 and v_2 and two STAs u_1 and u_2, such that u_1 can only be associated to v_1 and u_2 to v_2. We additionally assume that all links are in one collision domain and that the IP layer bit rate of the links is 10 Mbit/s. In this case, the max–sum optimization would allocate 10 Mbit/s to u_1 and 0 Mbit/s to u_2—a very unfair assignment. To avoid such deprivation, fairness needs to be introduced into the optimization model.

In a hotspot scenario, traffic demands of users are not known *a priori*. We hence apply the classic concept of *max–min fairness*, which does not require predetermined traffic demands.

Definition 1. *By $\vec{\boldsymbol{r}}$, we denote the bit rate allocation vector $\boldsymbol{r}$ with its entries sorted in non-decreasing order. Following the classical definition of* [16], *we call a rate allocation $\vec{\boldsymbol{r}}$ max–min fair, if $\boldsymbol{r} \in \boldsymbol{R}$; that is, it is feasible and if for any other feasible allocation $\vec{\boldsymbol{s}}$; if $\vec{s}_i > \vec{r}_i$, there exists some j such that $\vec{r}_j < \vec{r}_i$ and $\vec{s}_j < \vec{r}_j$; that is, if $\boldsymbol{r}$ is max–min fair, it is not possible to increase one entry of $\boldsymbol{r}$ without decreasing a smaller entry.*

Finding a max–min fair allocation is closely related to the *lexicographic ordering* of rate vectors. Nace and Pioro [33] define the lexicographic order of two vectors as follows:

Definition 2. *We call a vector $\boldsymbol{r} \in \mathbb{R}^m$ lexicographically greater than vector $\boldsymbol{s} \in \mathbb{R}^m$, $\boldsymbol{r} \succ \boldsymbol{s}$, if there exists an index $i \in \{1, \ldots, m-1\}$, such that $r_j = s_j$ for all $j = \{1, \ldots, i\}$ and $r_{i+1} > s_{i+1}$. If $\boldsymbol{r} \succ \boldsymbol{s}$ or $\boldsymbol{r} = \boldsymbol{s}$, we write $\boldsymbol{r} \succeq \boldsymbol{s}$.*

Definition 3. *A lexicographic maximization* lex max $\boldsymbol{r}$: $\boldsymbol{r} \in R$ *finds the lexicographically greatest vector $\boldsymbol{r}$ in the feasible set R.*

Intuitively, a vector $\vec{\boldsymbol{r}} \in R$ is max–min fair, if there exists no other vector $\boldsymbol{s} \succeq \boldsymbol{r}$. We will use this relationship in the following algorithms.

3. SOLUTION ALGORITHMS FOR MESHMAX

Our aim is to solve lex max $\boldsymbol{r} : \boldsymbol{r} \in R$. As such an optimization problem cannot be solved directly with standard LP or MILP solution algorithms, we proceed with presenting three algorithms for solving the MESHMAX problem (i.e., the MILP defined by Equations (1)–(6)) with the max–min fairness objective. We describe one exact solution algorithm and two faster heuristics.

The MESHMAX problem needs to be solved each time a user enters or leaves the network or the network topology changes. The first algorithm (MESHMAX-OPT) that we present computes the optimal max–min fair allocation in single-path networks. This is computationally very intensive and only feasible for small networks. The second algorithm solves a relaxed version of the original MESHMAX problem, by allowing to split a flow as it traverses the network (MESHMAX-LP). The outcome of this algorithm is then used to derive a rounded single-path version.

The third algorithm (MESHMAX-FAST) takes into account the two-tier nature of WMNs, with infrastructure nodes and STAs, that generate service requests. Instead of solving the MESHMAX problem for all STAs, it solves it for MAPs and then uses a matching algorithm to associate STAs to MAPs.

3.1. Optimal max–min fair rate allocation (MESHMAX-OPT)

We use the idea of *conditional means*, which was introduced in [34] to compute a max–min fair resource allocation over non-convex sets. The basic idea of this approach is to define $\tau_k = \sum_{i=1}^{k} \vec{r}_i$, which represents the k-th cumulative ordered value and can be computed by solving the following optimization problem:

$$\tau_k = \textbf{minimize} \sum_{u \in U} r_u w_{uk} \tag{9}$$

$$\sum_{u \in U} w_{uk} = k \tag{10}$$

$$0 \leq w_{uk} \leq 1 \qquad \forall u \in U \tag{11}$$

With w, we denote a continuous variable to weight the importance of each individual r. Note that this problem is nonlinear, as both w and are r variables. Taking the dual of the program leads to the following linear formulation, in which τ can be computed:

$$\tau_k = \textbf{maximize}\, k\beta_k - \sum_{u \in U} \lambda_{ku} \tag{12}$$

$$\beta_k - r_u \leq \lambda_{ku} \qquad \forall u \in U \tag{13}$$

$$\lambda_{ku} \geq 0, \qquad \forall u \in U \tag{14}$$

Ogryczak *et al.* [34] then show that solving lex max $\boldsymbol{r}$ is equivalent to solving lex max $\{\tau_1, \ldots, \tau_{|U|}\}$. The latter lexicographic maximization problem can then be solved by iteratively solving the following optimization problem (OP1):

$$\textbf{maximize } \tau_k \tag{15}$$

$$\tau_k \le k\beta_k - \sum_{u\in U} \lambda_{ku} \tag{16}$$

$$\tau_l^* \le l\beta_l - \sum_{u\in U} \lambda_{ku} \qquad \forall l \in \{1,\ldots,k-1\} \tag{17}$$

$$\beta_l - r_u \le \lambda_{lu} \qquad \forall l \in \{1,\ldots,k-1\}, u \in U \tag{18}$$

$$\lambda_{iu} \ge 0 \tag{19}$$

$$\{r_1,\ldots,r_{|U|}\} \in R \tag{20}$$

where β is an unbounded continuous variable and τ_l^* is the optimal solution of iteration l. Constraints (16)–(19) and variables β and λ are introduced for calculating the conditional mean. For details on transforming a non-convex lexicographic optimization problem into the form previously, we refer the reader to [34] or [33]. OP1 cannot be used to compute τ_1^*. This can be carried out by solving the following similar (and simpler) problem (OP2):

$$\textbf{maximize } \tau_1 \tag{21}$$

$$-r_u + \tau_1 \le 0 \qquad \forall u \in U \tag{22}$$

$$\{r_1,\ldots,r_{|U|}\} \in R \tag{23}$$

The max–min fair rate allocation vector $\boldsymbol{r}$ is then computed with Algorithm 1.

Algorithm 1 Max-min Fair Bandwidth Allocation for Multi-Commodity Single-Path Networks

Input : $\boldsymbol{R}$
Output : Max-min fair rate allocation vector $\boldsymbol{r}^*$
(1) $r_1^* := \tau_1^* :=$ Solve OP2
(2) foreach k $\in \{2, \ldots, |U|\}$
(3) $\quad \tau_k :=$ Solve OP1
(4) $\quad r_k^* := \tau_k - \tau_{k-1}$
(5) end

3.2. Relaxed max–min fair rate allocation (MESHMAX-LP)

The MESHMAX-OPT is a particular case of an unsplittable maximum flow problem and, with respect to the specific application that we consider, requires a long time to be solved in reasonable quality. We thus consider faster solution approaches, starting by considering a relaxed version of the problem that can be solved more easily. Specifically, we drop the single-path routing requirement, thus allowing the flow to be split among multiple paths between every origin–destination pair. We remark that solving the splittable version of the max–flow problem is computationally easier and the solution provides an upper bound on the unsplittable version. The solution to the relaxed splittable problem is then given as input to a fast heuristic that derives an (in general non-optimal) unsplittable solution.

$$\textbf{maximize } \alpha \tag{24}$$

$$\sum_{(i,j)\in E} f_{ij}^u - \sum_{(j,i)\in E} f_{ji}^u = F^u(j)$$

$$\text{with } F^u(j) = \begin{cases} \alpha & j = u \wedge u \in U\backslash U' \\ -\alpha & j = v_u \wedge u \in U\backslash U' \\ 0 & \text{otherwise} \\ d_u & j = u \wedge u \in U' \\ -d_u & j = v_u \wedge u \in U' \end{cases} \qquad \forall j \in V, \forall u \in U \tag{25}$$

$$\sum_{u\in U} f_{ij}^u / b_{ij} + \sum_{(k,l)\in E:((i,j),(k,l))\in H} \sum_{u\in U} f_{kl}^u / b_{kl} \le \eta \qquad \forall (i,j) \in E \tag{26}$$

$$f_{ij}^u \ge 0 \qquad \forall u \in U, (i,j) \in E \tag{27}$$

To solve the problem of max–min fair rate allocation with splittable flows, we first reformulate OP2 to OP3 (Equations (24)–(27)) and then use the well-known water-filling approach [35].

The OP3 maximizes α, the rate of non-saturated STAs, subject to rates already computed for saturated STAs. For this purpose, we define set U', which contains all saturated STAs. Furthermore, we define a parameter d_u ($u \in U'$), which denotes the traffic demand of a saturated STA u. Equation (25) enforces rate α for each non-saturated STA and a rate d_u for each saturated STA. Constraints (26) and (27) enforce capacity constraints and positive flow rates.

We then iteratively compute d_u with Algorithm 2: At each iteration, newly saturated STAs are identified and added to set U', until all STAs are saturated. To check if an STA u is saturated, one can solve OP3 and fix the demands of all STAs except for u. If the rate r_u can not be increased, STA u is saturated.

We use the results of Algorithm 2 to compute a single-path routing solution for the max–flow problem by consecutively selecting the maximum capacity path for each STA. The maximum capacity path for an STA can be found by constructing a new graph from all edges with positive flow

Algorithm 2 Max-Min Fair Bandwidth Allocation for Multi-Commodity Multi-Path Networks

Input : V, G, H, U, η, b
Output : Max-min fair rate allocation vector $\boldsymbol{r}$

```
d := 0, U' := ∅
while (U\U' ≠ ∅) do
    α := Solve OP3
    U'_new := Identify saturated demands
    U' := U' ∪ U'_new
    foreach u ∈ U'_new
        d_u := α
    end
end
r := d
```

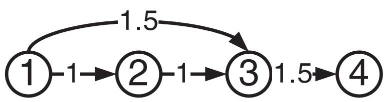


Figure 4. Network for which Algorithm 2 does not terminate.

(computed by Algorithm 2) for the corresponding STA. On this graph, a single-commodity unsplittable flow problem is solved. To our experience, this new graph tends to be small (as we only consider edges that carry flow of the respective STA). Hence, the maximum capacity path can be found very fast by solving the MILP formulation through CPLEX. We remark that strictly polynomial time solution algorithms for this problem exist (e.g., [36,37]). For example, Hu [37] requires solving a Minimum Spanning Tree (MST) problem and theoretically has a lower computational complexity than the MILP formulation. However, when solving the MILP with a state-of-the-art solver such as CPLEX, which has been refined over years and has reached extremely high level of efficiency, it can practically beat also specialized non-commercial prototypal implementation of polynomial time algorithms. We tested for a network with 100 edges and 39 vertices. CPLEX could solve such instance size in 0.01 s, whereas the MST algorithm required 0.33 s. Hence, we use the MILP formulation for all experiments.

Algorithm 2 does not require to introduce the new variables β and λ. On the basis of our computational experience, Algorithm 2 performs better than Algorithm 1. However, Algorithm 2 cannot be applied to the original MESHMAX problem (single-path) as there is the possibility that a saturated user is not found and hence the algorithm may not terminate. This can be shown by the simple example depicted in Figure 4 and the corresponding feasible region as depicted in Figure 5: We consider two users, with demands associated with origin–destination pairs $(1, 3)$ and $(1, 4)$, respectively. In the first iteration of Algorithm 2, each user is assigned 1 Mbit/s. Now, as the paths of both users are not fixed, each demand can potentially be routed via link $(1, 3)$, and none of the users is saturated, as either one of them could potentially be improved to 1.5 Mbit/s. Hence, the termination condition (have only saturated users) of the loop in Algorithm 2 will never be fulfilled.

Graphically, Algorithm 2 can be interpreted as follows: It first finds a point on the efficient frontier of Figure 5 at which both users have the same rate. It then checks if it can improve the rate of one user without decreasing the rate of the other. In the case of single-path routing, the algorithm first finds point $(1, 1)$ but then cannot decide in which direction to progress.

In contrast, when applying Algorithm 2 to multipath networks, both users are assigned with 1.25 Mbit/s in the first iteration, and all links are saturated. Hence, for a non-convex feasible region, as typical for single-path routing, Algorithm 1 needs to be used.

3.3. Fast heuristic solution algorithm (MESHMAX-FAST)

In the case of real network instances, solving MESHMAX-LP corresponds to solving a large LP, making it too complex for online optimization. We can reduce the problem complexity, by considering (i) that in mesh-connected hotspots, most of the traffic is exchanged between the STAs and the Internet (and not among MNs or STAs) and (ii) that typically, there are many STAs but only a few MAPs. In the following, we decompose the

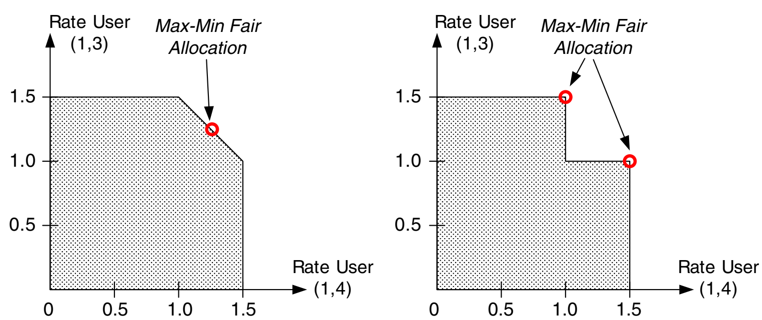


Figure 5. Feasible set for multipath (left) and single-path routing (right).

original MESHMAX problem into a number of subproblems, which can be individually solved fast:

(1) Flow-maximization: Find the maximum flow (under fairness considerations) between STAs and the Internet.
(2) (Initial) STA-MAP assignment: Find an MAP for each STA.
(3) STA-MAP re-assignment: Move STA between MAPs, to increase the minimum STA rate.
(4) Routing: Compute a feasible routing for each STA.

As shown in Figure 6, the subproblems are solved iteratively, until the minimum per STA rate cannot be improved anymore. Instead of directly facing a multi-commodity max–flow problem that requires time to obtain solutions of reasonable quality, we first compute a single-commodity max–flow with the Internet as source and the MAPs as sinks and an overall STA-MAP assignment (computationally simpler). We then use the results to find a feasible solution w.r.t. Equations (24)–(27), that is, the original MESHMAX problem.

This approach works particularly well when the flow traversing an MAP is divided into flows for many STAs and when most traffic is between the STAs and the Internet.

3.3.1. Subproblem I: flow maximization.

We build a reduced connectivity graph $G'(V', E')$, where $V' = M \cup W \cup \{t, t'\}$ and E includes the original mesh backbone links from E as well as edges from all MAP to t' and from all $i \in W$ to t. t is a consolidated source node representing the Internet, and t' is a consolidated sink node. $H'(E', K')$ is the conflict graph of G' and is created as described in Section 2.

We aim to find $\boldsymbol{f}$, which maximizes the flow from the Internet that is passing through MAPs, weighted by the parameter $\boldsymbol{a}$, which is the number of STA associated to an MAP. The corresponding linear optimization problem (LP4) is defined in Equations (28)–(33).

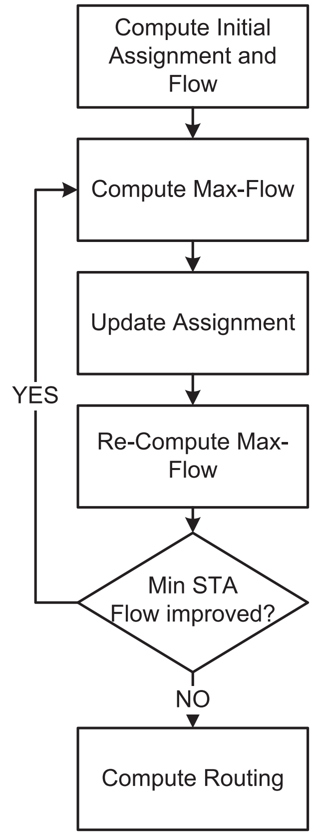


Figure 6. Sequence diagram of the MESHMAX-FAST algorithm.

Similar to Section 2, Equations (29) and (30) ensure flow conservation and capacity constraints. Equations (31) and (33) make sure that STA rates do not differ by more than a configurable factor q ($q = 1$ enforces equality) and that each STA u receives a flow rate of least d_u. With Equation 33, the flow passing through an MAP is distributed to its connected STA equally. Note that LP4 has no integer constraints and that its size does not depend on the number of STA, which makes it much easier to solve than OP1–OP3.

$$\textbf{maximize} \sum_{j \in V'} a_j f_{jt'} \tag{28}$$

$$\sum_{(i,j)\in E'} f_{ij} - \sum_{(j,i)\in E'} f_{ji} = F(j)$$
$$\text{with } F(j) = \begin{cases} \sum_{(i,j)\in E'} f_{ij} & j = t \\ 0 & \text{otherwise} \\ -\sum_{(i,j)\in E'} f_{ij} & j = t' \end{cases} \qquad \forall j \in V' \tag{29}$$

$$f_{ij}/b_{ij} + \sum_{(k,l)\in E':((i,j),(k,l)\in H'))} f_{kl}/b_{kl} \le \eta \qquad \forall (i,j) \in E' \tag{30}$$

$$q a_j f_{it'} \le a_i f_{jt'} \qquad \forall i, j \in V' : a_i, a_j > 0 \tag{31}$$

$$f_{ij} \ge 0 \qquad \forall (i,j) \in E' \tag{32}$$

$$\sum_{(j,t')\in E'} f_{jt'} \ge a_j d_j \qquad \forall j \in V' \tag{33}$$

3.3.2. Subproblem II: establishing STA-MAP assignments.

Each STA should be associated to exactly one MAP. We formulate the STA-MAP assignment problem as a maximum cardinality bipartite matching problem. In this problem, a graph is partitioned into two sets of vertices U and S, so that all edges have one endpoint in U and the other endpoint in S. A matching is a set of edges K such that every vertex is an endpoint of at most one edge in K. Finding the set K of maximum cardinality is called maximum matching problem. This corresponds to finding the max–flow in an augmented graph, a task that can be accomplished for example by the Hopcroft–Karp algorithm [28] or by solving an LP. In our implementation, we relied on the direct use of CPLEX to solve the problem, instead of using our implementation of the Hopcroft–Karp (in our preliminary tests, the latter found the optimal solution to the STA-MAP assignment with 30 clients in 0.35, against the 0.1 s needed by CPLEX).

The STA-MAP assignment is calculated on a graph with the vertex set $U \cup S$, where U are STA and S is a set of virtual service slot nodes. Each MAP m is split into σ_m service slot nodes. σ depends on the capacity of an MAP and will be calculated in Algorithm 5. Edges between nodes in U and S are created according to connectivity given by G. Every $e \in K$ is an association between an MAP and an STA. Figure 7 depicts an example of a graph for finding a matching.

3.3.3. Subproblem III: increasing minimum STA rate.

Given are the flow rate vector $\boldsymbol{f}$, graph G, and a matching K. We would like to find a new association vector $\boldsymbol{a} = \{a_1, \ldots, a_{|V|}\}$ and a corresponding matching K, which maximizes the minimum STA rate. This problem could be formulated as an MILP. As integrality constraints make the problem hard to solve in reasonable time for larger problem instances, we develop a simple heuristic algorithm. The algorithm is based on a re-association graph $R(V, Z)$.

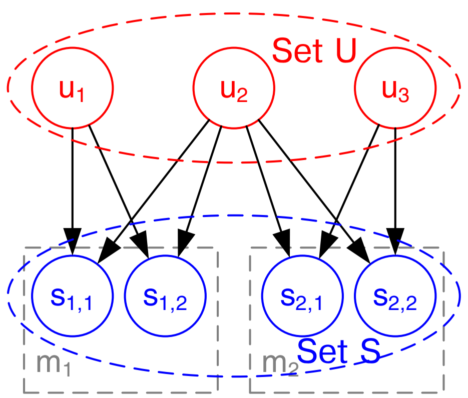


Figure 7. Example: finding a matching, where STA 1 can connect to MAP 1, STA 3 to MAP 2, and STA 2 to both ($\sigma = 2$).

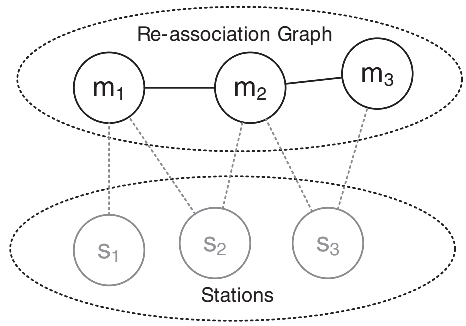


Figure 8. Example re-association graph. s_1 can connect to m_1, s_2 to m_1 and m_2, and s_3 to m_2 and m_3.

An edge $(u, v) \in Z$ is created if there exists an STA that can be associated to both u and v. If there exists a path $p^{u \to v}$, it is possible to increase the number of STAs associated to u and decrease the one of v by re-associating STAs, potentially requiring STAs on other MAPs to re-associate.

For example, in Figure 8, s_1 and s_2 are first connected to m_1, and s_3 is connected to m_2; m_3 has no STA connected. There is no STA on m_1 that could be moved directly to m_3. But following the graph on the reassociation graph, one can see that s_2 can be moved to m_2, and s_3 can be moved to m_3. Hence, we can decrease the association count a of m_1, by increasing the association count on m_3.

Algorithm 3 checks all MAPs pairwise to see if by moving STA along the re-association graph, the minimum STA rate can be increased, while considering the link capacity and the interference according to the contention graph. If this is possible, K is updated accordingly.

Algorithm 3 Increasing minimum STA rates by re-associations

Input : $\boldsymbol{f}, \boldsymbol{a}, R, K$ **Output** : K

```
asc := Sort MAPs by per STA rate, ascending
dsc := Sort MAPs by per STA rate, descending
foreach h ∈ dsc
  foreach l ∈ asc
    if (Re-association graph has a path from l to h and
      Moving a STA from h to l increases the min rate and
      according to R capacity is available) then
       Update K and a
    end
  end
end
```

3.3.4. Subproblem IV: routing.

Given are link flow rates $\boldsymbol{f}$ for the $t \to t'$ max flow and the matching K. For each STA $u \in U$ associated at MAP m, we want to find a path $p^{t \to u} = \{(t, i), \ldots, (m, u)\}$ capable of forwarding f_{mt}/a_m flow without violating capacity constraints imposed by the link flow rates. In general, the $t \to t'$ max flow is only feasible for split flows.

To obtain unsplit flows for individual users, we make use of the following observation: The total flow of an MAP m comprised a_m individual flows, which provides a natural way to split up the total flow of an MAP without splitting traffic of an individual user. We create a routing graph RG from G' by removing all edges with zero flow. Further, let c_{ij} denote the available capacity on edge (i, j). Initially, c_{ij} equals f_{ij}. The cost of each edge (i, j) is set to $(1/c_{ij})^{10}$, so that the use of links with low capacity is penalized. Algorithm 4 finds a path for each user u that does not violate flow constraints for a rate r'_u. The rate is computed as the minimum of r_u and the available capacity on the path. In lines 17–23, the rates are adjusted, by equally splitting rates among all STA sharing the same path.

Algorithm 4 Routing of user traffic

Input : $\boldsymbol{r}, \boldsymbol{c}, RG$
Output : Set of paths P and adjusted rate vector $\boldsymbol{r'}$

(1) $c := f$
(2) Sort users U by ascending rate
(3) foreach u $\in U$
(4) $map :=$ Get MAP of user u based on matching K
(5) $p^{t \to u} :=$ Find minimum cost path from t to u
(6) $r'_u := \mathbf{min}(c_{ij}\, \forall (i,j) \in p^{t \to u}, r_u)$
(7)   foreach $(i,j) \in p^{t \to u}$
(8) $c_{ij} := c_{ij} - r'_u$
(9)   if $c_{ij} > 0$ then
(10) $cost_{ij} := 1/c_{ij}{}^{10}$
(11)   else
(12) $cost_{ij} := \infty$
(13)   end
(14)   end
(15)   Add $p^{t \to u}$ to P
(16) end
(17) foreach u $\in U$
(18) $S :=$ Find all other $v \in U$ with same path as u
(19) $rate := \sum_{v \in S \cup \{u\}} r'_v / (|S| + 1)$
(20)   foreach v $\in S \cup u$
(21) $r'_v := rate$
(22)   end
(23) end

3.3.5. Solution algorithm.

We introduce Algorithm 5 to iteratively solve subproblems I–IV, aiming to maximize the minimum STA rate. Line 1 initializes low, which denotes the minimum throughput that any of the STAs receives. In lines 2–3, an initial matching K is calculated using (**get_matching**) and the number of associated STA computed (**get_ass_cnt**). We call the algorithm MESHMAX-FAST if the initial matching is computed by associating each STA to the MAP with the lowest hop count to the next gateway. For MESHMAX-FAST*, we compute the maximum flow $t \to t'$ on a graph with edges $E \cup \{(u, t') \forall u \in U\} \cup \{(t, v) \forall v \in M\}$ and vertices $V \cup \{t, t'\}$ (i.e., the original graph G is augmented with a consolidated source node t and a sink node t', which are connected to the gateways and STA, respectively). For each STA, we select the MAP from which it receives the highest flow. In lines 4–5, the per STA flow rates are computed (**get_maxflow**), and the smallest are assigned to low.

Lines 6–14 constitute the main loop of the algorithm. In line 8, the link flow rates $\boldsymbol{f}$ are computed by solving LP4 (**get_maxflow**). Equality of all STA is enforced ($q = 1$), and no minimum rates are required ($m = 0$). Lines 9–11 update the matching K by calling Algorithm 3 (**update_matching**) and recompute low_{cur}.[†] In lines 12–13, the flow rates are computed again, this time requiring no fairness ($q = 0$) but guaranteeing at least low_{cur} to each STA. The main loop is repeated until the lowest STA rate does not increase anymore. Finally, routes and adjusted rates are computed using the algorithm 4.

Algorithm 5 MESHMAX-FAST solution algorithm

Input : G, H, U
Output : Rate allocation vector $\boldsymbol{r}$ and paths $\boldsymbol{P}$

(1) $low_{old} := -1$
(2) $K := \mathbf{get\_matching}(\sigma := |U|)$
(3) $a := \mathbf{get\_ass\_cnt}(K)$
(4) $f := \mathbf{get\_maxflow}(q := 1, d := 0, a)$
(5) $low_{cur} := \mathbf{min}(f/a)$
(6) while $low_{cur} > low_{old}$ do
(7) $low_{old} := low_{cur}$
(8) $f := \mathbf{get\_maxflow}(q := 0, d := a * low_{cur}, a)$
(9) $K := \mathbf{update\_matching}(K, f)$
(10) $a := \mathbf{get\_ass\_cnt}(K)$
(11) $low_{cur} := \mathbf{min}(f/a)$
(12) $f := \mathbf{get\_maxflow}(q := 0, d := a * low_{cur}, a)$
(13) $low_{cur} := \mathbf{min}(f/a)$
(14) end
(15) $\mathbf{get\_routes}(f, K)$

4. NUMERICAL PERFORMANCE ANALYSIS

In this section, we compare the performance of the three considered algorithms. All algorithms are implemented in OPL modeling language, and the Linear Programming problems were solved by IBM ILOG CPLEX 12.3 [38]. The tests were performed on a Linux server with an Intel E5606 Quad-Core CPU (2.13 GHz) and 4 GB RAM.

Specifically, the main result that we will show is that when the real time limit imposed by an online mesh problem is considered, even a state-of-the-art LP solver such as CPLEX is not able to find solutions of reasonable quality within the time limit. In contrast, our heuristic approach based on the four sub-algorithms provides a solution at high quality within the time limits required for online optimization of networks.

[†]Note: $\boldsymbol{f}/\boldsymbol{a}$ is a piece-wise division of a flow rate vector $\boldsymbol{f}$ and association count vector $\boldsymbol{a}$

Wirel. Commun. Mob. Comput. 2015; **15**:924–946 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

We compared the solution quality in 30 randomly generated topologies. In each topology, 20 MNs were uniformly scattered over an area of 500×500 m^2. Non-connected networks were discarded. In the network, four gateway nodes were randomly placed and connected to the Internet with a 100 Mbit/s Ethernet connection. The mesh backbone was operated at 54 Mbit/s PHY rate (yielding approximately 32 Mbit/s application layer rate), and the STA-MAP links at 18 Mbit/s (yielding approximately 10 Mbit/s application layer rate). STAs were randomly dropped over the coverage area of the network.

4.1. Throughput

Figure 9 depicts minimum, 90-percentile, maximum, and average throughput as a mean value over 30 random networks. The minimum and 90-percentile throughput show how well the worst-off users are supported, whereas the average and maximum throughput indicate how well the overall network resources are utilized.

The MESHMAX-OPT represents an upper bound for the minimum throughput because OP2 maximizes the throughput of the worst STA (Section 3). MESHMAX-FAST* almost achieves this upper bound, guaranteeing on average 98% of the optimum. This surprisingly good performance can be explained as follows: Optimizing split flows from the Internet to each MAP is a good approximation of optimizing unsplittable flows from the Internet to each STA, as each MAP aggregates traffic of many STAs. The approach of MESHMAX-LP to first optimize split flows for each STA individually and then approximate an unsplit version works slightly worse. For a few STAs, it might occur that the difference between the split and the approximated unsplit throughputs is high. However, as a look at the 90%-ile throughput reveals, this does not happen too frequently. In terms of 90%-ile throughput, MESHMAX-LP almost achieves the performance of MESHMAX-OPT, especially in terms of the 90%-ile throughput.

Interestingly, in terms of minimum throughput, MESHMAX-FAST performs much worse than MESHMAX-FAST*. This shows that using a good initial matching in Algorithm 5 is critical for the performance. If the algorithm starts from a bad initial matching, the reshuffle operation might not attempt to move STA to a dedicated MAP, which is required for a good solution.

The ratio between the maximum and the minimum rates is a simple measure for how fairly resources are distributed. Among all the Pareto-efficient rate allocations, the max–min fair rate allocation minimizes this ratio. Therefore, it also measures how well a rate allocation approximates a max–min fair rate allocation. For MESHMAX-OPT, the ratio was on average 3.19 (max.

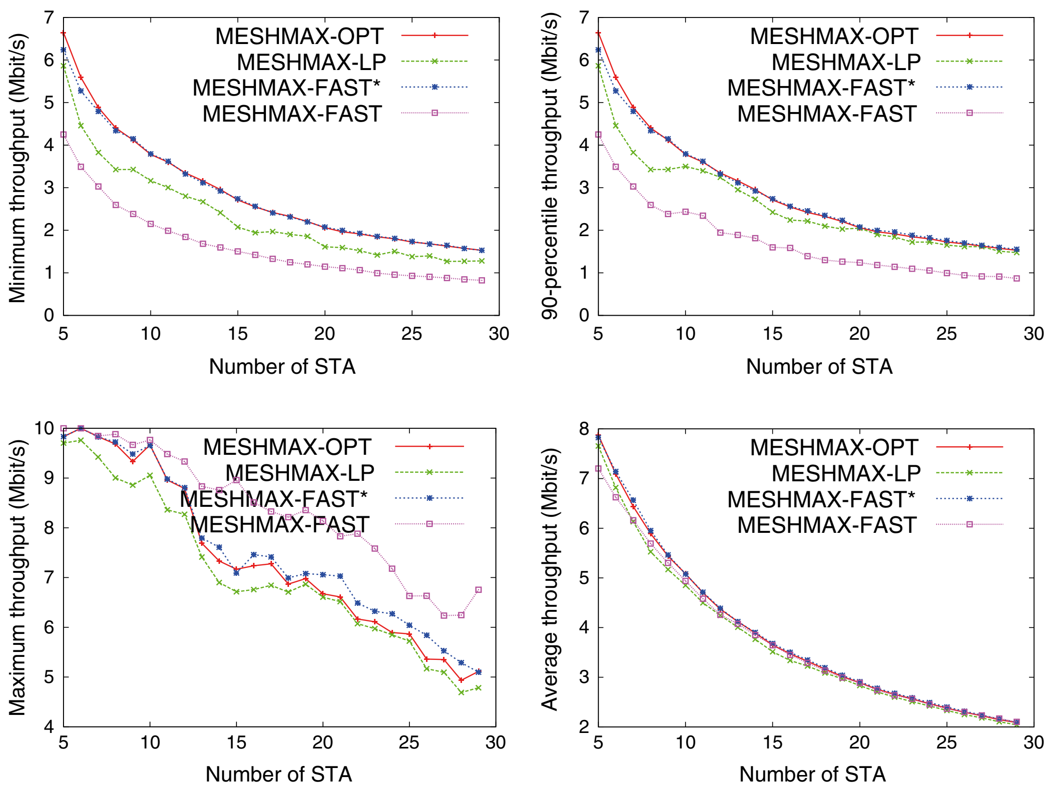


Figure 9. Minimum, 90-percentile, maximum, and average throughput for 30 random networks.

12.03). Instead, it is 3.60 (max. 15.00) for MESHMAX-LP, 3.48 (max. 24.06) for MESHMAX-FAST*, and 6.32 (max 29.06) for MESHMAX-FAST.

For two topologies with a total of 15 and 25 STAs, we made deeper investigations on the relative performance of MESHMAX-LP, MESHMAX-FAST, and MESHMAX-FAST*. In Figure 10, we plot the CDF of the minimum throughput relative to MESHMAX-OPT, which was obtained by considering 30 random STA distributions over the area. A relative performance of 1 is equal to the optimal solution. For all three heuristic algorithms, the relative performance depends on the location of the STA. For example, with MESHMAX-LP, the minimum rate can be equal to 49% and 100% of the optimal solution. A similarly large difference can be observed for MESHMAX-FAST. With MESHMAX-FAST*, the performance is between 70% and 100%, with an average of 98% and a median of 100%. In general, it is desirable that the worst case performance of a heuristic is as high as possible.

4.2. Run-time

As the algorithm is used for online optimization of the network and executed each time an STA joins or leaves the network or the association opportunities change because

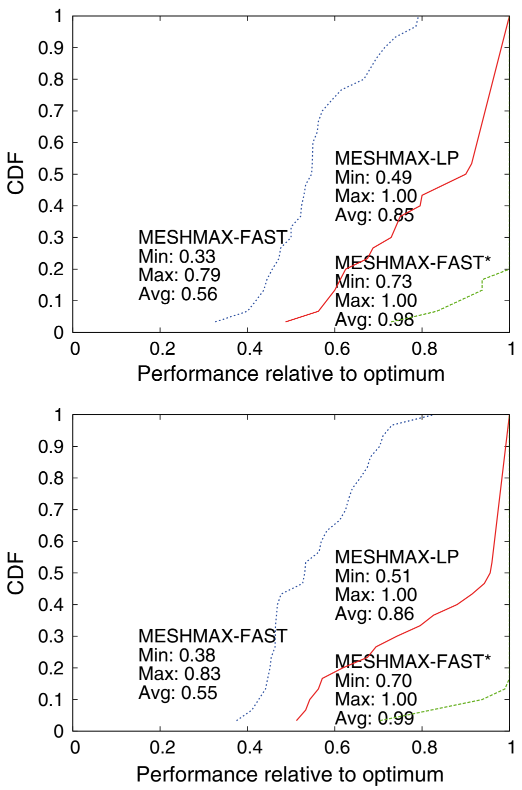


Figure 10. Empirical CDF of minimum throughput relative to optimum for 15 (top) and 25 (bottom) STA.

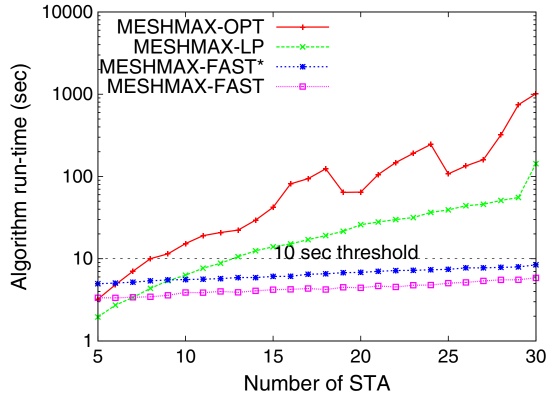


Figure 11. Comparison of algorithm run-time (averaged over 30 random topologies).

of user mobility, the run-time is critical. The rate of such changes depends on the network size and on the usage. Ghosh *et al.* [39] studied more than 1.3 million connections in public WLAN hotspots. In high load situations, the average arrival rate is roughly one customer per minute. To account for bursty arrivals, the run-time of the algorithm should thus be considerably shorter than 1 min, for example, 10 s.

Figure 11 compares the average run-time of the three algorithms in a random topology with 20 MNs and 5–30 STA (observe the log-scale on the y-axis). As expected, the run-time of MESHMAX-OPT and MESHMAX-LP increases at (almost) exponential rate. MESHMAX-OPT and MESHMAX-LP can be used for online optimization (when the threshold is 10 s) in the evaluated networks only when up to 8 and 13 STAs are present respectively. The run-time growth of MESHMAX-FAST and MESHMAX-FAST* is much slower.

When a number of STAs between 5 and 15 is present, MESHMAX-FAST* is slower than MESHMAX-FAST but is relatively faster when more STAs are present. This is because initial matching of MESHMAX-FAST* takes longer but then fewer iterations of the main loop of the algorithm are required.

In [39], the mean number of concurrent STA per AP during busy hours does not exceed 10. Further assuming a network size of 20 MAPs, a target network size of 200 STAs should be solved within the 10 s constraint. For 30 random networks of such size, the average run-time for MESHMAX-FAST was 4.40 s. Even with 300 STAs, the run-time is only 8.36 s.

Larger networks often can be partitioned into smaller logical units. For example, instead of optimizing a network that spans several floors of a building, one can optimize each floor individually. When there is no interference between the logical units and STAs of one unit cannot use MAPs of another unit, optimizing those partitions results in a global optimum.

4.3. Discussion

On the basis of the results of the numerical analysis, we make the following key observations: (i) in terms of solution quality and run-time, the direct solution of a straightforward relaxation (MESHMAX-LP) of the original MILP problem (MESHMAX-OPT) is not as good as an algorithm that considers the structure of the network (MESHMAX-FAST*); (ii) solving MESHMAX-OPT requires too much time to be solved during the online optimization of networks of practical size. However, its solution can be used by system designers to establish a bound on the quality of the solution obtained by heuristics; (iii) MESHMAX-FAST* is a practical heuristic, which reveals to be very useful to find solutions of good quality within the time constraint associated with the application.

5. EVALUATION AND SIMULATION

The numerical analysis provides valuable insights on the performance of the algorithms. However, the algorithms are based on simplified models of the wireless channel and user traffic. We hence conducted a series of ns-2 simulations to investigate how suited the simplified model of the wireless channel and MAC is for the MESHMAX algorithms. Furthermore, the ns-2 simulations enable us to compare the performance of the MESHMAX algorithms to several state-of-the-art heuristics for AP selection.

5.1. Scenario

All simulations were conducted with ns-2 2.33 and the ns-miracle extensions [40].[‡] Each MN is equipped with two radios, of which one is operated using IEEE 802.11a at 5 GHz and solely used in the mesh backbone. The other radio is tuned on 2.4 GHz and provides an AP interface to the stations using IEEE 802.11g. On the backbone, all the radio devices use the same channel, whereas for the STA access, the MAPs are assigned non-interfering channels. The PHY rates were fixed to 36 Mbit/s. Packets are dropped randomly according to the signal-to-interference-plus-noise ratio of a packet.

All MGWs are connected to a core router (as in Figures 2 and 3), which represents the NOC and acts as exchange point to the Internet. If not stated otherwise, the fixed line from the MGW to the core router is operated at 100 Mbit/s. The NOC collects monitoring information from the network, executes the optimization algorithm, and configures the forwarding tables at the MNs and the rate shapers at the core router and MNs. Such functionality could easily be implemented in real deployments using the OpenFlow-based approach in [41]. The rate shapers are configured according to the flow rates computed by the optimization algorithm.

[‡]The source code of all algorithms, simulation scripts, and topologies is available for download at http://www.cs.kau.se/~pdely/downloads

The comparison consists of two network topologies: The first network consists of 15 MNs (of which four are MGWs) and 15 STA deployed in a random way. The other network resembles a subset of a real WMN deployed in Chaska [42]. It consists of 67 MNs (16 MGWs) and 50 randomly dropped STA.

The MESHMAX algorithms were compared with two schemes that can be implemented in a fully distributed way: an RSS-based association policy and a minimum hop-count scheme. In the RSS-based scheme, an STA associates to the MAP with the highest signal strength (default policy in most modern operating systems). In the hop-count scheme, each MAP broadcasts its hop-count towards the next gateway in periodic beacon messages. An STA then associates to the MAP with lowest hop count.

Each simulation was run for 180 s and repeated 30 times with different random STA drops. The error bars in the plots later are the standard deviation of the 30 random network instantiations. TCP throughput was measured with TCP New Reno, Selective ACK option, and one ACK per segment (no delayed ACKs). As a comparison, we also measured the UDP throughput with backlogged CBR traffic and 1400 bytes datagram length. This allows us to evaluate the impact of the TCP congestion control.

5.2. Throughput performance

In Figures 12 and 13, we plot the UDP and TCP throughput for the random network. For the MESHMAX algorithms, we additionally plot the analytical optimum. MESHMAX-FAST* and MESHMAX-OPT are about equal, followed by MESHMAX-FAST. With the RSS-based scheme, the minimum throughput is about 20 times lower than with the MESHMAX schemes. The hop-count scheme is much better, as it avoids having STA using MAPs far from the next gateway. The performance of the MESHMAX algorithms very closely match the predictions made by the model. On average, 99% of the model predictions are reached.

With TCP, the picture changes slightly: TCP implements a congestion control algorithm to determine the send rate and requires ACKs, which both reduce the available rate and can exacerbate unfairness issues. With TCP, the RSS-based scheme leads to almost complete starvation of some flows. It is a well-known behavior of TCP that flows that start close to the gateway can starve other longer flows (see for example [43]). The hop-count scheme improves the situation because STAs are associated close to the GWs. The MESHMAX algorithms, in addition, shape the rates, so that some flows cannot completely capture all resources.

The analytical optimum and the TCP throughput diverge more than the UDP throughput. The simulations reach on average 84% of the analytical throughput. This is not surprising, as the congestion control reacts to losses of TCP DATA or TCP ACK frames by reducing the send rate. If one TCP frame is lost, it takes a while until TCP recovers to the rate computed by the MESHMAX algorithm. Hence, the average rate over the whole simulation time is lower than the maximum reachable by a TCP connection.

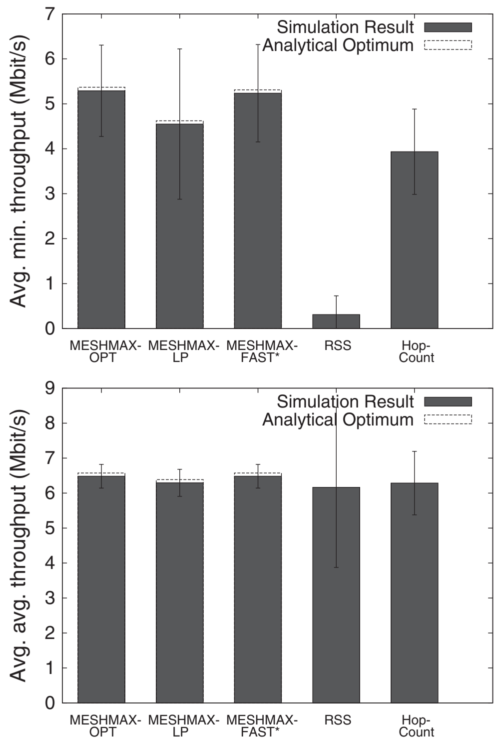


Figure 12. Minimum (top) and average (bottom) UDP throughput in a random network.

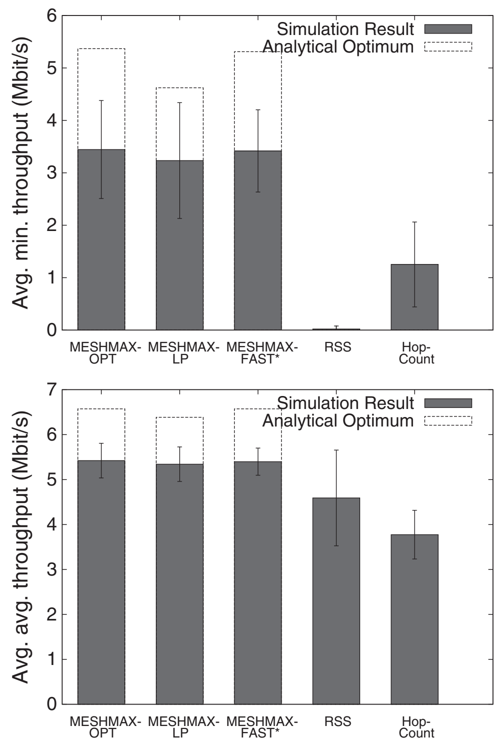


Figure 13. Minimum (top) and average (bottom) TCP throughput in a random network.

The Chaska topology is too large for MESHMAX-OPT and MESHMAX-LP to be solved in acceptable time. We therefore only compared the performance of MESHMAX-FAST* and the RSS and hop-count policy. For both UDP and TCP, MESHMAX-FAST* performs better than the RSS-based and the hop-count-based schemes. With UDP traffic and MESHMAX-FAST*, each STA receives on average 5.6 Mbit/s, which is 22% more than with the RSS-based scheme and 27% more than with the hop-count-based scheme. With TCP traffic, MESHMAX-FAST* on average provides 4.5 Mbit/s for each STA, which is 20% and 40% higher than the RSS and hop-count schemes. With the RSS and the hop-count schemes, some TCP flows are completely starved, resulting in zero throughput. MESHMAX-FAST* at least provides on average 0.44 Mbit/s to the worst-off STA.

5.3. Impact of bottlenecks at gateways

Many WLAN hotspots are connected to the Internet via an xDSL line, which might be slower than the wireless backhaul and hence present a performance bottleneck. We analyzed how sensitive the performance of the MESHMAX algorithms is to different gateway connection speeds. Figure 14 plots the minimum UDP throughput relative to the optimum given by MESHMAX-OPT as an average of 30 simulations using random topologies.

The measurements reveal that there is an impact for all algorithms, albeit no clear trend can be seen for some of them. MESHMAX-LP performs better with higher gateway speeds because the difference between split and unsplit flow rate is lower in that case. MESHMAX-FAST* shows no clear trend. However, it is always outperforming the other schemes.

5.4. Increase in network scalability

From a network operators point of view, it is interesting how many customers can be supported with a minimum

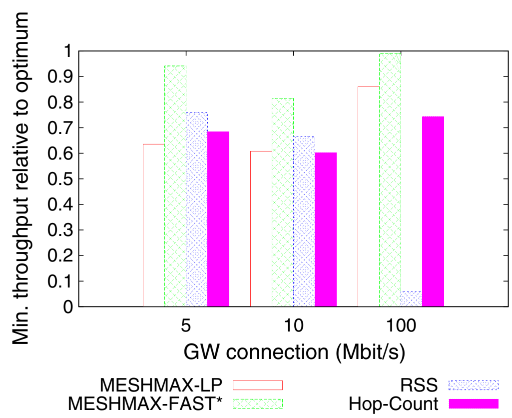


Figure 14. Impact of different gateway connection speeds.

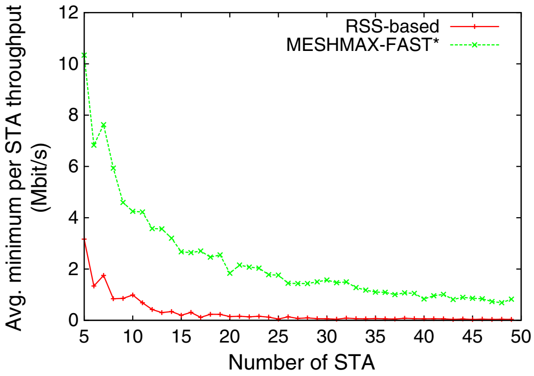


Figure 15. Increased network scalability through MESHMAX algorithms.

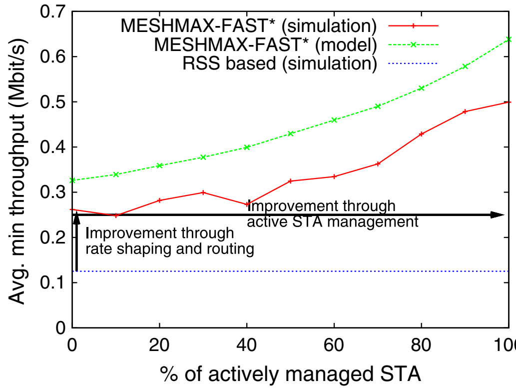


Figure 16. Average minimum throughput when associations are controlled on a fraction of all STA.

rate. In Figure 15, we plot the simulated average minimum throughput value for each STA over 30 random topologies for MESHMAX-FAST* and the RSS-based association. If the network operator would like to provide on average at least 1 Mbit/s rate for each STA, then the default scheme (RSS) can only support seven concurrent STAs. With MESHMAX-FAST*, 39 users can be supported. Figure 15 shows the actual value of the proposed optimization algorithm: With a given network infrastructure and rate requirement, a much higher number of customers can be served. For a 1 Mbit/s rate requirement, a 5.5 fold improvement in the number of supported users is possible. Or conversely, for a given user population of, for example, 15, the minimum rate can be increased from 200 kbit/s to 2.6 Mbit/s.

5.5. Importance of active STA management

We call an STA *actively managed* if it can inform the optimization algorithm, which MAPs are in reach, and if it can follow a handover request. In practice, not all STA are actively managed, as the IEEE 802.11 standard does not provide mechanisms for monitoring reachability of MAPs and performing handovers. Only newer standards, such as IEEE 802.11k [44] and IEEE 802.21 [45], enable such features. The MESHMAX algorithms do not distinguish between actively or non-actively managed STA. However, for non-actively managed STA, the MAP is selected by the STA in a (potentially) non-optimal way and cannot be changed by the algorithm, even if better MAPs might be in reach.

In Figure 16, we plot the average minimum UDP throughput of 30 random topologies with 50 STAs, given that only a fraction of the STAs is actively managed and the rest uses the default RSS-based association scheme. As a comparison baseline, we also provide the throughput if all STAs use the RSS-based association and no rate shaping is performed. Not surprisingly, an increase of actively managed STAs allows the algorithm to find a better network configuration, and hence, the throughput is higher. The throughput increases from 0.25 to 0.5 Mbit/s, when going from 0% to 100% of actively managed STA.

The comparison of MESHMAX-FAST* with 0% actively managed STA and the RSS-based scheme shows that limiting flow rates and optimizing the routing double the throughput from 0.125 to 0.25 Mbit/s. This shows that even in a network with 100% legacy IEEE 802.11 STA, major performance improvements are possible.

6. IMPLEMENTATION IN REAL NETWORKS

Although a presentation of a practical implementation of the proposed schemes is out of the scope of this paper, we briefly discuss an implementation considering the architectures proposed in [8,41] or [46]. For example, with [41], it is possible to exercise fine grained control over routing and traffic shaping in mesh networks using OpenFlow.

The optimization schemes require the active cooperation of STAs to report information about which APs are in reach and to allow controlling handovers between APs. IEEE 802.11k [44], a recent IEEE standard that describes the exchange of monitoring information between APs and STAs, can be used to obtain information about available connection opportunities and interference from the STA. Alternatively, an MAP can monitor the wireless channel to detect which devices are in reach, even if a device is not associated the MAP.

The IEEE 802.21 [45] or IEEE 802.11v [47] can be used to trigger almost seamless handovers from one MAP to another. Dely *et al.* [41] showed through test-bed measurements that mechanisms provided by [45] or [47] result in average of 210 ms in interruption during the handover. IEEE 802.11v is an amendment to the latest IEEE 802.11 standard and hence is likely to be deployed widely.

If STAs do not support any of the aforementioned standards, they can still be forced to connect to the MAP computed by the optimization algorithm: By forcefully disassociating an STA from an MAP and setting MAC filters, an STA can be forced to connect a different MAP with the same service set identifier. This mechanism is not as seamless as handovers triggered by IEEE 802.21. According to our measurements, it takes approximately 3 s for an STA to find a new MAP and connect to it.

If STAs leave and join very frequently and they do not allow active management through IEEE 802.21 or IEEE 802.11v, frequent re-associations and the resulting interruptions may lead bad user experience. Nevertheless, network operators may still use the algorithms we described: By constructing the graph for finding the STA/MAP assignments in such a way that already connected STAs are forced to remain connected to their current MAP, frequent re-associations are avoided. This of course may lead to a suboptimal solution, as the operator can only optimize routing and rate allocations in such a case. By using [41], routing and rate allocations can be changed by the network operator using OpenFlow without disruption to the user.

The operator can, in addition, solve the optimization problem allowing STAs to associate to any MAP and compare the solution to the scenario where STAs need to remain at their current MAP. If by rearranging the STAs, the solution quality becomes considerably better, the operator then can reconfigure the network accordingly. This scheme involves a trade-off between how often a complete rearrangement of STA is allowed and the gap between the current and the optimal network configuration (which potentially requires a complete rearrangement of STA). Characterizing such a trade-off is an interesting problem, which we leave for the future work.

7. CONCLUSIONS

In this paper, we have presented and compared several different approaches for optimizing the throughput and fairness of STAs in mesh-connected hotspots. The presented network simulations show that the proposed optimization algorithms yield major improvements of flow rates, in particular for users that otherwise would be starved. For network operators, this translates into more satisfied users. The exact solution algorithm provides an upper bound, which can be used as a benchmark. The MESHMAX-FAST* heuristic can be implemented and used for online optimization of networks.

This paper is focused on the optimization problems and solution algorithms. We however did not address the question of an architecture for implementing such algorithms in this paper. Currently, we are designing and evaluating an architecture based on Software Defined Networking using OpenFlow, which is based on standard hardware and software can provide all functionality and flexibility for running such an algorithm [41].

Another interesting topic for future investigations is to apply the MESHMAX algorithms in parallel to the multi-radio multichannel approach in WMNs. By jointly optimizing the MAC layer and the channel assignment, as proposed in [48] and [49], additional performance improvement could be achieved. Many traffic demand-aware channel assignment algorithms sequentially optimize routing and channel assignment. The MESHMAX algorithms could be used as one step in such global optimization process as we detailed in the annex, but more research needs to be done to achieve a practical solution to the problems.

ACKNOWLEDGEMENTS

We would like to thank Nico Bayer, Dmitry Sivchenko, Hans Einsiedler, and Martin Grötschel for their helpful comments in the development of this work. Moreover, we would like to thank the anonymous reviewers for their valuable comments and suggestions that helped to significantly improve the paper.

The work was financially supported by Deutsche Telekom Innovation Laboratories as well as by the *German Federal Ministry of Education and Research* (BMBF), project *ROBUKOM* [50], grant 03MS616E.

APPENDIX A: OPTIMIZATION OF MULTI-RADIO/MULTICHANNEL NETWORKS

We distinguish two different approaches for channel assignment: the *integrated channel assignment problem* (CA-INT) and the *sequential channel assignment problem* (CA-SEQ). With CA-INT, we refer to the problem of assigning channels, while at the same time considering all the other aspects of the MESHMAX problem, namely finding STA/MAP associations, routes, and rate allocations. Solving MESHMAX with CA-INT provides optimal solution quality, but as we will show later, it is computationally infeasible for larger networks.

Thus, in practice, one often solves the problems sequentially, leading to the CA-SEQ approach outlined in Section A.2.

A.1. Feasible solution set for CA-INT

In the following, we extend the previously defined feasible solution set to the case where MNs are equipped with multiple radios that can be operated on orthogonal channels. We assume each node $i \in V$ is equipped with θ_i radios that can be tuned to a channel $c \in C$, where C denotes the set of orthogonal channels. For example, in IEEE 802.11g, channels 1, 6, and 11 are orthogonal. Each radio can be tuned to exactly one channel, and to communicate with a neighbor, both the sender and the receiving node need to have a radio tuned to the same channel. The assignment of channels to a node is represented by the binary variable a_i^c, where a value of 1 indicates that channel c is assigned to one of the radios of node i and 0 means it is not assigned.

$$\sum_{(i,j)\in E}\sum_{c\in C} f_{ij}^{uc} - \sum_{(j,i)\in E}\sum_{c\in C} f_{ji}^{uc} = F^u(j)$$

$$\text{with } F^u(j) = \begin{cases} \sum_{(v_u,i)\in E}\sum_{c\in C} f_{v_u i}^{uc} & j = u \\ 0 & \text{otherwise} \\ -\sum_{(u,j)\in E}\sum_{c\in C} f_{uj}^{uc} & j = v_u \end{cases} \qquad \forall u \in U, j \in V \tag{A1}$$

$$\sum_{(i,j)\in E}\sum_{c\in C} x_{ij}^{uc} - \sum_{(j,i)\in E}\sum_{c\in C} x_{ji}^{uc} = X^u(j)$$

$$\text{with } X^u(j) = \begin{cases} 1 & j = u \\ 0 & \text{otherwise} \\ -1 & j = v_u \end{cases} \qquad \forall u \in U, j \in V \tag{A2}$$

$$\sum_{u\in U} f_{ij}^{uc}/b_{ij} + \sum_{(k,l)\in E:((i,j),(k,l))\in H}\ \sum_{u\in U} f_{kl}^{uc}/b_{kl} \le \eta \qquad \forall (i,j) \in E, c \in C \tag{A3}$$

$$f_{ij}^{uc} \le b_{ij}\cdot x_{ij}^{uc} \qquad \forall u \in U, c \in C, (i,j) \in E \tag{A4}$$

$$\sum_{c\in C} a_i^c \le \theta_i \qquad \forall i \in V \tag{A5}$$

$$2x_{ij}^{uc} \le a_i^c + a_j^c \qquad \forall u \in U, c \in C, (i,j) \in E \tag{A6}$$

$$f_{ij}^{uc} \ge 0 \qquad \forall u \in U, c \in C, (i,j) \in E \tag{A7}$$

$$x_{ij}^{uc} \in \{0,1\} \qquad \forall u \in U, c \in C, (i,j) \in E \tag{A8}$$

$$a_i^c \in \{0,1\} \qquad \forall c \in C, i \in V \tag{A9}$$

$$\text{maximize} \sum_{(i,j)\in E} REM(i,j) + s_{ij}$$

$$\sum_{c\in C} f_{ij}^c = d_{ij} + s_{ij} \qquad \forall (i,j) \in E \tag{A10}$$

$$\sum_{c\in C} x_{ij}^c = 1 \qquad \forall (i,j) \in E \tag{A11}$$

$$f_{ij}^c/b_{ij} + \sum_{(k,l)\in E:((i,j),(k,l))\in H} f_{kl}^c/b_{kl} \le \eta \qquad \forall (i,j) \in E, c \in C \tag{A12}$$

$$f_{ij}^c \le b_{ij}\cdot x_{ij}^c \qquad \forall c \in C, (i,j) \in E \tag{A13}$$

$$\sum_{c\in C} a_i^c \le \theta_i \qquad \forall i \in V \tag{A14}$$

$$2x_{ij}^c \le a_i^c + a_j^c \qquad \forall c \in C, (i,j) \in E \tag{A15}$$

$$f_{ij}^c \ge 0,\ x_{ij}^c \in \{0,1\} \qquad \forall c \in C, (i,j) \in E \tag{A16}$$

$$s_{ij} \ge 0 \qquad \forall (i,j) \in E \tag{A17}$$

We extend the initial feasible solution set by adding an index c to the flow and link variables f and x. The flow and path conservation constraints Equations (A1) and (A2) are rewritten so that the total amount of flows/paths is sent via all channels. The capacity constraint Equation (A3) is now per channel. Equation (A4) ensures that data is sent only via active links and channels. Equation (A5) enforces that at maximum θ_i channels are used by a node, and Equation (A6) makes sure that links can only be active if the sender and the receiver node have this channel assigned. Finally, Equations (A7)–(A9) define the domain of the variables f, x, and a.

The CA-INT can be solved with Algorithm 1. Subsequently, we call this problem MESHMAX-OPT/CA-INT.

A.2. Sequential channel assignment—CA-SEQ

Because of the computational hardness of CA-INT, in practice, one often prefers a sequential treatment of the channel assignment and the routing problem. The main idea of CA-SEQ is to couple the channel assignment and routing problems by using the following steps:

(1) In a first step, the MESHMAX problem is solved on a single radio network.

Table A1. Performance comparison of MESHMAX-LP/MESHMAX-FAST*/MESHMAX-OPT with CA-SEQ.

	MESHMAX-FAST*	MESHMAX-LP	MESHMAX-OPT
Minimum download rate (averaged over all simulation runs; Mbit/s)	2.19	2.49	3.19
Average algorithm run-time (s)	9.36	33.24	408.48

(2) The link flow rates computed in step 1 are used as input to the channel assignment problem described in Section A.2.1. This problem determines a channel assignment.
(3) On the basis of the channel assignment, one can compute a new interference graph, which contains only links operated at non-orthogonal channels. With this interference graph, the MESHMAX problem is solved again.

A.2.1. Channel assignment with CA-SEQ.

For the CA-SEQ approach, we need to compute a feasible channel assignment. CA-SEQ is independent of the actual channel assignment algorithm, as long as the network connectivity is ensured by the channel assignment. For example, [51] could be used. For sake of a complete description of the problem, we propose a simple MILP formulation of the channel assignment problem in Equations (A10)–(A17), which aims to maximize remaining link capacity after satisfying specified link traffic demands. More specifically, the flow on each link (i, j) has to be equal to the traffic demand d_{ij} plus a positive slack variable s_{ij} (Equation (A10)). The traffic demand d_{ij} is an input parameter to the problem, as, for example, computed by solving the MESHMAX problem on a single-channel instance of the same network. $REM(i, j)$ denotes the remaining capacity in the collision domain of link (i, j) and is computed as $REM(i, j) = \eta - f^c_{ij}/b_{ij} - \sum_{(k,l)\in E:((i,j),(k,l))\in H} f^c_{kl}/b_{kl}$. Including the slack capacity in the objective function makes sure that channels are assigned in such a way that not only the traffic demands are fulfilled but also the channel assignment is good when the demands are increased (i.e., the slack is used).

As mentioned earlier, a is channel to node assignment variable (Equations (A14) and (A15)), and x is channel to link assignment variable (Equation (A13)). The maximum number of different channels is bounded by the number of radios θ (Equation (A14)). Note that this problem has no flow conservation constraints, no bundle constraints, and no user indices on f and x. Hence, the problem is computationally easier to solve than CA-INT.

A.2.2. Variants of CA-SEQ.

Depending on the MESHMAX algorithm used to solve CA-SEQ, we distinguish the following variants:

- MESHMAX-FAST*/CA-SEQ
- MESHMAX-LP/CA-SEQ
- MESHMAX-OPT/CA-SEQ

The MESHMAX-FAST*/LP/OPT refers to the algorithms used in steps 1 and 3 of CA-SEQ. The channel assignment is found with the MILP formulation as previously described. This sequential execution of the algorithm of course leads to a suboptimal performance, when compared with the integrated solution provided by MESHMAX-OPT/CA-INT. However, we remark again that for real deployments, the solution of the complete optimization problem is not practicable because it requires too long time. Our original heuristic provides instead good quality solutions in much less time. As a potential extension of this algorithm, one could use different algorithms in steps 1 and 3, for example, MESHMAX-FAST* in step 1 and MESHMAX-OPT in step 3.

A.3. Numerical performance evaluation

We have evaluated the performance of the proposed multichannel/multi-radio optimization problems and their respective solution algorithms with 30 random network topologies, each consisting of 25 mesh routers and 25 stations. Each mesh router is equipped with two radios, which can be tuned to three orthogonal channels. Even on a machine with 144 GB of RAM and manual tuning of the symmetry breaking settings of the solver, CPLEX could not compute a solution to MESHMAX-INT. We therefore only report the results of sequential approaches in Table A1.

As expected, MESHMAX-OPT* provides the best solution quality, followed by MESHMAX-LP and MESHMAX-FAST*. Although the difference in achievable download rates is not very large, the algorithm run-times differ drastically. MESHMAX-FAST* is three times faster than MESHMAX-LP and almost 44 faster than MESHMAX-OPT. MESHMAX-FAST* is therefore the only algorithm among the three that is suitable for practical online-network optimization.

We think that a refined integration of the steps of the sequential approach could improve even more the performance of our original approach.

APPENDIX B: COMPUTATION OF THE IP-LAYER BIT RATE

The IP-layer bit rate of a link depends on the PHY-layer bit rate, which is determined as follows: For each link, a node can choose a PHY-layer bit rate from a set of bit rates $\{\zeta_1, \ldots, \zeta_k\}$. Each bit rate ζ_i corresponds to a modulation and coding scheme, for which a minimum signal-to-interference-plus-noise ratio of γ_i is required. Given a

 Wirel. Commun. Mob. Comput. 2015; 15:924–946 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

link (u, v), a node u then chooses the highest bit rate ζ_i for which the following inequality holds:

$$\frac{\mathcal{P}_u \mathcal{G}_{uv}}{\mathcal{N} + \mathcal{P}_i \mathcal{G}_{iv}} \geq \gamma_i \qquad \forall (i, j) \in \mathcal{C}(u, v) \qquad \text{(B1)}$$

where $\mathcal{P}_u$ denotes the transmission power of node u, $\mathcal{G}_{uv}$ is the channel gain ratio on link (u, v), $\mathcal{N}$ is the thermal noise, and $\mathcal{C}(u, v)$ is the *collision domain* of link (u, v). The collision domain $\mathcal{C}(u, v)$ contains all links that share the same frequency and which cannot be active at the same time with link (u, v). In TDMA-based systems, the scheduler needs to ensure that a link is not scheduled simultaneously with one or more of the links in the collision domain.

In the case of the IEEE 802.11 standard, the collision domain is determined implicitly by carrier sensing. Each node u has a carrier sensing threshold δ_u. If the RSS is above the sensing threshold, the node backs off and does not transmit. To ensure error-free reception of all links, the following inequality must hold:

$$P_u G_{ui} \geq \delta_i \qquad \forall (u, v) \in E : (i, j) \in C(u, v) \qquad \text{(B2)}$$

If a node i is the collision domain of link (u, v), the carrier sensing threshold must be low enough to detect a transmission on link (u, v). This ensures that there are no hidden nodes, that is, nodes that cannot hear transmission that they could interfere with. Clearly, one would like to choose the highest possible bit rate ζ (to maximize throughput) and the lowest possible carrier sensing threshold δ (to increase spatial reuse). From a practical point of view, changing δ is not possible on most wireless cards. Hence, we assume that δ is fixed and the bit rate ζ is set statically, so that inequality (B1) and (B2) are fulfilled for all links.

We model the IP-layer bit rate for link (u, v) with PHY rate ζ_j as

$$b_{uv} = \sum_{i=1}^{\texttt{MTUSIZE}} \pi_i \left(i \Big/ \left(OH_{\text{fixed}} + \frac{OH_{\text{var}} + i}{\zeta_j} \right) \right) \qquad \text{(B3)}$$

where i is the payload length (in bits), MTUSIZE is the maximum transferable payload length (in bits), π_i is the probability of sending a packet of length i, OH_{fixed} are the fixed duration components of the overhead that are not dependent on the PHY rate (e.g., channel access or PHY preamble), and OH_{var} accounts for PHY rate dependent overhead (e.g., MAC header).

REFERENCES

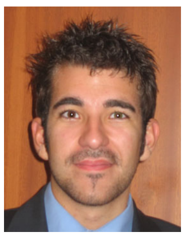
1. EU-MESH: Enhanced, Ubiquitous, and Dependable Broadband Access using MESH Networks - D2.1: Usage Scenarios and Application Requirements, 2008. Public Deliverable The EU-MESH Consortium ICT-215320-EU-MESH-D2.1, 4.
2. Bejerano Y, Han S, Li L. Fairness and load balancing in wireless lans using association control. *IEEE/ACM Transactions on Networking* 2007; **15**(3): 560–573.
3. Li W, Cui Y, Wang S, Cheng X. Approximate optimization for proportional fair ap association in multi-rate wlans. In *Wireless Algorithms, Systems, and Applications*, Vol. 6221, Pandurangan G, Anil Kumar V, Ming G, Liu Y, Li Y (eds), Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, 2010; 36–46.
4. Bahl P, Hajiaghayi M, Jain K, Mirrokni S, Qiu L, Saberi A. Cell breathing in wireless lans: algorithms and evaluation. *IEEE Transactions on Mobile Computing* 2007; **6**: 164–178.
5. Dely P, Kassler A, Bayer N, Einsiedler H, Peylo C. Optimization of wlan associations considering handover costs. *EURASIP Journal on Wireless Communications and Networking* 2012; **2012**(1): 255.
6. Vasan A, Ramjee R, Woo T. Echos - enhanced capacity 802.11 hotspots. In *Proceedings of IEEE INFOCOM 2005*. IEEE: Miami, USA, 2005; 1562–1572.
7. Vasudevan S, Papagiannaki K, Diot C, Kurose J, Towsley D. Facilitating access point selection in ieee 802.11 wireless networks. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*. USENIX Association: Berkeley, California, USA, October, 2005; 26–26.
8. Ahmed N, Keshav S. Smarta: A self-managing architecture for thin access points, In *Proceedings of the 2006 ACM CoNEXT*, Lisboa, Portugal, 2006; 9:1–9:12.
9. Murty R, Padhye J, Chandra R, Wolman A, Zill B, A. Designing high performance enterprise wi-fi networks, In *Proceedings of the 5th UNISEX Symposium on Networked Systems Design and Implementation*, San Francisco, California, 2008; 73–88.
10. D'Andreagiovanni F, Mannino C, Sassano A. GUB covers and power-indexed formulations for wireless network design. *Management Science* 2013; **59**(1): 142–156.
11. D'Andreagiovanni F. Pure 0-1 programming approaches to wireless network design. *4OR: A Quarterly Journal of Operations Research* 2012; **10**(2): 211–212.
12. Kim J, Gong H, Nahm K. Access point selection tradeoff for IEEE 802.11 wireless mesh network. In *4th IEEE Consumer Communications and Networking Conference 2007*. IEEE: Las Vegas, NV, USA, 2007; 818–822.
13. Luo L, Raychaudhuri D, Liu H, Wu M. Joint association, routing and bandwidth allocation for wireless mesh networks, In *IEEE GLOBECOM 2008*, December 2008; 1–8.

14. Athanasiou G, Korakis T, Ercetin O, Tassiulas L. Dynamic cross-layer association in 802.11-based mesh networks, In *Proceedings of IEEE INFOCOM 2007*, Anchorage, Alaska, USA, May 2007; 2090–2098.
15. Määttä J, Bräysy T. A novel approach to fair routing in wireless mesh networks. *EURASIP Journal on Wireless Communication Networks* 2009; **2009**: 5:1–5:13.
16. Bertsekas D, Gallager R. *Data Networks*, 2nd ed. Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1992.
17. Dong Q, Banerjee S, Liu B. Throughput optimization and fair bandwidth allocation in multi-hop wireless lans, In *Proceedings of INFOCOM 2006*, Barcelona, Spain, April 2006; 1–12.
18. Tang J, Hincapie R, Xue G, Zhang W, Bustamante R. Fair bandwidth allocation in wireless mesh networks with cognitive radios. *IEEE Transactions on Vehicular Technology* 2010; **59**: 1487–1496.
19. Leith DJ, Cao Q, Subramanian VG. Realising max-min fairness in 802.11e mesh networks. In *Proceedings of the 5th IEEE International Conference on Wireless Pervasive Computing,* ISWPC'10. IEEE Press: Piscataway, NJ, USA, 2010; 424–429.
20. Raniwala A, Pradipta D, Sharma S. End-to-end flow fairness over ieee 802.11-based wireless mesh networks, In *Proceedings of IEEE INFOCOM 2007*, Anchorage, Alaska, USA, May 2007; 2361–2365.
21. Pioro M, Zotkiewicz M, Staehle B, Staehle D, Yuan D. On max-min fair flow optimization in wireless mesh networks. *Ad Hoc Networks, Special Issue on Models and Algorithms for Wireless Mesh Networks* 2011. DOI: dx.doi.org/10.1016/j.adhoc.2011.05.003.
22. Luo L, Raychaudhuri D, Liu H, Wu M, Li D. Improving end-to-end performance of wireless mesh networks through smart association, In *Proceedings of IEEE WCNC 2008*, Las Vegas, NV, USA, April 2008; 2087–2092.
23. Staehle D, Staehle B, Pries R. Max-min fair throughput in multi-gateway multi-rate mesh networks, In *2010 IEEE 71st Vehicular Technology Conference (VTC 2010-Spring)*, Taipei, Taiwan, May 2010; 1–5.
24. Jun J, Sichitiu M. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications* 2003; **10**: 8–14.
25. Kumar VSA, Marathe MV, Parthasarathy S, Srinivasan A. Algorithmic aspects of capacity in wireless networks. In *Proceedings of ACM SIGMETRICS 2005,* SIGMETRICS '05. ACM: New York, NY, USA, 2005; 133–144.
26. Vanhatupa T, Hännikäinen M, Hämäläinen TD. Performance model for IEEE 802.11s wireless mesh network deployment design. *Journal of Parallel and Distributed Computing* 2008; **68**: 291–305.
27. Huang J-H, Wang L-C, Chang C-J. Coverage and capacity of a wireless mesh network, In *Proceedings of International Conference on Wireless Networks Communications and Mobile Computing 2005*, Vol. 1, Maui, HI, USA, June 2005; 458 –463.
28. Ahuja RK, Magnanti TL, Orlin JB. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1993.
29. Duffy K, Malone D, Leith D. Modeling the 802.11 distributed coordination function in non-saturated conditions. *IEEE Communications Letters* 2005; **9**(8): 715–717.
30. Daneshgaran F, Laddomada M, Mesiti F, Mondin M. On the linear behaviour of the throughput of IEEE 802.11 dcf in non-saturated conditions. *IEEE Communications Letters* 2007; **11**: 856–858.
31. Garetto M, Salonidis T, Knightly EW. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. *IEEE/ACM Transactions on Networking* 2008; **16**: 864–877.
32. Dely P, Kassler AJ, Sivchenko D. Theoretical and experimental analysis of the channel busy fraction in IEEE 802. 11. In *Procedings of Future Network and MobileSummit 2010*, Cunningham P, Cunningham M (eds). IIMC International Information Management Corporation: Florence, 2010; 1– 9.
33. Nace D, Pioro M. Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial. *Communications Surveys Tutorials, IEEE* Quarter 2008; **10**: 5–17.
34. Ogryczak W, Pioro M, Tomaszewski A. Telecommunications network design and max-min optimization problems. *Journal of Telecommunications and Information Technology* 2005; **4**: 43–56.
35. Radunovic B, Le Boudec J-Y. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking* 2007; **15**(5): 1073–1083.
36. Peinhardt M, Kaibel V. On the bottleneck shortest path problem. *Technical Report ZIB-Report 06-22*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2006.
37. Hu TC. The maximum capacity route problem. *Operations Research* 1961; **9**(6): 898–900.
38. IBM ILOG CPLEX. http://www.ibm.com/software/integration/optimization/.
39. Ghosh A, Jana R, Ramaswami V, Rowland J, Shankaranarayanan N. Modeling and characterization of large-scale wi-fi traffic in public hot-spots, In *Proceedings of IEEE INFOCOM 2011*, Shanghai, China, April 2011; 2921–2929.
40. Baldo N, Maguolo F, Miozzo M, Rossi M, Zorzi M. ns2-MIRACLE: a modular framework

for multi-technology and cross-layer support in network simulator 2. In *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools,* ValueTools '07. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): ICST, Brussels, Belgium, Belgium, 2007; 16:1–16:8.
41. Dely P, Kassler A, Bayer N. OpenFlow for wireless mesh networks, In *Proceedings of IEEE International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN 2011)*, July 2011; 1–6.
42. Chaska Wireless. http://www.chaskamn.com/internet-solutions/.
43. Luglio M, Roseti C, Savone G, Zampognaro F. Tcp performance on a railway satellite channel, In *International Workshop on Satellite and Space Communications, 2009. IWSSC 2009*, Siena, Italy, September 2009; 434–438.
44. Draft STANDARD for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Wireless Network Management, July 2009. IEEE Unapproved Draft Std P802.11v/D7.0.
45. IEEE Standard for Local and Metropolitan Area Networks- Part 21: Media Independent Handover, 2009. IEEE Std 802.21-2008, c1–301, 21.
46. Krishnamurthy SV, Faloutsos M, Mhatre V. MDG: measurement-driven guidelines for 802.11 WLAN design. *Proceedings of Mobicom 2007* 2007; **18**: 722–735.
47. IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan networks-specific requirements-part ii: Wireless lan medium access control (mac) and physical layer (phy) specifications: Amendment 9: Interworking with external networks, 2011. Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11w-2009, IEEE Std 802.11n-2009, IEEE Std 802.11p-2010, IEEE Std 802.11z-2010, and IEEE Std 802.11v-2011, 1–208, 25.
48. Bononi L, Di Felice M, Molinaro A, Pizzi S. A cross-layer architecture for effective channel assignment with load-balancing in multi-radio multi-path wireless mesh networks. *International Journal of Communication Systems* 2009; **22**: 1267–1296.
49. Bononi L, Di Felice M, Molinaro A, Pizzi S. Enhancing multi-hop communication over multi-radio multi-channel wireless mesh networks: a cross-layer approach. *Computer Communications* 2011; **34**(13): 1559–1569.
50. Bley A, D'Andreagiovanni F, Hanemann FA. Robustness in communication networks: Scenarios and mathematical approaches. In *Proceedings of the ITG Symposium on Photonic Networks*. VDE Verlag: Leipzig, Germany, 2011; 1–8.
51. Ramachandran KN, Belding EM, Almeroth KC, Buddhikot MM. Interference-aware channel assignment in multi-radio wireless mesh networks, In *Proceedings of IEEE INFOCOM 2006*, Barcelona, Spain, April 2006; 1–12.

AUTHORS' BIOGRAPHIES

Peter Dely is a post-doctoral researcher in the field of wireless network optimization and software-defined networks. He holds a PhD degree in computer science from Karlstad University. He is a recipient of several awards, such as the first price in the ACM MobiCom Student Research Competition 2012.



Fabio D'Andreagiovanni has been a Senior Researcher in the Department of Optimization at Zuse-Institut Berlin since 2010 and Member of the DFG Research Center MATHEON at Technical University Berlin since 2013. He has received his MSc summa cum laude in Industrial Engineering (2006) and PhD in Operations Research (2010) at Sapienza Università di Roma and has been a Research Scholar in the Department of Industrial Engineering and Operations Research at Columbia University in the City of New York (2008-2009). His research on Network Design and Robust Optimization has received several awards, such as the Accenture MSc Prize 2006, the INFORMS Telecom Doctoral Dissertation Award 2010, and the ESF-JSPS Excellence Award "Mathematics for Innovations" 2012.

Andreas Kassler has been a full professor in the Department of Computer Science at Karlstad University in Sweden since 2007, which he joined in 2005 as Associate Professor. He was an Assistant Professor at Nanyang Technological University (NTU) in Singapore in 2003-2004. He has received the MSc degree in Mathematics and Computer Science from Augsburg University, Germany, in 1995 and the PhD degree in Computer Science from the University of Ulm, Germany, in 2002. He has been a Research Scholar at UCLA in 2012. His research on Software-Defined Networking has received several awards such as the ACM Mobicom SRC Award in 2012. He is a member of the IEEE.