A Fair Scheduling for Wireless Mesh Networks

Naouel Ben Salem	Jean-Pierre Hubaux

Laboratory of Computer Communications and Applications (LCA)
EPFL – Lausanne, Switzerland
naouel.bensalem@epfl.ch	jean-pierre.hubaux@epfl.ch

***Abstract*— Wireless Mesh Networks (WMNs) represent a new and promising paradigm that uses multi-hop communications to extend WiFi networks: By deploying only one hot spot (directly connected to the Internet) and several transient access points (TAPs), an Internet Service Provider (ISP) can extend its coverage and serve a large number of clients using a single broadband connection. Unfortunately, if the medium access protocol is poorly designed or inadequate, it can lead to severe unfairness and low bandwidth utilization. In this paper, we propose a fair scheduling mechanism that optimizes the bandwidth utilization in the mesh network. Our solution assigns transmission rights to the links in the WMN and maximizes the *spatial reuse* (i.e., the possibility for links that do not contend to be activated at the same time). We show that our solution is fair and collision-free, and we evaluate its efficiency by means of simulations.**

1 INTRODUCTION

Over the past few years, WiFi networks have become increasingly popular. However, because WiFi communications are short-range, mobile clients need to be in the immediate vicinity of the Internet hot spot to get connectivity; the ISPs have to deploy other hot spots at well-chosen locations to extend the coverage of their networks. However, the acquisition of strategic locations is not always possible due to the *Not In My Back Yard* site acquisition problem [17].

A promising, flexible and low-cost extension of WiFi networks is the concept of Wireless Mesh Networks (WMNs): By allowing multi-hop communications between access points, it is possible for hundreds of Internet users to share a single broadband connection. Indeed, in a WMN, only one hot spot (HS) is connected to the Internet; the rest of the WMN is comprised of transient access points (TAPs) that use wireless communications to transfer their clients' traffic to and from the "wired" hot spot (HS) (see Figure 1).

Several WMNs are already deployed and operational [20, 14, 12] and for these networks, mobile[1] clients usually have to pay a monthly fee for the high-speed Internet connection. However, all the clients in WMN use the same hot spot HS and therefore the throughput they enjoy can fluctuate wildly depending on their distribution in the WMN. But as the clients pay the same flat rate, the throughput sharing should also be fair. Furthermore, as shown in [8], the TAPs that are more than 2 hops away from HS may starve (i.e., their clients may not be able to send or receive traffic), which is highly unfair.

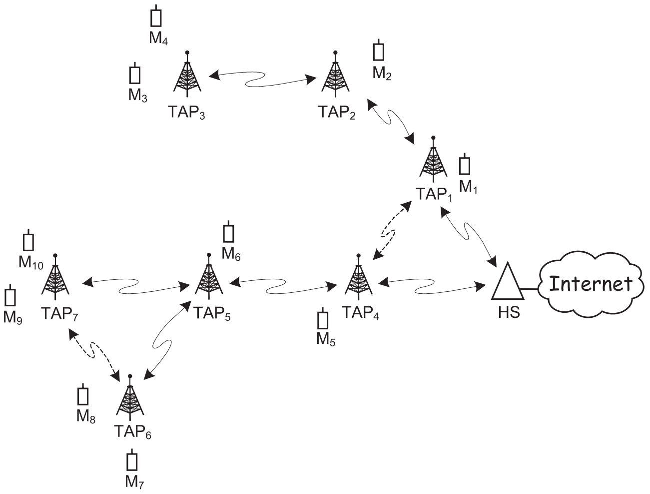

Fig. 1. A Wireless Mesh Network (WMN) comprised of 1 hot spot (HS), 7 transient access points (TAPs) and 10 mobile clients (Ms). HP is directly connected to the Internet whereas the TAPs have to rely on wireless links to get Internet connectivity. The solid lines represent communication links and the dashed lines represent undesired interference.

We propose a scheduling that (i) ensures per-client fairness and (ii) optimizes the bandwidth utilization in the mesh network. The solution assigns transmission rights to the links in a Spatial TDMA (or STDMA [18]) fashion[2] and is collision-free. We chose of the link-based transmission rights assignment rather than the node-based assignment based on the results of [10] where Gronkvist shows that the link-based assignment is preferable in the case of high traffic loads, which is part of our assumptions (see the system model in Subsection 3.1).

This paper is organized as follows: In Section 2, we present the state of the art and we compare our scheduling algorithm to existing solutions. In Section 3, we present the notation, the assumptions and the rationale of the solution. We give the details of the proposed solution in Section 4. We evaluate the efficiency of our solution in Section 5 and discuss several aspects of the proposed protocols in Section 6. Finally, we conclude and present the future work in Section 7.

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number $5005 - 67322$.

[1] Even though the clients are not necessarily mobile, we assume mobility in this paper because it represents the most general case.

[2] In TDMA, no two links can be activated at the same time, whereas in STDMA, two or more links can be activated at the same time if they do not mutually contend. More details are provided in Subsection 3.2.1.

2 STATE OF THE ART

Mesh Networks: In [2], P. Bahl et al. discuss the challenges introduced by the implementation and the deployment of public-area wireless networks (PAWNs) (network security, privacy, authentication, mobility management, provisioning of key services, etc.). They describe CHOICE [3], a PAWN that they have designed and implemented. They describe the architecture and components of CHOICE, the service models it supports, and the location services and context-aware applications that they have implemented and deployed in it.

In [1], Akyildiz, Wang and Wang present a survey on recent advances and open research issues in WMNs and they point out that revising the design of MAC protocols based on TDMA or CDMA is an important research topic. Another overview of mesh networking technology is provided by Bruno, Conti and Gregori in [6].

STDMA Scheduling: In [18], Nelson and Kleinrock define a broadcast channel access protocol called *spatial TDMA* (STDMA), which is designed to operate in a multi-hop packet radio environment where the location of the nodes is fixed. The defined protocol assigns transmission rights to nodes in the network in a local TDMA fashion and is collision-free. The authors propose several slot allocation methods and present an approximate solution that determines the capacity assignment for the links of the network and minimizes the average delay of messages in the system.

In [10], Gronkvist compares the *node assignment* and the *link assignment* methods. The author shows that only the connectivity of the network and the input traffic load of the network is needed in order to determine whether the node or the link assignment is preferable.

In [5] and [24], Bjorklund, Varbrand and Yuan develop mathematical formulations for resource optimization for both *node-oriented* and *link-oriented* allocation strategies. They present a column generation approach that yields optimal or near-optimal solutions. The difference with [10] is that, in [5] and [24], the authors prove the NP-hardness and present a different mathematical formulation.

Fairness in Mesh Networks: In [4], Bejerano, Han and Li propose an algorithm that determines the user-AP associations that ensure max-min fair bandwidth allocation. They study the association control problem and consider bandwidth constraints of both the wireless and backhaul links. Their formulation of the problem indicates the strong correlation between fairness and load balancing, which allows for the usage of load balancing techniques to obtain a near optimal max-min fair bandwidth allocation. Since this problem is NP-hard, they present algorithms that achieve a constant-factor approximate max-min fair bandwidth allocation.

In [8], Gambiroza, Sadeghi and Knightly study per-TAP fairness and end-to-end performance in WMNs (multi-hop wireless backhaul networks). They propose an inter-TAP fairness algorithm that aims to achieve the per-TAP fairness objectives without modification to TCP. This work is the closest to our work, but there are a few fundamental differences:

- The definition of fairness: In [8], the authors consider a per-TAP fairness that is very well suited if a *parking lot-like* scenario[3] is considered, whereas we consider a per-client fairness that is more appropriate if we consider a WMN where all the clients pay the same monthly flat rate, which is the case we consider in this paper. We give a formal definition of the per-client fairness in Subsection 3.2.1.
- The network topology: In [8], the authors consider a single network branch, whereas we consider a network with several branches.
- The traffic model: In [8], the authors consider inter-TAP communications that do not involve the wired access point, whereas in this paper, we consider that the clients are using the WMN to get Internet connectivity and therefore, we assume that the traffic is always from the clients to HS (upstream traffic) or from HS to the clients (downstream traffic).

3 SYSTEM MODEL

In this paper, we consider a mesh network that is composed of one hot spot (HS), n transient access points denoted by TAP_i, $i = 1..n$ and m mobile clients denoted by M_j, $j = 1..m$. An example of such a mesh network is given in Figure 1 (with $n = 7$ and $m = 10$). The TAPs rely on multi-hopping to provide Internet connectivity to the mobile clients. Therefore, a TAP has to handle not only its own clients' traffic (i.e., the traffic of the mobile clients that are within its immediate vicinity) but also the traffic of the mobile clients connected to some other TAPs (e.g., in Figure 1, we can see that TAP_2 handles the traffic of its own client M_1 and also the traffic of M_2, M_3 and M_4).

Therefore, we represent the mesh network as a directed graph[4] where HS and the TAPs are the vertices (i.e., the set of vertices is $\mathcal{V}= \{HS, TAP_i, i = 1..n\}$). A link (i,j) between TAP_i and TAP_j means that these two TAPs are within transmission range of each other. We use the index $i = 0$ or $j = 0$ to refer to a link from HS or to HS, respectively (see Figure 2). The set of mobile clients is denoted by $\mathcal{M}= \{M_i, i = 1..m\}$.

The link (i,j) can be (i) a *communication link* (represented with solid lines in Figure 2), i.e., a link that is intentionally used to send the traffic to or from HS, or (ii) an *interference link* (represented with dashed lines in Figure 2), i.e., a link that is unintentionally activated by neighboring TAPs.

A communication link is *upstream* if it is used to handle the traffic from the mobile clients to HS and *downstream* if it is used to handle the traffic from HS to the mobile clients.

We will denote by:

- $\mathcal{UL}$ the set of upstream communication links.
- $\mathcal{DL}$ the set of downstream communication links.
- $\mathcal{IL}$ the set of interference links.

The *load* $l_{i,j}$ of a link (i,j) is defined as the number of mobile clients that are using it to transmit their traffic to or

---

[3] In the parking lot scenario, many cars attempt to leave a parking lot simultaneously using a single exit. Details can be found in [8].

[4] The mesh network is constructed as a tree (see the communication links in Figure 2). However, the existence of interference links between the TAPs leads to a graph.

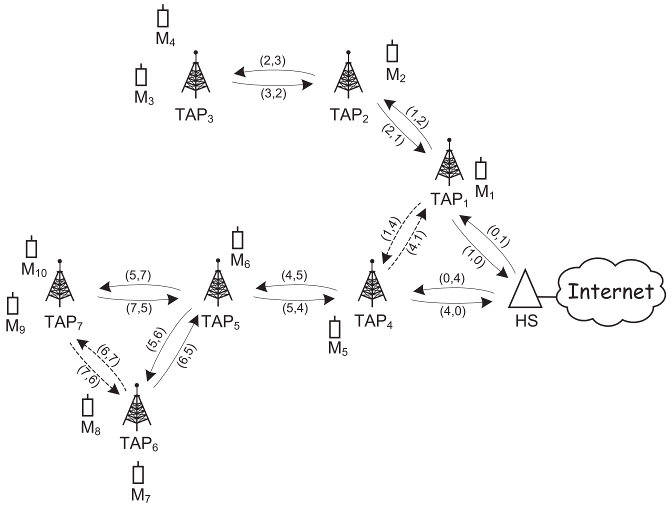

Fig. 2. The up-stream and down-stream communication links and the interference links corresponding to the mesh network presented in Figure 1: $\mathcal{UL}$={(1,0);(2,1);(3,2);(4,0);(5,4);(6,5);(7,5)}, $\mathcal{DL}$={(0,1);(1,2);(2,3);(0,4);(5,4);(5,6);(5,7)}, and $\mathcal{IL}$={(1,4);(4,1);(6,7);(7,6)}.

from the Internet. Link (i, j) is said to be *active* if $l_{i,j} > 0$. We therefore define $\mathcal{AUL}$, $\mathcal{ADL}$ and $\mathcal{AIL}$, respectively, as the sets of *active* up-stream communication links, down-stream communication links and interference links.

3.1 Assumptions

We assume, for the sake of simplicity:

- HS and all the TAPs in the mesh network are under the control of a single operator.
- No mobile client is directly served by HS; HS plays only the role of relay for the TAPs to and from the Internet.
- The mobile clients are paying the same flat rate and therefore the available throughput should be shared fairly among the mobile clients simultaneously connected to the mesh network.
- The topology of the mesh network is fixed and known by HS and the TAPs.
- HS and all the TAPs use omnidirectional antennas.
- The *up-stream traffic* (i.e., traffic from the TAPs to HS), the *down-stream traffic* (i.e., traffic from HS to the TAPs) and the control messages are sent using three orthogonal channels.
- A fourth orthogonal channel is used for the AP-MN communication.
- All communication links in the mesh network have the same capacity C.
- The mobile clients are sending and receiving data at saturation rate, i.e., there are always packets to be transmitted from the mobile clients to HS and vice versa.

We discuss the way to relax some of these assumptions in Section 6.

3.2 A Fair Scheduling for WMNs

As already stated in Section 1, we propose a collision-free scheduling algorithm that ensures per-client fairness and, at the same time, optimizes the bandwidth utilization in the WMN.

Given that the upstream traffic and the downstream traffic are sent over two orthogonal channels, we define one scheduling for each kind of traffic; in the *upstream scheduling* (respectively *downstream scheduling*), we specify the transmission rights assignment of upstream links (respectively downstream links). We use the symbols $\mathcal{L}$ and $A\mathcal{L}$ to refer to $\mathcal{UL}$ and $A\mathcal{UL}$, respectively, when describing the upstream scheduling, and to $\mathcal{DL}$ and $A\mathcal{DL}$, respectively, when describing the downstream scheduling.

3.2.1 The Per-Client Fairness

Our solution is a collision-free scheduling algorithm that assigns transmission rights to the network links. We call *cycle* of the schedule the time needed to activate all the upstream (respectively downstream) links in the WMN according to our upstream (respectively downstream) scheduling algorithm. The cycle keeps repeating until the next scheduling update (see Section 4 for the details about the scheduling update).

Let ρ_a be the throughput of a client M_a that is connected to TAP_α. The flow f_a of client M_a traverses route r_a (the route from TAP_α to HS, and vice versa), with a number of hops h_a. Let also $t^{(i,j)}$ and $t^{(i,j)}_{f_a}$ be the duration of the activation of link (i, j) during the cycle and the amount of time dedicated to flow f_a on link (i, j), respectively.

Given that all links have capacity C, the per-client fairness is respected if we have:

$$\rho_a = \rho_b, \qquad \forall a, b \in \mathcal{M} \tag{1}$$

where ρ_a can be computed as

$$\rho_a = \min_{(i,j)\in r_a} \frac{t^{(i,j)}_{f_a}}{cycle} \cdot C$$

The *network throughput* Γ can be computed as

$$\Gamma = \sum_{i=1}^{m} \rho_i \tag{2}$$

and in order to maximize Γ, we need to have

$$t^{(i,j)}_{f_a} = t^{(x,y)}_{f_a}, \qquad \forall (i, j), (x, y) \in r_a$$

Therefore, the amount of time dedicated to flow f_a should be the same for all the links on r_a; we denote this duration by t_{f_a}. The per-client fairness condition (1) gives

$$t_{f_a} = t_{f_b}, \qquad \forall a, b \in \mathcal{M}$$

Therefore, the amount of time dedicated to each flow on each link should be the same; we call this time a *time slot*, we denote it by ts and, without loss of generality, we consider it as the time unit.

Let us call T the (integer) number of time slots in the cycle, expressed in this unit. Given the assumption that the clients always have packets to send or to receive, each client sends (or receives) the same throughput ρ:

$$\rho_a = \rho = \frac{C}{T}, \qquad \forall a \in \mathcal{M} \tag{3}$$

Note that the per-client fairness leads to a cycle where each flow carries exactly the amount of data that can be sent during one time slot ts.

An Example of Fair Scheduling: A simple collision-free scheduling algorithm that fulfills the per-client fairness requirement is the TDMA scheduling (see Figure 3), where each link (i, j) is activated during a period of time $l_{i,j} \cdot ts$; one time slot ts dedicated to each of the clients that are simultaneously using (i, j).

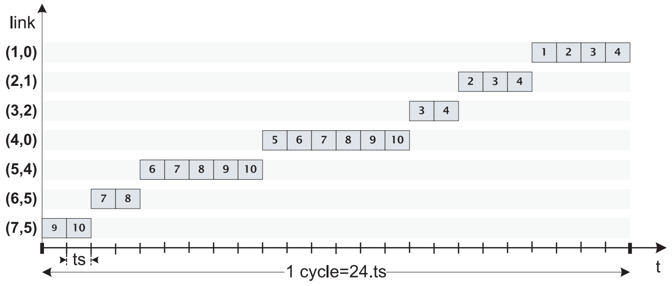

Fig. 3. A possible TDMA scheduling (without spatial reuse) for the upstream links in the mesh network of Figure 2. Each number represents the client whose flow is currently using the considered link. For this scheduling example, we have $T = 24$.

For this scheduling example, we have $T = \sum_{(i,j)\in\mathcal{L}} l_{i,j}$ and Equation (3) is fulfilled, which means that this TDMA scheduling guarantees the per-client fairness. Furthermore, as no two links are activated at the same time, this scheduling is also collision-free. However, it does not guarantee an optimal utilization of the network resources. Indeed, by allowing *spatial reuse* [18], i.e., simultaneous activation of non-contending links, we can optimize the bandwidth utilization. An example of spatial reuse is provided in Figure 4.

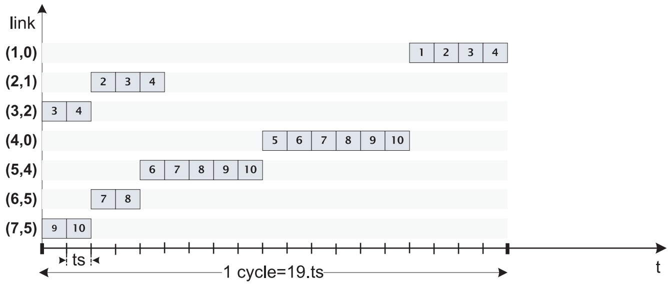

Fig. 4. A possible TDMA scheduling with spatial reuse for the upstream links in the mesh network of Figure 2. For this scheduling example, we have $T = 19$.

3.2.2 *Optimization of Bandwidth Utilization*

As explained in the previous subsection, we intend to devise a fair scheduling that also optimizes the bandwidth utilization, and therefore maximizes the value of the network throughput:

$$\Gamma = m \cdot \frac{C}{T} \tag{4}$$

To maximize Γ, we need to minimize T, while respecting the fairness condition (3).

Given that a TDMA scheduling without spatial reuse is always possible, we can consider the value $T_{max} = \sum_{(i,j)\in\mathcal{L}} l_{i,j}$ as the higher bound for T; it corresponds to the case where all the links in the WMN mutually contend. T's lower bound depends on the topology of the network and the position of the mobile clients, and corresponds to an optimal spatial reuse. We give a possible approximation for this lower bound in Section 4.

4 DETAILS OF THE SOLUTION

As already mentioned in Subsection 3.2.1, we present, in this Section, the details of FS, a fair scheduling that approximates the optimal spatial reuse. We compare in Subsection 5.3 the solution obtained using FS to the optimal solution.

Our solution is comprised of three main components: *Construction of the Compatibility Matrix*, *Construction of the cliques* and *Definition of the Fair Scheduling* and is executed by HS and all the TAPs if a given TAP_i detects one of the following events:

- The *join* event: This event corresponds to a connection of a client M_j to TAP_i.
- The *leave* event: This event occurs when M_j closes its connection with TAP_i.
- The *handoff* event: This event occurs when a given client M_j moves from the coverage of TAP_j to the coverage of TAP_i.

The rationale of the solution is the following: We first construct the *compatibility matrix*, which contains the links that can be activated at the same time (see Subsection 4.1). Then, we define different possible *cliques*, i.e., sets of links that can *all* be simultaneously activated (see Subsection 4.2). Finally, we define a combination of cliques and we use it as a new fair scheduling (see Subsection 4.3).

4.1 Construction of the Compatibility Matrix

Our concept of compatibility matrix is similar to the one used in [18]:

$$CM = [cm_{x,y}], \qquad 1 \le x, y \le |\mathcal{AL}|$$

where $|\mathcal{AL}|$ is the set of active links[5].

We assume that all links in $\mathcal{AL}$ are sorted according to a certain order; thus, the x'th row and column in CM correspond to the x'th link in $\mathcal{AL}$.

Let us assume that the x'th and y'th positions in $\mathcal{AL}$ correspond to links (i_1, j_1) and (i_2, j_2), respectively. Therefore, we have:

$$cm_{x,y} = \begin{cases} 0 & \text{if } x = y \\ 0 & \text{if links } (i_1, j_1) \text{ and } (i_2, j_2) \text{ mutually} \\ & \text{contend} \\ 1 & \text{otherwise} \end{cases}$$

---

[5] As already mentioned in Subsection 3.2.1, we define a scheduling for upstream links and another scheduling for downstream links and, therefore, we need a compatibility matrix (CM) for each scheduling. The construction of both CMs being symmetrical, we use the symbol $\mathcal{AL}$ to refer to $\mathcal{AUL}$ or $\mathcal{ADL}$ for the construction of the upstream or downstream CM, respectively.

The compatibility matrix will reflect the fact that a given TAP can only:

- transmit or receive,
- receive from one TAP at a time, and
- send to one TAP at a time.

We also need to make sure that all TAPs that could interfere with a sending or a receiving TAP remain silent.

For the WMN of Figure 2, we have the following upstream compatibility matrix:

$$\text{CM} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad (5)$$

where the rows correspond to links $(1,0)$, $(2,1)$, $(3,2)$, $(4,0)$, $(5,4)$, $(6,5)$ and $(7,5)$, respectively.

The compatibility matrix can be represented as a graph $\mathcal{G}$, which we call the *compatibility graph* and where the vertices correspond to the links in $A\mathcal{L}$. If the x'th and y'th positions in $A\mathcal{L}$ correspond to links (i_1, j_1) and (i_2, j_2) respectively, there is an edge between vertices (i_1, j_1) and (i_2, j_2) if $cm_{x,y} = 1$.

The compatibility graph corresponding to the compatibility matrix (5) is represented in Figure 5.

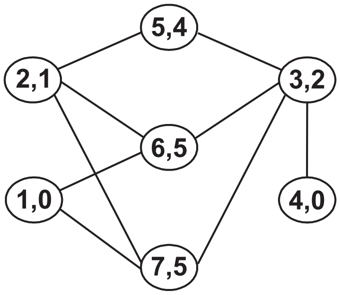

Fig. 5. The up-stream compatibility graph of the mesh network presented in Figure 2.

4.2 Construction of the Cliques

Given the compatibility matrix CM, we can construct the set of all possible cliques for the corresponding network, a *clique* being a set of links that can *all* be enabled at the same time. We denote by k the cardinality of the clique ($k > 0$).

Several cliques of cardinality k can exist for the same compatibility matrix CM; we denote by Cl_k^a the a-th clique of cardinality k and by $(\beta,\gamma)_k^a$ the most loaded link in Cl_k^a:

$$l_{(\beta,\gamma)_k^a} = \max_{(i,j) \in Cl_k^a} l_{i,j}$$

.

In the compatibility graph $\mathcal{G}$, Cl_k^a corresponds to:

- The vertex (i,j) if $Cl_k^a = Cl_1^a = \{(i,j)\}$,
- The arc between vertices (i_1, j_1) and (i_2, j_2) if $Cl_k^a = Cl_2^a = \{(i_1, j_1), (i_2, j_2)\}$), and
- A clique (i.e., a complete subgraph [Ref Skiena90]) composed of the vertices that are in Cl_k^a if $k > 2$.

All the links in Cl_k^a can be activated simultaneously. We denote by d_k^a the number of time slots that are reserved, on the cycle, for Cl_k^a; d_k^a corresponds to the number of time slots that are required to transmit the traffic of the most loaded link in the clique:

$$d_k^a = l_{(\beta,\gamma)_k^a}$$

Each link (i,j) in Cl_k^a is activated during $l_{i,j}$ time slots and is idle during the $d_k^a - l_{i,j}$ remaining time slots[6]. Therefore, the clique Cl_k^a generates a *gain* $g(Cl_k^a)$ where:

$$g(Cl_k^a) = \Big(\sum_{(i,j) \in Cl_k^a} l_{i,j} \Big) - d_k^a$$

The value of $g(Cl_k^a)$ corresponds to the cumulative number of slots that would have been necessary to transmit the traffic on each of the links in Cl_k^a, other than $(\beta,\gamma)_k^a$, separately (i.e., a TDMA scheduling without spatial reuse).

In this phase, we search for *all* possible cliques corresponding to the compatibility matrix CM and we define, for each of these cliques, the gain $g(Cl_k^a)$. We discuss the complexity of the clique enumeration in Subsection 6.2.

The choice of a combination of cliques defines the new scheduling (see Subsection 4.3).

4.3 Definition of the Fair Scheduling

We define a scheduling s as a set of cliques that fulfills the following two conditions:

$$\bigcup_{Cl \in s} Cl = A\mathcal{L} \qquad (6)$$

and

$$Cl1 \cap Cl2 = \emptyset, \qquad \forall Cl1, Cl2 \in s \qquad (7)$$

Condition (6) guarantees that all the active links (i.e., links with a load $l_{i,j} > 0$) are activated at least once during the cycle, whereas Condition (7) guarantees that each of these links is activated exactly once (see Figure 6). We discuss our motivation for requiring Condition (7) in Subsection 6.3.

Based on the list of cliques we obtained during the *Clique Construction* phase, we can define the set $\mathcal{S}$ of all possible schedulings. To each element s in $\mathcal{S}$ corresponds a cycle duration T_s and a gain g_s, where

$$T_s = \sum_{Cl_k^a \in s} d_k^a$$

and

$$g_s = \sum_{Cl_k^a \in s} g(Cl_k^a)$$

Given that our goal is to propose a fair scheduling that minimizes the duration T of the cycle (i.e., maximizes the

---

[6] We discuss the possibility of relaxing this assumption in Subsection 6.3.

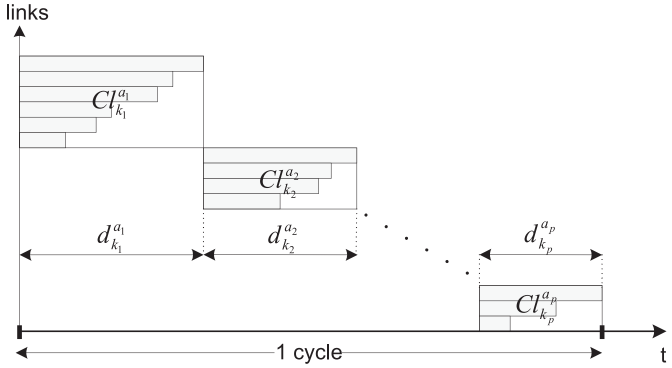

Fig. 6. Conditions (6) and (7) guarantee that, during the cycle, each active link (i, j) is activated exactly once; the activation duration for (i, j) is $l_{i,j}$ time slots.

spatial reuse), we have to find the optimal[7] scheduling s^* such that:

$$T_{s^*} = \min_{s \in \mathcal{S}} T_s$$

However, finding the optimal scheduling would require considering all possible clique combinations fulfilling Conditions (6) and (7)). In order to reduce the complexity of such an exhaustive search, we propose FS, a fair scheduling algorithm that approximates the optimal solution s^*; we denote by $\hat{s}$ the scheduling provided by our algorithm FS and we discuss the difference between s^* and $\hat{s}$ in Subsection 5.3.

The idea behind FS is based on the intuition that minimizing the duration T_s of the cycle is equivalent to maximizing the gain g_s. The rationale of FS is the following:

1) First, we search for the clique $\widehat{Cl}_1$ that has the highest gain (this clique is likely to be a maximal clique).
2) we set $\hat{s} = \{\widehat{Cl}_1\}$.
3) While Condition 6 is not yet satisfied
    - We search for the clique $\widehat{Cl}_i$ that has the highest gain, among the cliques that do not intersect with the members of $\hat{s}$ (for Condition 7 to be satisfied).
    - We add $\widehat{Cl}_i$ to $\hat{s}$.

We evaluate the efficiency of our algorithm in Section 5.

5 EVALUATION OF THE SOLUTION

In this Section, we first prove that our solution indeed leads to a fair collision-free scheduling. Then, we study, by means of simulations, the efficiency of our fair scheduling. Finally, we discuss the optimality of our fair scheduling FS.

5.1 The Fair Collision-free Scheduling Proof

In this Subsection, we consider the scheduling $\hat{s}$ that is given by our fair scheduling algorithm FS.

Proposition 1: $\hat{s}$ is a fair scheduling.

Proof: Conditions (6) and (7) guarantee that, during the cycle, each active link (i, j) is activated exactly once during $l_{i,j}$ time slots (see Subsection 4.2). Therefore, each end-to-end flow is activated during one time slot ts, which allows each flow client to send (or receive) the same amount of data $ts \cdot C$ and shows that $\hat{s}$ is a fair scheduling. □

Proposition 2: $\hat{s}$ is a collision-free scheduling.

Proof: The scheduling $\hat{s}$ being a *disjoint union* of cliques (i.e., a union of cliques whose members are pairwise disjoint), two links that are in two different cliques in $\hat{s}$ never contend as they are activated at two different time periods (see Figure 6). Furthermore, a clique is, by definition, a set of non-contending links. Therefore, $\hat{s}$ is a collision-free scheduling. □

5.2 Efficiency of our Fair Scheduling

5.2.1 Simulations Setup

We used the Matlab simulator [11] to implement the three components of our solution (*Compatibility Matrix Construction*, *Cliques Construction* and *Scheduling Update*).

We conducted two sets of simulations. In the first set, we consider a one-dimensional mesh network, with 10, 15, 20 and 25 TAPs, respectively (see Figure 7) whereas in the second set, we consider the two-dimensional mesh network composed by 8, 16, 24 and 32 TAPs, respectively (the 8, 16, 24 and 32 first TAPs of the topology introduced by Figure 8).

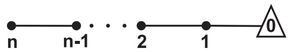

Fig. 7. The topology of the one-dimensional mesh network. We consider 10, 15, 20 and 25 TAPs, respectively.

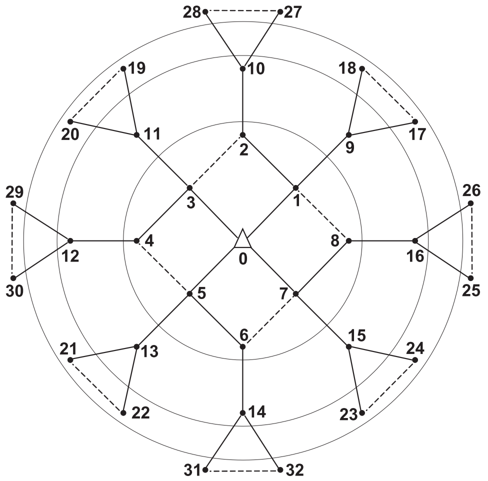

Fig. 8. The topology of the two-dimensional mesh network. We consider a network of 8, 16, 24 and 32 TAPs, respectively.

For each of these network configurations, the number of clients is twice the number of TAPs ($m = 2 \cdot n$) and we consider three different distributions of the clients in the network:

- The uniform distribution: Exactly 2 clients are connected to each TAP.
- The peripheral distribution: The clients are more numerous on the periphery[8] of the network than in the center.

---

[7] If several optimal schedulings exist, we can choose one at random.

[8] We consider the TAPs that are the furthest from the wired hot spot (HS) as the periphery and the TAPs that are the closest to HS as the center.

- The central distribution: The clients are more numerous in the center of the network than on the periphery.

For each network topology, we run one simulation for each combination of network size and client distribution. We compare the duration of the scheduling cycle T we obtain using our solution to the cycle we obtain when there is no spatial reuse ($T_{max} = \sum_{(i,j)\in\mathcal{L}} l_{i,j}$) which, as shown in Subsection 3.2.2, represents the upper bound for T.

5.2.2 Simulation Results

The simulation results for the one-dimensional mesh network and for the two-dimensional mesh network are plotted in Figures 9 and 10, respectively.

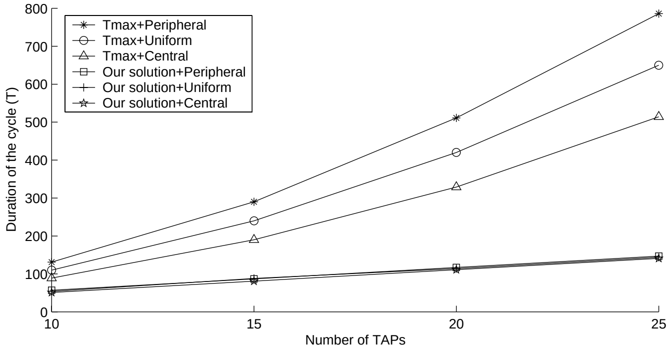

Fig. 9. Simulation results for the one-dimensional mesh network: Our solution leads to a much lower T/m ratio compared to the TDMA scheduling without spatial reuse.

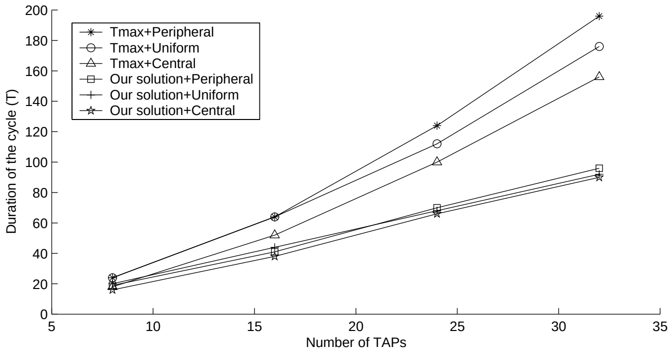

Fig. 10. Simulation results for the two-dimensional mesh network: Our solution leads to a much lower T/m ratio compared to the TDMA scheduling without spatial reuse.

We can clearly see that the duration of the scheduling cycle increases with the size of the network for all scenarios.

Our scheduling algorithm performs much better than the TDMA scheduling without spatial reuse, and leads to a much lower T/m ratio. Indeed, the duration of the cycle increases almost linearly for our solution whereas it increases exponentially for the TDMA scheduling without spatial reuse.

Furthermore, the simulation results show that our fair scheduling attenuates the variations introduced by the different client distributions.

5.3 Optimality of our Fair Scheduling

As stated in Subsection 4.3, the scheduling $\hat{s}$ we obtain by using our fair scheduling FS is an approximation of the optimal scheduling s^*.

In order to compare $\hat{s}$ to s^*, we implemented the algorithm that searches for the optimal solution. This algorithm enumerating all possible schedulings, and returns the scheduling s^* s.t. $T_{s^*} = \min_{s\in\mathcal{S}} T_s$. We ran one simulation for each of the scenarios described in Subsection 5.2 and we compared the results with the results we obtained by running our fair scheduling FS. The simulation results showed that resulting schedulings $\hat{s}$ to s^* are identical for all the scenarios we considered, which means that our FS algorithm approximated very well the optimal solution.

6 DISCUSSION

In this Section, we discuss several aspects of our solution and we relax some of the assumptions of Subsection 3.1.

6.1 Topology Discovery

HS launches the topology discovery operation at the initialization phase (i.e., when the mesh network is first deployed) or when the network topology is modified (e.g., a TAP is added or removed); to do so, HS can use an ad hoc routing protocol[9] (e.g., DSR [15], AODV [22], ...). Upon receipt of (i) all requested routes and (ii) the list of neighbors of each AP in the network, HS constructs the network topology (including the interference graph) and informs all the TAPs about it. All the messages used to construct the network topology are exchanged over the control channel.

Given that the mesh network is under the control of a single operator, we can assume, without loss of generality, that all links in the mesh network are stable over time.

6.2 Complexity of the Solution

With the *Compatibility Matrix Construction* phase and the FS algorithm being polynomial, the complexity of our fair scheduling mechanism depends on the complexity of the *Clique Construction* phase. Indeed, during the *Clique Construction* phase, we enumerate all the possible cliques corresponding to the compatibility matrix CM; the clique enumeration problem is proven to be NP-hard [16, 9].

However, the relatively small size of the WMN and the utilization of optimized algorithms such as [7] or [23] can make the clique enumeration phase, and therefore our scheduling solution, much more efficient and fast.

As future work, we intend to evaluate the exact complexity of our solution and to define, under different mobility assumptions, traffic conditions and client distributions, the frequency at which the scheduling is updated and the time required for this updating.

[9] To secure the topology discovery phase, HS can use a secure routing protocol such as [13], [19] or [21].

6.3 Link Activation Duration

In Subsection 4.2, we state that, if the clique Cl_k^a is chosen for the scheduling, each link (i,j) in Cl_k^a is activated during $l_{i,j}$ time slots and is idle during the $d_k^a - l_{i,j}$ remaining time slots. The reason behind this decision in the following: Let us assume that $l_{i,j} < d_k^a$ and that the link (i,j) is activated during $l'_{i,j}$ where $l'_{i,j} > l_{i,j}$. TAP_j will therefore receive an amount $(l'_{i,j} - l_{i,j}) \cdot C$ of extra data. If this extra data were transmitted from TAP to TAP, to or from a given client, the fairness requirement would be violated. Therefore, TAP_j has to drop this extra data or to store it locally; the first option may lead to data loss (and retransmission), whereas the second may lead to storage problems at the TAPs. Therefore, for the sake of simplicity, we assume that the link (i,j) is activated during exactly $l_{i,j}$ time slots.

The same reasoning motivates Condition (7); if a link (i,j) were in two different cliques, it would be activated twice during the same cycle, which may violate the fairness requirement.

However, as future work, we intend to relax these two assumptions and to define more sophisticated scheduling algorithms. Indeed, the scheduling presented in Figure 4 is a fair (and optimal) collision-free scheduling that does not fulfill Condition (7); these two assumptions are excluding schedulings such as the one presented in Figure 4, which may lead to suboptimal solutions.

6.4 Capacity Reuse

If a client that is connected to the WMN remains idle for a long period of time, we should update the scheduling, otherwise we will have poor bandwidth utilization.

Therefore, when a client M_i is idle during a given timeout T_o (T_o can be expressed in number of cycles), the access point TAP_i to which this client in connected assumes that the client is disconnected and informs HS and all the other TAPs about this disconnection; HS and all the TAPs update their schedulings accordingly. When M_i is active again, TAP_i considers this activation as a new connection, informs HS and the other TAPs about it and they all update the scheduling accordingly.

7 CONCLUSION

In this paper, we propose a scheduling mechanism for WMNs that ensures per-client fairness and optimizes the bandwidth utilization. The solution is collision-free and assigns transmission rights to the links in the WMN while maximizing the *spatial reuse*.

We evaluated our scheduling mechanism by means of simulations and we have shown that it is efficient and, for some aspects of the solution, optimal.

As future work, we intend to consider the different aspects we discussed in Section 6. We also want to extend our solution to cases where the TAPs are equipped with sector antennas and there are different link capacities in the WMN.

ACKNOWLEDGMENTS

The authors would like to thank Jun Luo, Jacques Panchard and Maxim Raya for their useful feedback. Thanks also to Mario Čagalj and Imad Aad for helpful discussions and comments.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: A Survey. *Computer Networks Journal (Elsevier)*, 47(4), 2005.

[2] P. Bahl, A. Balachandran, A. Miu, W. Russell, G. Voelker, and Y.M. Wang. PAWNs: Satisfying the Need for Ubiquitous Connectivity and Location Services. *IEEE Personal Communications Magazine (PCS)*, 9(1), 2002.

[3] P. Bahl, A. Balachandran, and S. Venkatachary. Secure Wireless Internet Access in Public Places. In *Proceedings of the IEEE Conference on Communications*, 2001.

[4] Y. Bejerano, S.-J. Han, and L. Li. Fairness and Load Balancing in Wireless LANs Using Association Control. In *Proceedings of MobiCom*, 2004.

[5] P. Bjorklund, P. Varbrand, and D. Yuan. Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach. In *Proceedings of INFOCOM*, 2003.

[6] R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Communications Magazine*, March 2005.

[7] N. Chiba and T. Nashizeki. Arboricity and Subgraph Listing Algorithms. *SIAM J. Comput*, 14, 1985.

[8] V. Gambiroza, B. Sadeghi, and E. Knightly. End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks. In *Proceedings of MobiCom*, 2004.

[9] M. Garey and D. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. *New York: W. H. Freeman*, 1983.

[10] J. Gronkvist. Assignment Methods for Spatial Reuse TDMA. In *Proceedings of MobiHOC*, 2000.

[11] Matlab Simulator.
http://www.mathworks.com/.

[12] MeshNetworks Granted Experimental License by the FCC to Test and Deploy Mobile Mesh 4.9Ghz Solutions.
http://www.tmcnet.com/usubmit/2004/Sep/1075816.htm.

[13] Y. C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proceedings of Mobicom*, 2002.

[14] D. Jackson. Motorola Announces First Mesh Deployment. http://mrtmag.com/news/motorola_mesh_deployment_022805/.

[15] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski and Korth, editors, *Mobile Computing*. Kluwer Academic Publishers, 1996.

[16] R. Karp. Reducibility Among Combinatorial Problems. *Complexity of Computer Computations*, 85-103, 1972.

[17] Victoria W. Kipp. The battle of NIMBY. PRIMEDIA Business Magazines & Media Inc, 2002.

[18] S. Nelson and L. Kleinrock. Spatial TDMA: A Collision-Free Multihop Channel Access Protocol. *IEEE Transactions on Commnuications*, 33(9), 1985.

[19] P. Papadimitratos and Z. J. Haas. Secure Routing for Mobile Ad Hoc Networks. In *Proceedings of CNDS*, 2002.

[20] S. Rupley. A Moveable Mesh. http://www.pcmag.com/article2/0,1895,1641379,00.asp.

[21] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer. An Authenticated Routing Protocol for Secure Ad hoc Networks.. In *Journal on Selected Areas in Communications special issue on Wireless Ad hoc Networks*, March 2005.

[22] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. IETF Internet draft, draft-ietf-manet-aodv-10.txt (Work in Progress). March 2002.

[23] E. Tomita, A. Tanaka, and H. Takahashi. The Worst-case Time Complexity for Finding all the Cliques. Technical report, UEC-TR-CI, 1988.

[24] P. Varbrand and D. Yuan. Maximal Throughput of Spatial TDMA in Ad Hoc Networks. In *White paper, Sept.*, 2003.