Scheduling and Dynamic Relocation for IEEE 802.11s Mesh Deterministic Access

C. Cicconetti, L. Lenzini, E. Mingozzi
Dipartimento di Ingegneria dell'Informazione – University of Pisa
Via Diotisalvi, 2 – 56122 Pisa, ITALY
{c.cicconetti, l.lenzini, e.mingozzi}@iet.unipi.it

Abstract—**Deployment of Wireless Mesh Networks (WMNs) is becoming increasingly popular due to the low-impact and low-cost features of wireless devices. This is especially true for WMNs based on IEEE 802.11 which, however, does not include native support for multi-hop relaying. This gap is being filled by the Task Group 's' of the IEEE 802.11 which has recently published an amendment in order to add mesh functions to the popular standard IEEE 802.11. Among the enhancements proposed, Mesh Deterministic Access (MDA) allows mesh routers to negotiate periodic collision-free transmission opportunities, called MDAOPs, to the Medium Access Control (MAC) layer on a hop-by-hop manner. Control messages are exchanged to advertise the reserved MDAOPs in the two-hop neighborhood. MDA lays the foundations for enabling QoS provisioning functions in IEEE 802.11s WMNs, such as end-to-end bandwidth reservation, call admission control, and traffic engineering. In this paper we study the problem of scheduling MDAOPs, which is left unspecified by the standard, and propose two baseline algorithms. Furthermore, we provide evidence that performance under MDA can be significantly degraded by the unknown interference of traffic flows outside the two-hop neighborhood. A dynamic relocation procedure is proposed in order to combat this phenomenon, thus providing traffic flows established in the WMN with stable performance. This procedure does not need any modifications to the standard MDA procedure. To the best of our knowledge, this is the first study that tackles this problem in the context of IEEE 802.11s. The effectiveness of the proposed algorithms is evaluated by means of a packet-level simulation.**

Wireless Mesh Networks; IEEE 802.11s; Quality of Service; Scheduling; Mesh Determistic Access

I. INTRODUCTION

The interest in Wireless Mesh Networks (WMNs) is rapidly increasing due to the growing potential of its recently envisaged applications [3]. In a WMN, terminals are arranged in a two-tier architecture [1]: client and backbone. The latter consists of fixed Access Points (APs) and relays connected in a wireless multi-hop manner. Each AP provides a set of mobile terminals with intra-WMN or Internet access. Unlike mobile terminals, backbone devices have an unlimited power supply, high computational capabilities, and are usually managed by a single network provider.

Despite the vast diffusion of proprietary solutions and the huge amount of research studies, Medium Access Layer (MAC) protocols, explicitly intended for WMNs, have not been standardized so far. The only exception is the mesh mode of IEEE 802.16, which, however, received far less approval from industry and academia than its point-to-multipoint counterpart. For this reason the working group 802.11 of the IEEE has established the Task Group 's' in July 2004 with the goal of extending IEEE 802.11 [12] so as to provide mesh networking support. This standard revision is expected to be completed in 2009. Therefore, the most recently released stable draft version D1.08 as of this writing [11] is used in this work as reference. An IEEE 802.11s consists of three types of devices: Mesh Points (MPs) which form the wireless backhaul, Mesh Access Points (MAPs) which act as gateways to the WMN for the mesh clients, and Mesh Portals (MPPs) which interface the WMN with other IEEE 802 LAN segments. It is typically assumed that MPPs provide access to the Internet. Mesh clients are not required to be aware of the presence of the WMN and will simply associate to the best MAP sensed. Since both MAP and MPP entities are co-located with devices implementing MP functions, in the following we refer to MPs as *nodes* regardless of whether they also implement MAP/MPP functions or not.

In addition to routing and path selection, several optional mechanisms have been added as MAC functions to exploit the inherent properties of WMN networks, including the Mesh Deterministic Access (MDA) function, the intra-mesh congestion control mechanism, and the mesh power management. In this work, we focus only on MDA, which allows any node to reserve a time interval, called MDA opportunity (MDAOP), for transmitting data to a neighbor in a periodic manner. The base period is the Mesh DTIM Interval which is enforced by means of the mesh beaconing and synchronization procedure. The latter is mandatory for all nodes and refines the infrastructure beaconing procedures of IEEE 802.11. The transmitting and receiving nodes then advertise the MDAOP reservation to their neighbors, which will re-broadcast it to ensure that all the nodes in the two-hop neighborhood become aware of the MDAOP. MDA appears to be a viable and promising solution for implementing advanced Quality of Service (QoS) functions in an IEEE 802.11s-based WMN, because it provides nodes with a means to access the channel in controlled conditions, with virtually no collisions from neighboring nodes.

However, although the draft IEEE 802.11s provides the means to setup MDAOPs, it does not provide any policy, either informative or mandatory, for positioning the MDAOP within the Mesh DTIM Interval. Moreover, because of how MDAOP reservation messages are propagated in the WMN, MDA im-

This work has been carried out within the framework of the NADIR project, funded by the Italian Ministry of University and Research.

plicitly assumes that nodes outside the two-hop neighborhood of a link do not interfere with it, even though transmissions take place at the same time. Since in a real WMN this is not the general case [21], severe performance degradation (e.g. in terms of the packet loss rate) could be experienced in practice by certain traffic flows.

The objective of this paper is to assess the performance of MDA and investigate its potential limits using realistic network assumptions. Specifically, we define two baseline algorithms to schedule MDAOPs, whose performance, in terms of the network capacity, is evaluated via packet level simulations. Furthermore, we show that performance may be significantly affected by interfering transmissions from outside the two-hop neighborhood. To recover from this degradation, we propose following a procedure which reactively triggers the dynamic relocation of the interfered MDAOP.

The rest of the paper is organized as follows. In Section II we describe the MDA function of the IEEE 802.11s and introduce the terminology used in the paper. In section III we then describe the algorithms for MDAOP scheduling and dynamic relocation, which are then evaluated in Section IV. The related work is covered in Section V. We conclude the paper with Section VI which also discusses future work directions.

II. IEEE 802.11s MDA

In this section we describe the MDA function. The interested reader can find a detailed survey of the IEEE 802.11s draft in [10]. For the ease of notation, the term *nodes* will be used to refer to IEEE 802.11s-enabled MPs. Furthermore, in order to simplify our description, we assume that all the nodes support MDA and that the network is in a steady state, i.e. Mesh Links (*links* hereafter) are not established or torn down for a long period of time. In fact, as specified by IEEE 802.11s, each node keeps an up-to-date list of links which connect themselves to their neighbors, using the Mesh Peer Link Management protocol. If two nodes do not share a link but have at least one common neighbor, we call them two-hop neighbors. Finally, since in MDA the minimum time allocation unit is 32 µs, we call this interval a *slot*.

MDA allows any two neighbors to negotiate a periodic time interval, called MDAOP, during which channel access is performed with lower contention than during non-MDA intervals. Specifically, access to the medium in IEEE 802.11 is based on Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), whose basic parameters are: a minimum contention window (CW_{min}), a maximum contention window (CW_{max}), and an Arbitration Inter-Frame Space (AIFS). If the Enhanced Distributed Channel Access (EDCA) function is supported [2], these parameters can be tuned on a per-service class basis to provide differentiated service. As far as MDA is concerned, the standard specifies dedicated values of CW_{min}, CW_{max}, and AIFS which are intended to make channel access much more efficient. Nodes are in fact allowed to start transmitting earlier, after an idle period is detected, and with smaller backoff. This can be done since transmissions during an MDAOP are supposed to be protected by the MDA procedure for slot negotiation, which is described below. Note that MDA requires a means for all nodes to be synchronized at fixed time boundaries, which are called Delivery Traffic Indication Message (DTIM) intervals. This is achieved through the IEEE 802.11s procedure for beaconing and synchronization in the WMN.

When a node, called *requester*, wants to establish an MDAOP towards a neighbor, called *granter*, it sends an MDAOP Setup Request message (MREQ) which is a unicast MAC control frame and includes the following fields: the MDAOP *set ID* which is used to uniquely identify the MDAOP in combination with the MAC addresses of the requester and granter nodes; the *duration* of the MDAOP in slots, and the *offset* of the MDAOP, in slots with respect to the beginning of the DTIM interval. Optionally, the *periodicity* can also be included to specify how many MDAOPs are to be allocated within the DTIM interval. For instance, if the DTIM interval is 120 slots, the duration 10 and the offset 15, a periodicity of 3 implies that MDAOPs are allocated at the offsets 15, 55, and 95 with respect to the beginning of the DTIM interval. For simplicity, this opportunity is not considered in this work even though the algorithms proposed in Section III can be enhanced to exploit this feature in a straightforward manner. The granter node replies to the requester via a unicast MDA Setup Reply (MREP) message, which can be used to either accept the MDAOP indicated or reject it. In the latter case, an alternative MDAOP can be optionally provided as a hint to the requester in case it wishes to re-iterate the request.

Both MREQs and MREPs are unicast, thus they are only exchanged between the two nodes involved in the MDAOP setup. Information about the active MDAOPs is then disseminated through MDAOP Advertisement (MADV) messages which are broadcasted periodically by all nodes. MADVs include the *TX-RX times* and *interfering times* reports, which are two separate lists of MDAOPs, and the *Medium Access Fraction* (*MAF*). The TX-RX times set is the list of MDAOPs for which the advertising node is either a requester or a granter. The interfering times set is the list of MDAOPs for which one of the neighbors of the advertising node is either a requester or a granter, but itself is neither. This list can easily be kept up-to-date by copying the MDAOPs in the TX-RX times received by the neighbors. Finally, the MAF indicated by a node is the ratio between the sum of its TX-RX times and interfering times, to the DTIM interval. The amount of channel capacity reserved for MDA transmissions is upper bounded by the system parameter called *MAF limit* which must be enforced by all nodes when negotiating new MDAOPs.

The procedure for MDAOP negotiation is illustrated by means of a simple example. Figure 1 depicts part of an IEEE 802.11s WMN, and we focus on the three nodes A, B and C: B is a neighbor of both A and C which in turn are two-hop neighbors. Assume that there are already two served MDAOPs between C and one of its neighbors (which might be B or any other node not shown in Fig. 1), which are then advertised as TX-RX times by C, and another MDAOP that is advertised by

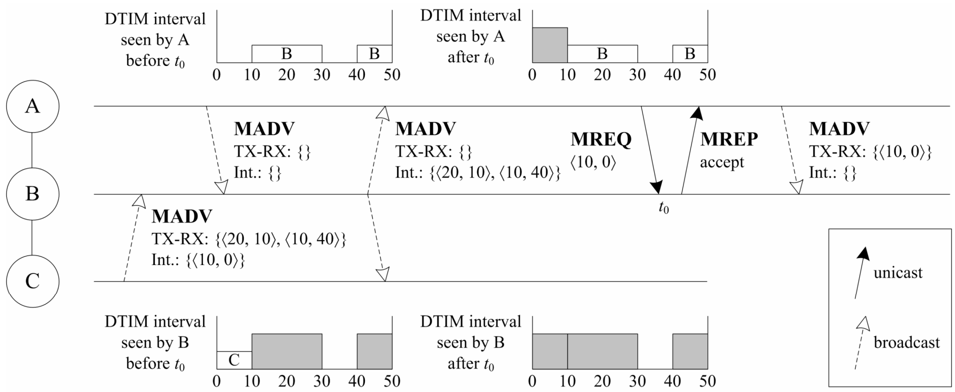


Figure 1. Example of the negotiation of an MDAOP from node A to node B. Notation $\langle x, y\rangle$ means that the MDAOP has duration x and offset y. The Set ID field is not reported. The DTIM interval is assumed to be 50 slots.

C in its interfering times. As can be seen, A does not know about MDAOP ⟨10, 0⟩ since it occurs outside its two-hop neighborhood. On the other hand, A is aware of MDAOPs ⟨20, 10⟩ and ⟨10, 40⟩, even though it is neither the granter nor the requester for them. Therefore, it is perfectly safe, from MDA's perspective, that A requests ⟨10, 0⟩ from B which thus accepts it at time t_0. From now on ⟨10, 0⟩ will be advertised by B among its TX-RX times, as reported in Fig. 1, which also shows the MADVs subsequent to time t_0.

We stress that MDA implicitly assumes that interference outside the two-hop neighborhood is negligible. In the example above, this means that transmissions from A to B in ⟨10, 0⟩ are assumed not to mutually interfere with those in overlapping slots that D knows about.

III. MDAOP SCHEDULING AND DYNAMIC RELOCATION

In the section above we described the MDA procedure specified by IEEE 802.11s. However, as already mentioned, there are some gaps that need to be filled in order to implement the MDA function. First, the procedure carried out by each node to select the location of requested MDAOPs is left unspecified in the standard. This is covered in the first part of this section. Afterwards, we point out that performance can be degraded when MDAOPs are allocated in partially overlapping slots by nodes outside the reciprocal two-hop neighborhoods. Therefore, we propose a mechanism, called *dynamic relocation*, to detect and solve this undesirable situation. Dynamic relocation does not need any modifications to the standard MDA procedures and is based on information readily available at the MAC layer of the nodes.

A. MDAOP Scheduling Algorithms

As described in Section II, each node collects TX-RX times and interfering times advertised in the MADVs broadcasted by its own neighbors. These reports are used to remove possible locations, i.e. sets of contiguous slots, from the portion of the DTIM interval available for requesting and granting MDAOPs. Let us define T_i as the set of locations advertised by node i as its TX-RX times, and I_i as the set of locations advertised as its interfering times:

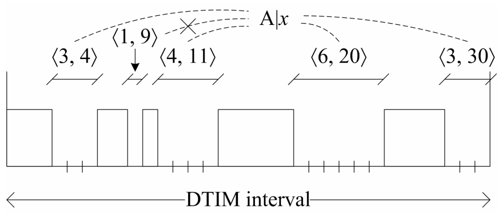
$$I_i = \bigcup\{T_j \,|\, j \in N_i\} \setminus T_i, \tag{1}$$

where N_i is the set of neighbors of node i. If we focus on a given node r, which requests an MDAOP from node g, the set $\bar{\mathcal{A}}$ of locations that are *not* available to be scheduled is then given by:

$$\bar{\mathcal{A}} = \left(\bigcup\{T_i \,|\, i \in N_r\}\right) \cup I_g. \tag{2}$$

Ideally the best MDAOP scheduling algorithm always finds a location for any MDAOP within the DTIM interval provided that the MAF limit is not exceeded. In fact, since MDAOPs are not prioritized in IEEE 802.11s, the key objective is to maximize the DTIM interval utilization. However, this goal might not be achieved in practice because of the following two reasons: (i) MDAOPs are set up and torn down dynamically over time so as to follow the distribution of user-generated traffic within the WMN, and (ii) the duration of MDAOPs is variable, depending on the characteristics of the traffic that is to be routed via the MDA path and on the physical rate of the links along it. Therefore, the MAF limit might not be achievable in practice due to MDAOP *fragmentation*. If we refer again to the example in Fig. 1, B cannot request a 20-slot MDAOP from A regardless of the MAF limit. However, if the MDAOP ⟨20, 10⟩ were located as ⟨20, 20⟩, this would have been possible.

The problem of placing MDAOPs within the DTIM interval resembles that of memory or file system management [18], which is well known in the operating systems literature. However, the existing approaches rely on the ability of reorganizing page allocation by moving memory chunks, though at some cost. This opportunity appears to be impractical in the context of MDAOP scheduling due to the need of renegotiating MDAOPs, which requires time and consumes channel capacity. We argue that MDAOPs should be relocated as few times as possible, and only if this is required, e.g. if MDA transmissions experience significant performance degradation or cannot take place at all.



free location / algorithm	⟨3, 4⟩	⟨1, 9⟩	⟨4, 11⟩	⟨6, 20⟩	⟨3, 30⟩
rnd-fit	✓	×	✓	✓	✓
best-fit	✓	×	×	×	✓

Figure 2. Free eligible (✓) and ineligible (×) locations with random-fit and best-fit, respectively, if the MDAOP size is 3 slots.

In the following, we propose two baseline algorithms for scheduling MDAOPs. Without loss of generality, we assume that the MAF limit is 1 so as to have a more concise explanation. Let $\mathcal{A}$ be the union of free locations of contiguous slots obtained by removing $\bar{\mathcal{A}}$ from the DTIM interval, and $\mathcal{A}|x$ be the subset of free locations large enough to contain an MDAOP of duration x:

$$\mathcal{A}|x = \{\langle d_i, o_i \rangle \in \mathcal{A} | d_i \geq x\}. \quad (3)$$

MDAOP scheduling consists of selecting an element of $\mathcal{A}|x$, which is then conveyed by the requester to the granter into the MREQ message.

The first algorithm we consider for MDAOP selection is the *best-fit* algorithm, which selects the smallest free location $\langle d_{best}, o_{best} \rangle$ which is able to contain an MDAOP of duration x, i.e.:

$$\langle d_{best}, o_{best} \rangle = \arg \min_{\langle d_i, o_i \rangle \in \mathcal{A}|x} \{d_i - x\}. \quad (4)$$

If there are many such locations, the tie is broken randomly. The rationale behind this choice is that, in general, having small gaps in the DTIM interval may lead to capacity wastage. In fact, such small available locations constrain the size of MDAOPs that could be actually allocated, even though the cumulative size of the gaps would suffice to place those MDAOPs. Therefore, having "compact" MDAOP allocation should allow for getting closer to the MAF limit, i.e. maximize utilization. The best-fit algorithm has been shown to attain such property [5], though in a different context.

On the other hand, reducing the amount of fragmentation may not be the only means to maximize MDA utilization. In fact, as already mentioned, overlapping MDAOPs scheduled by nodes outside the two-hop neighborhood might degrade the performance due to physical interference on the receiver side. Therefore, we consider a second algorithm for MDAOP selection, namely the *random-fit* algorithm, which simply selects the location randomly among the set of feasible ones, i.e.

$$\langle d_{rnd}, o_{rnd} \rangle = random\{\langle d_i, o_i \rangle \in \mathcal{A}|x\}. \quad (5)$$

The intuition behind this choice is that, by selecting locations randomly, we expect that, at least at light and medium loads, the chance of having partially overlapping MDAOPs, irrespective of their any-hop neighborhood, is reduced.

An example to compare the set of eligible locations with random-fit and best-fit is illustrated in Fig. 2, when an MDAOP of 3 slots is requested. Location ⟨1, 9⟩ is not eligible with any algorithm because the MDAOP would not fit into it, while locations ⟨4, 11⟩ and ⟨6, 20⟩ are not considered by best-fit because they do not leave the minimum residual.

Both algorithms can be implemented efficiently by employing standard programming techniques which require each node to keep an updated list of the busy and free locations [14]. Since nodes in the backbone tier of WMNs are expected to have relatively high computational capabilities, and the number of slots in a DTIM interval is in the order of thousands, no performance penalty in the WMN should occur due to the execution of any of the selected algorithms for MDAOP scheduling. Clearly, the proposed algorithms are by no means exhaustive. However, we believe that a detailed analysis of these baseline policies will shed some light on the issues of MDAOP scheduling, and can then be used as the starting point for devising solutions that are more sophisticated.

B. *Dynamic Relocation*

In wireless networks, interference between concurrent transmissions is affected by several factors including hardware characteristics (e.g., antenna gain, receiver sensitivity) and environment phenomena (e.g., path-loss, shadowing) which can be hardly predicted during network planning, or automatically detected while the network is operated. Therefore, we argue that relying on the "no interference outside two-hop" assumption makes MDA prone to time-dependent and location-dependent performance degradation because of the interference at the receiver nodes. In order to counterbalance this effect, we propose to dynamically relocate MDAOPs so as to achieve stable performance of the traffic flows that use MDA.

Dynamic relocation consists of three functions which are detailed in the following: interference detection, virtual contention, and relocation. With regard to the first issue, i.e. *interference detection*, we argue that simpler solutions than those proposed in the literature for the general case of WMNs can be adopted. In fact, the existing approaches [13], which are typically based on active or passive monitoring, assume that the Distributed Coordination Function (DCF) of IEEE 802.11 is used for medium access. Therefore, they aim at distinguishing the collisions due to the random access procedure from the losses due to interference. However, in MDA, a collision is prevented from occurring because of the schedule negotiation through MREQ, MREQ, and MADV messages.

In detail, we propose that every node keeps an *interference balance* (or *balance*, for short) for each scheduled MDAOP where it acts as the transmitter. Let $b_{x,i}$ denote the current balance of the MDAOP i requested by node x. When the MDAOP is accepted, $b_{x,i}$ is set to an initial value b_{init}. The balance is

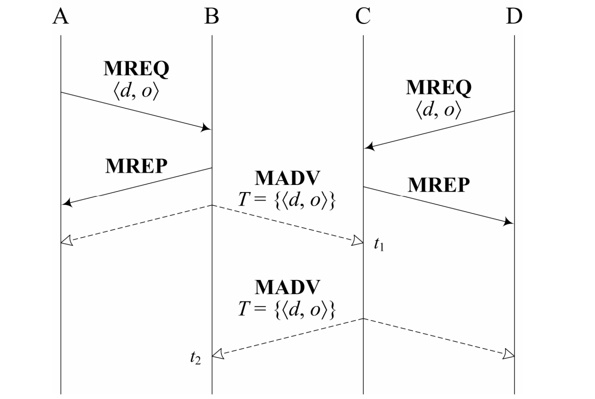


Figure 3. Example of contemporary allocation of an overlapping MDAOP ⟨d, o⟩ from A to B and from D to C, respectively. Such an undesirable situation is not prevented by the standard MDA procedure.

then updated every time a MAC frame is sent during MDAOP i as follows: if the transmitter receives an acknowledgment from the receiver node, $b_{x,i}$ is incremented by a fixed value b_{credit}, up to a maximum value b_{max}; otherwise, if the reception is not acknowledged, $b_{x,i}$ is decremented by a fixed value b_{debit}. When the balance becomes negative, it means that multiple MAC frames have been recently lost. In this case, the node assumes that there is at least one partially overlapping MDAOP somewhere outside its two-hop neighborhood, and the virtual contention function is triggered. Note that the values of b_{init}, b_{credit}, and b_{debit} impact on the performance of traffic flows: detecting interference too early might result in false detection since packets can be lost because of transient changes of the wireless channel due to, e.g., fading. On the other hand, late detection might unnecessarily delay MDAOP relocation that eventually leads to higher packet loss.

When interference is detected by a node, it runs a *virtual contention* procedure to decide whether dynamic relocation will actually occur or not. This step is needed to limit the number of concurrent relocations occurring in the network for the same MDAOP that might otherwise make the procedure unstable. Specifically, any node x keeps a probability $r_{x,i}$ that MDAOP i is relocated when interference is detected. In order to privilege MDAOPs that have been established for longer time intervals, $r_{x,i}$ is initialized to r_{max} when the MDAOP is first set up, and it is decreased by r_{step} at each DTIM interval, until a minimum value r_{min} is reached. In other words, if the MDAOP of a new flow entering the network interferes with another associated to an "older" flow, the latter has a lower chance of being relocated. If the MDAOP wins the virtual contention, then it is not relocated. Otherwise, the following actions are performed by node x: (i) MDAOP i is torn down by means of the MDA reservation messages ,(ii) the past location of the MDAOP is added to a *blacklist* of locations that cannot be scheduled to transmit to the granter of MDAOP i ,(iii) a new location is selected by means of the scheduling algorithm employed for allocation, and the procedure for setting up a renewed MDAOP is initiated. In any case, $b_{x,i}$ is reset to b_{init}. Entries in the blacklist, however, do not last forever. Since it is not known which node (or nodes) caused interference nor for

TABLE 1. TRAFFIC AND MDA CONFIGURATION.

Flow inter-arrival		Weibull, λ = variable, k = 2			
Flow duration		Lognormal, mean 20 s, std. dev. = 2			
CBR traffic		packet size = 1000 bytes, inter-arrival = 32 µs			
b_{init}, b_{max}	30, 50	MDA slot duration			32 µs
b_{credit}	1	DTIM interval			32 ms
b_{debit}	–10	Blacklist entry timeout			3 s
r_{min}	0.10	r_{max}	0.90	r_{step}	0.005

how long, a timer is set to expire after a timeout for each blacklisted location. When this happens, the location can be used again for allocation.

As a final remark, we note that, while the main motivation for dynamically relocating is to combat interference outside the two-hop neighborhood, as a side effect this also solves the problem arising when nodes *within* the two-hop neighborhood allocate partially overlapping MDAOPs at approximately the same time. Consider the simple example illustrated in Fig. 3, where node A requests an MDAOP from node B at approximately the same time as when node D requests an MDAOP from node C. Both granters, i.e. B and C, will accept the request because they are unable to decode the MREQ messages from A and D, respectively. In this example, node C becomes aware of the MDAOP overlap at time t_1, and node B at time t_1. The IEEE 802.11s standard does not specify how to deal with this situation that is instead automatically detected and solved by dynamically relocating the MDAOPs.

IV. PERFORMANCE EVALUATION

In this section, we assess the performance of the MDAOP scheduling and dynamic relocation procedures described above via accurate packet-level simulation. To this aim, we implemented MDA on top of a patched IEEE 802.11 module of the well-known ns-2 simulator [19]. In addition to the MDA access function, the protocol for MDAOP setup, tear-down and advertisement has been fully implemented. Moreover, our modifications to the original IEEE 802.11 code include support for multi-rate transmission and SINR-based physical interference model. Specifically, the SINR of the transmission from node s to node r, which takes place between time instants t_1 and t_2, is [20]:

$$SINR(s,r,t_1,t_2) = \frac{P_{s,r}}{\sum_{k \in \mathcal{T}(t_1,t_2) \setminus s} P_{k,r} + \mathcal{N}} \tag{6}$$

where $P_{i,j}$ is the received power at node j when node i is transmitting, $\mathcal{T}(t_1,t_2)$ is the set of nodes transmitting in the time interval $[t_1, t_2]$, and $\mathcal{N}$ is the background noise power. The received power, in turn, depends on the Euclidean distance $d(s,r)$, in meters, between node s and node r, the transmitted power P_T which is assumed to be the same for all nodes, the path loss exponent γ, and the signal wavelength λ, in meters:

$$P_{s,r} = \frac{P_T(\lambda/4\pi)^2}{d(s,r)^\gamma} \tag{7}$$

TABLE 2. PHYSICAL LAYER PARAMETER VALUES.

	Minimum SINR for each PHY rate							
PHY rate (Mb/s)	6	9	12	18	24	36	48	54
SINR (dB)	9	10	11	13	17	20	25	27
P_T	17 dBm							
Band	5.15 GHz							
$\mathcal{N}$	-95 dBm							

The MAC frame is assumed to be correctly received if the SINR is greater than or equal to the minimum value required by the IEEE 802.11 standard. The parameters used in the simulations, which refer to the IEEE 802.11a OFDM-based physical layer, are reported in Table 2. To quantify the impact of SINR-based interference on MDA with respect to the performance obtained in ideal conditions, we also run simulations where the "no interference outside the two-hop neighborhood" actually holds. We refer to the latter interference model as the *protocol model*.

Nodes are arranged in a regular grid topology of 5×5 nodes with a perturbation which is a "classical" assumption of the performance studies in WMNs (e.g, [4]). Specifically, node spacing is 100 m and the position of each node from the regular grid is perturbed by choosing a random angle uniformly in [0, 2π] and a radius uniformly in [0 m, 25 m]. The perturbation is kept the same for all the scenarios to make results comparable. To ensure that our findings are not biased by the specific placement selected, the whole set of simulations has been rerun with different perturbations, and we ensured that the qualitative behavior obtained with all the metrics considered was always the same. At simulation startup, the physical transmission rate of any link between two nodes *i* and *j* is the highest one such that the SINR at node *j* when node *i* alone is transmitting is above the corresponding threshold reported in Table 2. Physical transmission rates are then kept stable during simulation.

Five nodes of the WMN act as gateways to the Internet. These nodes are arranged in a cross placement, as illustrated in Fig. 4. The remaining twenty nodes, instead, act as APs. Traffic is generated by AP nodes only. Arrival of traffic flows on APs is modeled by means of a Weibull random variable that has been shown to capture the short- and medium-term characteristics of traffic flows in WMNs [16]. The duration of each flow is selected randomly according to Lognormal random variables [16]. All traffic flows are Constant Bit-Rate (CBR), i.e. packets of fixed size are injected at constant inter-arrival times. Note that, because of the different transmission rates, the number of slots per MDAOP vary on a per-link basis, possibly for the same traffic flow along the path to the destination. The TSPEC advertised by the sender node during flow admission is derived straightforwardly from the flow parameters. While this traffic characterization does not model the realistic behavior of traffic flows in a WMN, which depends on the user applications, it nevertheless allows us to derive a worst-case measure of the packet loss. In fact, admitted traffic flows consume the entire amount of the scheduled MDAOPs, which leads to maximum interference to other nodes in the WMN. The traffic

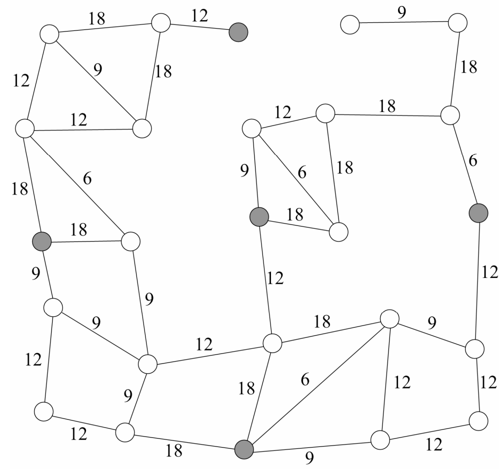


Figure 4. Network topology. Any link is represented as an edge between two nodes, whose weight is the link tramission rate, in Mb/s. Gateways are represented as shaded nodes, APs as empty nodes.

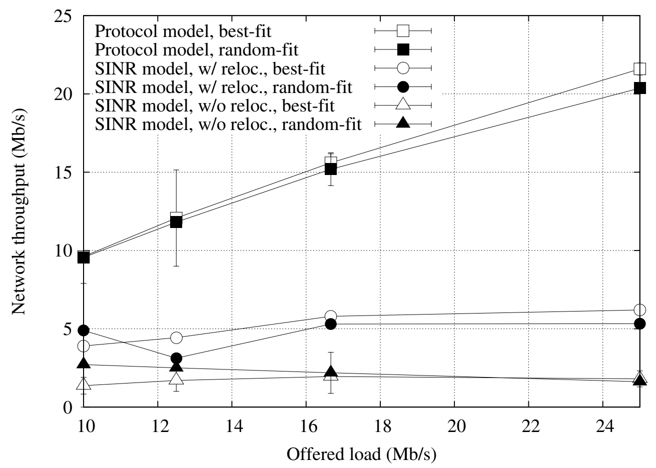


Figure 5. Network throughput vs. offered load. The best-fit and random-fit curves with the protocol model with relocation are almost identical to those without relocation, and are thus not shown.

parameters are those reported in Table 1. With regard to the parameters of interference detection, we have configured them empirically based on preliminary studies whose results are not included. The resulting values are shown in Table 1.

Several independent replications for each simulation scenario have been run, according to the method of independent replications [15], by keeping the positions of nodes fixed, as illustrated in Fig. 4. Mean values are then estimated along with 95% confidence intervals, which are not reported in figures whenever negligible.

In Fig. 5 we show the *network throughput* when the offered load is increased from 10 Mb/s to 25 Mb/s by reducing the average time between the arrival of two consecutive traffic flows at the APs. The network throughput is measured as the total number of bits that are correctly received by all nodes of the WMN in the unit of time. We consider the results obtained under the "protocol model" assumption first. As can be seen, the

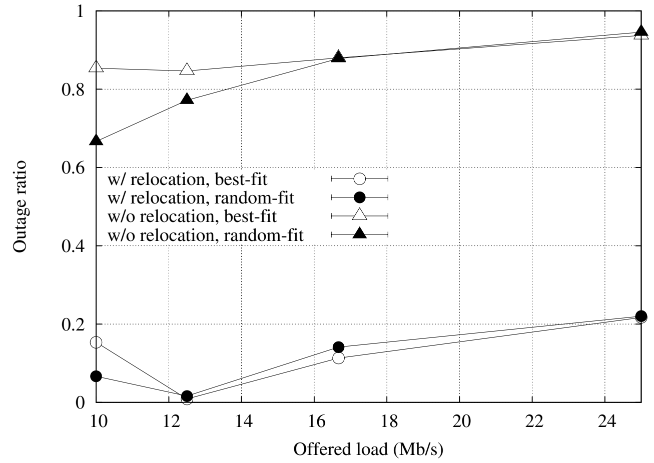


Figure 6. Outage ratio vs. offered load. The protocol model curves are not shown since the outage ratio is 0 under such ideal conditions.

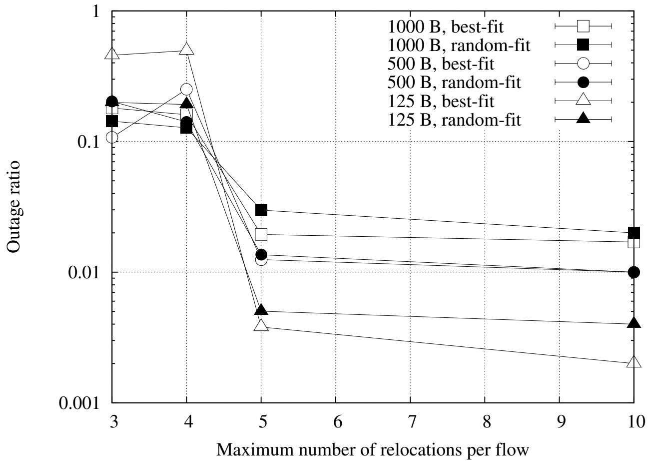


Figure 7. Outage ratio vs. maximum number of relocations per flow. Different values of the MDAOP size, in bytes, are considered, while the offered load is kept constant.

best-fit algorithm achieves slightly higher throughput than random-fit, especially at high loads. This is because positioning MDAOPs within the DTIM interval randomly increases the amount of fragmentation, and hence the probability that an MDAOP cannot be accepted even though the sum of available free slots would fit it in. In any case, however, the throughput obtained under the "protocol model" assumption is much higher than that yielded with the SINR interference model. Such an inflated performance gain confirms that the "protocol model" cannot adequately model the dynamics of a WMN, and is thus not considered in the remainder of the analysis.

The network throughput increases with the offered load when dynamic relocation is also enabled. Specifically, when the network is lightly loaded, i.e. with flow inter-arrival time equal to 20 s, the network throughput with random-fit is greater than that with best-fit. In fact, when allocation of the MDAOPs within the DTIM interval is sparse, flows are typically terminated because they exceed the maximum number of relocations which is set to 5 in these scenarios. Positioning randomly the MDAOPs helps reduce the chance that MDAOPs are allocated in the same sets of slots by nodes outside the respective two-hop neighborhoods. However, when the offered load increases, employing the best-fit algorithm pays off in terms of network throughput because it greatly reduces the DTIM interval fragmentation.

Finally, without dynamic relocation, regardless of the scheduling algorithm employed, the network throughput is substantially smaller than that with dynamic relocation. This is because interference of transmissions outside the two-hop neighborhood is not dealt with, thus a large fraction of packets cannot be decoded by the receiver nodes. Such failed transmissions waste channel capacity. In these conditions it is of the utmost importance to reduce the chance that MDAOPs are placed in non-overlapping sets of slots, which explains why the random-fit algorithm performs better than best-fit one. However, the difference between the curves becomes smaller as the offered load increases, until it eventually disappears at 25 Mb/s.

In Fig. 6 we show the *outage ratio* which is defined as the number of flows whose packet loss is greater than a threshold, equal to 5% in simulations, divided by the total number of traffic flows. It is unfeasible to use MDA as the medium access function without dynamic relocation because almost all traffic flows experience performance degradation due to high packet loss. When instead the dynamic relocation is enabled, the outage ratio is kept at a much lower value, mostly due to the transient phases during which the MDAOP is torn down and set up again after interference detection. Therefore, the outage ratio is expected to become even smaller if the average duration of the traffic flows increases.

In our second set of scenarios, we study the sensitivity of the dynamic relocation function to the following factors: maximum number of relocations per flow, which is increased from 3 to 10, and MDAOP size, in bytes (B), set to 250 B, 500 B, and 1000 B. The overall offered load in the WMN is kept constant, i.e. ~16.7 Mb/s, by setting the average flow inter-arrival time appropriately for each MDAOP size.

The outage ratio is shown in Fig. 7. As can be seen, the outage ratio is relatively high when the number of maximum relocations per flow is small. This is because limiting too much the number of relocations leads to a large fraction of MDAOPs being unnecessarily torn down soon after they enter the network, i.e. when it is more likely that dynamic relocation will be actually triggered after interference detection. All these flows are likely to experience packet loss greater than the 5% threshold while they are active. On the other hand, when MDAOPs are allowed to be relocated several times, they eventually find a stable location where to accumulate credits into their balance. This will reduce the chance of being relocated again if new interfering MDAOPs are allocated outside the respective two-hop neighborhoods. In the set of considered scenarios, five relocations are enough to make dynamic relocation stable in most

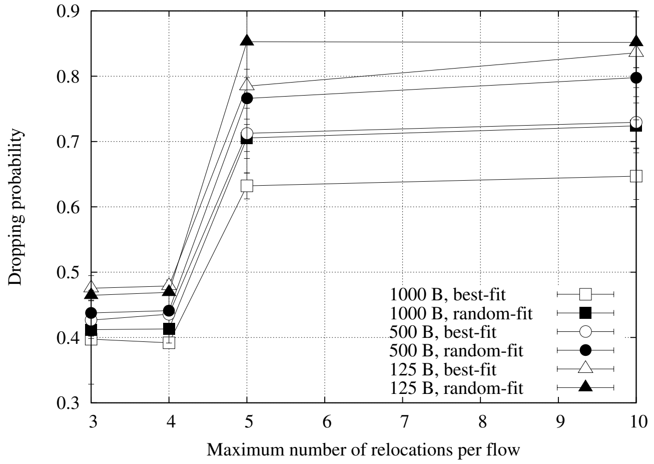


Figure 8. Dropping probability vs. maximum number of relocations per flow. Different values of the MDAOP size, in bytes, are considered, while the offered load is kept constant.

cases, which is why there is a steep decrease of the curves when the maximum number of relocations per flow is incremented from 4 to 5. In general, this "step" can occur at different values of the maximum number of relocations per flow, depending on the network configuration and traffic load conditions.

Since the network is moderately loaded, the best-fit algorithm achieves better performance than random-fit in all cases of interest, i.e. except when the maximum number of relocations per flow is smaller than or equal to 4. Finally, the outage ratio is greater with traffic flows requesting larger MDAOPs. In fact, the probability that interfering MDAOPs are allocated in overlapping slots increases with the MDAOP size. Therefore, large MDAOPs are relocated more often than small MDAOPs, which increase the packet loss and the chance that they are torn down due to the maximum number of relocations being exceeded.

Finally, in MDA there are two reasons why an MDAOP is definitely canceled, i.e. a new setup phase is not triggered for the MDAOP: either the maximum number of relocations per flow is exceeded or the scheduling algorithm cannot find a free location large enough to contain the MDAOP by considering the blacklisted slots as unavailable. We define the *dropping probability* as the ratio between the number of MDAOPs torn down because of the latter reason only to the total number of canceled MDAOPs. The dropping probability is shown in Fig. 8. As can be seen, when the maximum number of relocations per flow is smaller than or equal to 4, MDAOP tear down is mostly due to the maximum number of relocations per flow being exceeded, which confirms the hypothesis suggested to explain the behavior observed in Fig. 7. On the other hand, when the maximum number of relocations per flow increases, MDAOPs are mostly canceled because the DTIM interval is considered full. This phenomenon is especially evident with small MDAOPs, because there are more traffic flows concurrently active in the network, each contributing to the saturation of the DTIM interval.

V. RELATED WORK

To the best of our knowledge, this is the first work that defines the algorithms for MDA left unspecified by the recent draft standard IEEE 802.11s. The only work where MDA is evaluated is [9], which includes preliminary simulation results obtained with nodes placed in a chain topology only. Modifications of the IEEE 802.11s in order to provide end-to-end QoS are instead proposed by the authors of [23], which however require a major rewriting of the MAC layer.

On the other hand, there are several research works in the context of IEEE 802.11 WMNs. For instance, [8] was one of the first studies to systematically evaluate via simulation the inability of the IEEE 802.11 to provide end-to-end traffic flows with guaranteed performance due to intra- and inter-flow contention and spatial bias. Solutions proposed in this context, however, cannot be re-used with MDA since they assume CSMA/CA medium access function.

As far as other technologies are concerned, the IEEE 802.16 has been investigated in [17] as a candidate to achieve delay guarantees in the backbone of a WMN, while in [6], an end-to-end bandwidth reservation protocol for the mesh mode of IEEE 802.16 has been proposed for the same purpose. Finally, several researchers have developed solutions to provide QoS with Time Division Multiple Access (TDMA) based on MAC protocols (e.g., [7] [22]). However, the proposed algorithms cannot be applied straightforwardly to IEEE 802.11s MDA because of: (i) the difference between the signaling schemes, which is often assumed to be ideal; and (ii) the typical assumption made about "no interference outside two-hop neighborhood" which can lead to over-optimistic conclusions with MDA.

VI. CONCLUSIONS

In this paper, we have analyzed the MDA function of the draft standard IEEE 802.11s which is designed to enable controlling QoS in the backbone of a WMN. We have then proposed two algorithms, namely best-fit and random-fit, to schedule MDAOPs within the DTIM interval. Furthermore, we have observed that MDA is implicitly based on the assumption that interference due to nodes lying outside the two-hop neighborhood of a link is negligible. However, this assumption hardly holds in realistic conditions. We have therefore proposed an algorithm, called dynamic relocation, to tackle this problem.

We have assessed the performance of the proposed algorithms via packet-level simulation. The results have shown that, with a SINR-based interference model, dynamic relocation using MDA plays a crucial role in keeping the packet loss of traffic flows reasonably low. In the considered scenarios, if the MDAOPs were allowed to be relocated several times before being canceled, most of them eventually found a stable location where to stay until regular traffic flow termination. We have compared the performance obtained using the best-fit and random-fit algorithm in terms of the network throughput. Best-fit performed better than random-fit in all cases, expect when the network was lightly loaded, because of its ability to keep low the amount of DTIM interval fragmentation. In fact, the latter causes MDAOPs to be unnecessarily dropped only because the slots available for the allocation are not contiguous.

Our work can be extended in several directions, including: design of more sophisticated algorithms to schedule MDAOPs within the DTIM, and adaptive tuning of the dynamic relocation parameters based on traffic load and physical layer measurements. Additionally, applications based on MDA can be devised, such as centralized/distributed admission control schemes for end-to-end traffic flows.

ACKNOWLEDGMENT

We would like to thank Michele Soleri for his help in developing the simulation module for MDA in ns-2.

REFERENCES

[1] Akyildiz, I. F. and Wang, X. A survey on Wireless Mesh Networks. IEEE Commun. Mag., vol. 43, no. 9, Sep. 2005, pp. 23–30.

[2] Bianchi, G., Tinnirello, I, Scalia, L. Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations. IEEE Network, vol. 19, no. 4, 2005, pp. 28–34.

[3] Bruno, R., Conti, M. and Gregori, E. Mesh networks: Commodity multihop ad hoc networks. IEEE Commun. Mag., vol. 43, no. 3, Mar. 2005, pp. 23–131.

[4] Camp, J., Robinson, J. Steger, C., and Knightly, E. W. Measurement driven deployment of a two-tier urban mesh access network. ACM MobySys'06, Uppsala, Sweden, Jun. 19–22, 2007, pp. 96–109.

[5] Carter, B. A comparison of next-fit, first-fit, and best-fit. ACM Commun., vol. 20, no. 3, 1977, pp. 191–192.

[6] Cicconetti, C., Erta, A., Gardellin, V., Lenzini, L., and Mingozzi, E. End-to-end bandwidth reservation in IEEE 802.16 mesh networks. Proc. IEEE MeshTech'07, Pisa, Italy, Oct. 10, 2007.

[7] Djukic, P. and Valaee, S. Link scheduling for minimum delay in spatial re-use TDMA. Proc. IEEE INFOCOM'07, Anchorage, AK, USA, May 6–12, 2007, pp. 28–36.

[8] Gambiroza, V., Sadeghi, B., and Knightly, E. W. End-to-end performance and fairness in multihop wireless backhaul networks. Proc. ACM MobiCom'04, Philadelphia, PA, USA, Sept. 26–Oct. 1, 2004, pp. 287–301.

[9] Hiertz, G. R., Max, S., Zang, Y., Junge, T., and Denteneer, D. IEEE 802.11s MAC fundamentals. Proc. IEEE MeshTech'07, Pisa, Italy, Oct. 10, 2007.

[10] Hiertz, G., Max, S., Zhao, R., Denteneer, D., and Berlemann, L. Principles of IEEE 802.11s. Proc. IEEE ICCCN'07, Hololulu, Hawaii, USA, Aug. 13–16, 2007, pp. 1002–1007.

[11] IEEE P802.11s/D1.08. Draft amendment to standard – IEEE 802.11: ESS Mesh Networking. IEEE, Jan. 2008.

[12] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE, June 2007.

[13] Kim, K-H. and Shin, K. G. On accurate measurement of link quality in multi-hop Wireless Mesh Networks. Proc. ACM MobiCom'06, Los Angeles, CA, USA, Sep. 23–26, 2006, pp. 38–49.

[14] Knuth, D. E. The art of computer programming, volume 1. Third edition, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1997, ISBN 0-201-89683-4.

[15] Law, A. M. and Kelton, W. D. Simulation modeling and analysis. Third edition, McGraw-Hill, 2000, ISSN 0070592926.

[16] Meng, X., Wong, S. H. Y., Yuan, Y., and Lu, S. Characterizing flows in large wireless data networks. ACM MobiCom'04, Philadelphia, Pennsylvania, USA, Sep. 26–Oct. 1, 2004, pp. 174–186.

[17] Narlikar, G., Wilfong, G., and Zhang, L. Designing multihop wireless backhaul networks with delay guarantees. Proc. IEEE INFOCOM'06, Barcellona, Spain, Apr. 23–29, pp. 1–12.

[18] Nichols, M. A., Siegel, H. J., Dietz, H. G., Quong, R. W., and Nation, W. G. Eliminating memory for fragmentation within partitionable SIMD/SPMD machines. IEEE Trans. Parallel Distrib. Syst., vol. 3, no. 2, Jul. 1991, pp. 290–303.

[19] http://www.isi.edu/nsnam/ns/, network simulator 2, version 2.31, released on Mar. 10, 2007.

[20] Stuber, G. L.. Principles of Mobile Communication. Kluwer Academic Publishers, 2001.

[21] Takai, M., Martin, J., and Bagrodia, R.. Effects of wireless physical layer modeling in mobile ad hoc networks. Proc. MobiHoc 2001, Long Beach, CA, USA, Oct. 4–5, 2001, pp. 87–94.

[22] Vidhyashankar, V., Manoj, B. S., and Murthy, C. S. R. Slot allocation schemes for delay sensitive traffic support in asynchronous Wireless Mesh Networks. Computer networks (Elsevier), vol. 50, no. 15, Oct. 2006, pp. 2595–2613.

[23] Zhao, R., Walke, B., and Hiertz, G. R. An efficient IEEE 802.11 ESS mesh network supporting Quality-of-Service. IEEE J. Selected Areas Commun., vol. 24, no. 11, Nov. 2006, pp. 2005–2017.