FEBA: A Bandwidth Allocation Algorithm for Service Differentiation in IEEE 802.16 Mesh Networks

Claudio Cicconetti, *Member, IEEE*, Ian F. Akyildiz, *Fellow, IEEE*, and Luciano Lenzini, *Member, IEEE*

***Abstract*—In wireless mesh networks, the end-to-end throughput of traffic flows depends on the path length, i.e., the higher the number of hops, the lower becomes the throughput. In this paper, a fair end-to-end bandwidth allocation (FEBA) algorithm is introduced to solve this problem. FEBA is implemented at the medium access control (MAC) layer of single-radio, multiple channels IEEE 802.16 mesh nodes, operated in a distributed coordinated scheduling mode. FEBA negotiates bandwidth among neighbors to assign a fair share proportional to a specified weight to each end-to-end traffic flow. This way traffic flows are served in a differentiated manner, with higher priority traffic flows being allocated more bandwidth on the average than the lower priority traffic flows. In fact, a node requests/grants bandwidth from/to its neighbors in a round-robin fashion where the amount of service depends on both the load on its different links and the priority of currently active traffic flows. If multiple channels are available, they are all shared evenly in order to increase the network capacity due to frequency reuse. The performance of FEBA is evaluated by extensive simulations. It is shown that wireless resources are shared fairly among best-effort traffic flows, while multimedia streams are provided with a differentiated service that enables quality of service.**

***Index Terms*—Access protocols, packet reservation multiaccess, scheduling, wireless LAN.**

I. Introduction

WIRELESS mesh networks (WMNs) are emerging as a key technology for next generation wireless networking. Due to their several advantages compared to other wireless networks, WMNs are undergoing a very fast development progress and inspiring numerous applications. The WMN architecture, in general, consists of two tiers [1]: *backhaul* and *access* tiers where the *backhaul* tier consists of *wireless mesh routers* which create a multi-hop ad hoc network and provide Internet or intra-WMN connections to *wireless mesh clients* in the *access* tier. Wireless mesh routers are fixed devices with unlimited energy, high computational and communication capabilities.

Recently, some research has been conducted to use the well-known IEEE 802.11 technology for the backhaul tier which has performance problems in its current form [2]. Indeed, in the existing IEEE 802.11 technology there are few available channels [3], the transmission range is very limited [4] unless expensive external amplified antennae are employed, the medium access control (MAC) protocol achieves low performance for multi-hop traffic flows [5]. In particular, the fairness among traffic flows traversing a different number of hops is severely affected. Specifically, the available network capacity, accordingly the system throughput, decreases with the increasing number of hops because 1) some nodes experience backoff more than some others due to the "hidden node" problem, and 2) flows with a longer path length have more contentions for medium access than the flows originated closer to their destination (called "spatial bias" problem). More contentions result in higher probability for collisions and losses. Several solutions are suggested in the literature to solve this problem. These solutions use a MAC protocol based on time division multiple access (TDMA) [6], [7] with highly simplifying assumptions which make them impractical for actual network deployments. The working group IEEE 802.11s is established to investigate these research problems [8]. However, the research is still in the infancy phase [9], [10].

An alternative to IEEE 802.11 is the IEEE 802.16 standard [11] which is specifically designed for the backhaul tier of WMNs and includes a TDMA MAC protocol operating in *mesh mode* where nodes coordinate among themselves to transmit packets in a multi-hop manner. There are two coordination modes: *centralized* and *distributed*. In the *centralized* mode, the Base Station (BS) is responsible for defining the schedule of transmissions in the entire network. In the *distributed* mode, transmissions are scheduled in a fully distributed fashion without requiring any interaction with the BS. The distributed mode is more flexible and responsive than the centralized mode, since decisions are taken locally by nodes according to their current traffic load and physical channel status. In this study we consider the distributed mode alone.

In the distributed mode, the IEEE 802.16 standard specifies a MAC protocol to coordinate the transmission of control messages in a collision-free manner [12]. On the other hand, the bandwidth allocation problem in the distributed mode is left unsolved by the IEEE 802.16 standard so far except providing some control messages that may be used for this purpose, such as bandwidth requests and grants.

In this paper we propose a fair end-to-end bandwidth allocation (FEBA) algorithm for IEEE 802.16 nodes to negotiate bandwidth in a multi-channel environment.[1] Our contributions can be summarized as follows.

Manuscript received August 02, 2007; revised March 19, 2008; approved by IEEE/ACM Transactions on Networking Editor S. Das. First published October 03, 2008; current version published June 17, 2009.

C. Cicconetti and L. Lenzini are with the Dipartimento di Ingegneria dell'Informazione, University of Pisa, 56010 Pisa, Italy (e-mail: c.cicconetti@iet.unipi.it; l.lenzini@iet.unipi.it).

I. F. Akyildiz is with the Broadband Wireless Networking (BWN) Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ian@ece.gatech.edu).

Digital Object Identifier 10.1109/TNET.2008.2005221

[1]A preliminary version of this paper appeared in [13]

1) FEBA tackles the "spatial bias" problem by keeping separate queues at each node for each traversing traffic flow. Furthermore, differentiated service is provided by serving traffic flows proportionally to their priority, specified in the standard IEEE 802.16 MAC header.
2) The "hidden node" problem, which can lead to information asymmetry between flows that are one hop away from each other, can be substantially mitigated by FEBA through the so-called *regranting* procedure.
3) Unlike most solutions for TDMA MAC protocols, FEBA is able to react promptly to short-term variations of the traffic load in the network. FEBA is implemented in a fully distributed manner, thus, it does not incur the overhead of signaling towards/from a centralized node, which often makes existing solutions impractical.

The terminology adopted throughout this paper is introduced in Section II, which also covers an overview of the IEEE 802.16 mesh MAC protocol with distributed coordinated scheduling. In Section III, we state the motivation and objectives of this work, and provide a detailed description of the FEBA. Exhaustive simulation results are then reported in Section IV, and Section V addresses the related work. We conclude the paper in Section VI.

II. IEEE 802.16 Mesh

In this section, we discuss the aspects of the IEEE 802.16 MAC protocol relevant to the mesh mode with distributed scheduling. As already introduced, data transmission is coordinated among nodes in a fully distributed manner. Hereafter, we adopt the IEEE 802.16 terminology which defines two nodes that can communicate between each other as *neighbors*. In IEEE 802.16, a logical *link* is set up between any two neighbors by means of a link establishment procedure.

The time is partitioned into frames of fixed duration. Each frame consists of a control sub-frame and a data sub-frame, as illustrated in Fig. 1. Control sub-frames are partitioned into slots of fixed duration (hereafter, *control slots*), which are accessed by nodes based on the distributed election procedure specified by the standard. This ensures that, in a steady state, each node gets the opportunity to transmit control messages in a regular, though not periodical, manner. A control slot consists of seven OFDM symbols, two of which are used as a physical preamble to synchronize the receiver, and one is used as a guard symbol. Up to 16 control slots can be specified per frame. Data sub-frames consist of a fixed number of data mini-slots (hereafter, *slots*), up to 256. The number of bytes conveyed by a slot depends on the modulation and coding scheme (MCS) used by the sender to transmit data to the receiver. Every node dynamically adapts the MCS from neighbor to neighbor based on measurements of the received signal quality at the physical layer. However, control messages are transmitted using the most robust modulation and coding scheme, i.e., QPSK with code rate 1/2. An IEEE 802.16 mesh network can employ up to 16 non-interfering channels for data transmission to increase the available transmission capacity for nearby nodes which cannot exploit spatial reuse. However, control messages are transmitted by all nodes in the network in the same channel, e.g., channel Ch1 in Fig. 1.

Data transmission is coordinated by means of a three-way handshake procedure: 1) a node, namely the *requester*, asks

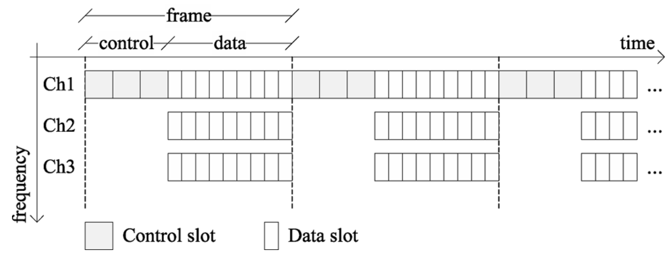

Fig. 1. Example of the frame structure with three channels.

a neighbor node, namely the *granter*, to allocate some bandwidth; 2) the granter advertises a set of slots as 'granted' to the requester; 3) the requester confirms that it will actually use that set of slots (or part thereof) to transmit data. This is carried out by means of mesh distributed schedule (MSH-DSCH) messages, which contain a list of information elements (IEs), classified by the IEEE 802.16 standard into the following four types. A *request IE* indicates that the requester has data addressed to the granter awaiting transmission, i.e., backlog. The granter reserves bandwidth for the requester using *grant IEs*, each containing a range of slots over a range of frames in a given channel. A grant is thus expressed as a triple $\langle \mathrm{slot\ range,\ frame\ range,\ channel} \rangle$, e.g., $\langle [3,8],[4,5],1 \rangle$ represents the slots numbered from 3 to 8 in the data sub-frame of the fourth and fifth frames since the grant is issued, in channel 1. The same set of parameters is also used in *confirmation IEs*, which are used by the requester to complete the three-way handshake procedure. Finally, *availability IEs* can be used to report slots that cannot be used by the requester to transmit or receive data.

Bandwidth negotiation in IEEE 802.16 mesh is implicitly based on the assumption that only the one-hop neighbors of a receiver can interfere with its ongoing data reception, sometimes referred to as "protocol-model" [14]. In other words, it is assumed that the cumulative interference of nodes that are two or more hops away from the receiver is negligible. This assumption is inherited by the bandwidth allocation algorithm that we develop and is discussed in Section III-G. Nodes need to keep track of all combinations $\langle \mathrm{slot,\ frame,\ channel} \rangle$ that cannot be granted to the requester if any of the following conditions is true: (i) the granter transmits/receives in $\langle \mathrm{slot,\ frame} \rangle$; (ii) the requester transmits/receives in $\langle \mathrm{slot,\ frame} \rangle$; (iii) one of the requester's neighbors transmits in $\langle \mathrm{slot, frame, channel} \rangle$. Conditions (i) and (ii) are needed because the nodes have a single radio, thus, they can either receive from or transmit on a single channel at a given time, while the condition (iii) results from the "protocol-model" assumption.

Once a grant has been confirmed by the requester, it is expected to transmit data, i.e., MAC Protocol Data Units (PDUs), in the slots that are allocated by the granter. Each PDU includes the IEEE 802.16 MAC header with the Node Identifier (Node ID) of the requester and the granter, and the PDU length, priority (3 bits) and drop precedence (2 bits). Additionally, a 32-bit Cyclic Redundancy Code (CRC) is added to ensure data reliability. If needed, the requester can fragment a MAC Service Data Unit (SDU) received from upper layers into multiple PDUs to limit the capacity wastage. Fragmenting SDUs incurs a small overhead penalty, i.e., 13 bytes/fragment, because the MAC header, including a fragmentation sub-header, must be added to each fragment.

III. Fair End-to-End Bandwidth Allocation

In this section we describe the components of FEBA which are aimed at fulfilling the objectives stated below in Section III-A. As already introduced, whenever a node gains access to the control sub-frame based on the standard distributed election procedure, it broadcasts an MSH-DSCH message to its neighbors. The content of the MSH-DSCH message is defined by FEBA. Specifically, in Section III-B, we describe the algorithm to share the node's bandwidth, i.e., to decide how many slots have to be requested from/granted to any neighbor in the outgoing MSH-DSCH message. As far as granting is concerned, slots are only assigned over a frame interval, called schedule horizon, whose duration is dynamically adapted on a link by link basis according to the procedure specified in Section III-C. The algorithm to select the arrangement of the granted slots within the schedule horizon is then described in Section III-D, which also considers the issues of allocating bandwidth over multiple channels. Moreover, in Section III-E we investigate the "hidden node" problem, while the algorithm employed by any node to schedule SDUs at each of its links is reported in Section III-F. Finally, we conclude in Section III-G with a discussion on the implementation issues of FEBA.

A. Objectives and Assumptions

We assume that the network topology is fixed. Moreover, although the quality perceived by nodes on different links may change over time, we assume that the links are stable enough not to be torn down. This assumption is motivated by the fact that IEEE 802.16 nodes are intended to be utilized as wireless mesh routers in the backhaul tier of a highly reliable WMN, which ensures stable routes. Finally, we assume that each node has a single radio interface which can dynamically switch among several available channels to one channel at a time. However, this is not a requirement because FEBA does not rely on the use of multiple channels. However, using multiple channels we can boost the overall throughput performance as we will demonstrate in the performance evaluation section (Section IV-A, Section IV-C).

The "fairness" is a desirable property for any MAC protocol employed as the backhaul tier of a WMN. Although this notion is well-defined in single-hop networks, it is not clear in the case of multi-hop networks. In the literature, the definition of a traffic flow typically depends on the target application of the proposed solution. For instance, a fairness reference model for WMN nodes is developed in [5] where they act as Transit Access Points (TAPs) for traffic to and from the Internet. Since each TAP is assumed to correspond to a single residence/public hot spot, all micro-flows originated at a TAP are treated as a single aggregate. Instead, in this work we do not consider any specific assumption on where the ingress or egress points of the WMN backhaul are located. Therefore, we argue that any flow of packets identified by a $\langle \mathrm{source}, \mathrm{destination} \rangle$ ought to obtain the same treatment from the network. Furthermore, since service differentiation is another beneficial property of a WMN, we introduce the following definition:

Definition 1 (Traffic Flow): A traffic flow i is a stream of IP datagrams from a source s_i to a destination node d_i, with a given priority (or weight) w_i. The throughput r achieved by any two traffic flows i, j that are continuously backlogged in a time interval should then be

$$\frac{r_i}{r_j} = \frac{w_i}{w_j}.$$

Therefore, the priority w_i of a traffic flow i is a quantitative measure of how much it should be allowed to take over another flow j with an overlapping (or partially overlapping) path and $w_j < w_i$.

Moreover, as it is known in wireless networks the transmission rate of links can vary over time. A typical problem is thus whether the fairness should be measured in terms of the amount of bandwidth, i.e., bytes, or time that is consumed by a traffic flow [15]. In the former case, nodes that perceive worse channel conditions than the others will consume more resources to transmit than their fair share of bandwidth. We consider this case to be the most relevant to our work, since nodes are assumed to be fixed, hence transmission rates are bound to remain the same for long periods.

As a summary, we aim at providing traffic flows, as defined above, with weighted max-min fair access to the network resources, in terms of throughput, regardless of their spatial bias. In general, a bandwidth allocation to different entities is said to be max-min fair if it is not possible to increase any bandwidth share without decreasing another bandwidth share which is already smaller than that [16]. The definition of *weighted* max-min fairness is straightforward.

The basic idea of FEBA is that each node assigns bandwidth requests and grants in a round-robin manner where the amount of allocated bandwidth in bytes, is proportional to the number of traffic flows weighted on their priorities. The inspiration of this approach comes from "classical" studies on bandwidth allocation in wireline networks, e.g., [17], where the max-min fairness is achieved through distributed round-robin scheduling at each network node. In fact, it is easy to prove that a bandwidth allocation is max-min fair if and only if each unsatisfied flow has at least a bottleneck node. A node is *bottleneck* for a given flow if (i) the node's bandwidth allocated to that flow is at least as large as the bandwidth share of any other flow traversing the same node, and (ii) the node's bandwidth is entirely allocated. This model cannot be applied directly to the case of WMNs , because the bandwidth of a node (or of a link) also depends on the amount of traffic that is carried by neighboring nodes (or links). Generalizations of the above model have thus been presented in the literature to include WMN constraints. For example in [18], the bandwidth allocation is realized on a per-clique rather than per-node or per-link basis. Alternatively, in [19] dynamic weights have been introduced depending on the congestion level in the neighborhood. More recently, a hop-by-hop congestion control scheme is developed in [20] that achieves proportional fairness among traffic flows which is shown to be stable under simplified assumptions. However, existing models still represent a very simplified version of reality, since they do not (fully) take into account effects due to, for example, packetized transmissions and MAC signaling, such as the three-way handshake for bandwidth reservation in the case of IEEE 802.16.

In this work we face a more complex system model which cannot be tackled by analytical modeling techniques. Thus, we use simulation modeling to evaluate the performance and validate the effectiveness of the suggested solutions as shown in Section IV.

In the remainder of this section, we define a traffic flow from node i to node j to be *active* if there are SDUs originated at node i directed to node j. Since in IEEE 802.16 there is no end-to-end signaling for data transmission, each node x along the path from i to j must keep track of the set of active flows. In particular, as soon as an SDU is received by x, it adds $\langle i, j, w \rangle$ to the set of active flows, if not present, based on the source and destination IP addresses, and the priority of that SDU. On the other hand, node x removes $\langle i, j, w \rangle$ from the set of active flows when it does not receive SDUs belonging to that flow within a timeout period which we assume to be equal to the default TCP Maximum Segment Lifetime (MSL), i.e., two minutes.

B. Request/Grant Procedure

As mentioned before, the so-called spatial bias in WMNs is mainly caused by multiple flows being aggregated at each hop. Therefore, the flows that traverse more hops need to contend for medium access more often than the flows with a shorter path length.

In [13] we explained the FEBA procedure for requesting/granting bandwidth in detail and gave a complete description of its data structures. In the following we explain the extended version of FEBA which supports differentiated services for traffic flows. Let a node X maintain two virtual queues towards any of its neighbors, say Y: the *requesting* queue and the *granting* queue. The occupancy of the former, i.e., the requesting queue, is the total amount of backlogged bytes directed to Y. On the other hand, the total amount of data enqueued at node Y directed to node X is the occupancy of the granting queue. Note that in IEEE 802.16 bandwidth requests and grants are expressed in units of slots. Since we keep track of the buffer occupancies in bytes, nodes must convert between bytes and slots, depending on the current MCS employed. A requesting queue is said to be active if there are SDUs waiting to be transmitted by the sender node for which no requests have been issued. A granting queue is said to be active if there are *pending* requests, i.e., the receiver node has not granted the entire amount of bytes requested by the sender.

In FEBA each active queue, both requesting and granting, is assigned a *weight* (ϕ) which is used by the bandwidth request/grant procedure below. The weight ϕ_i of any queue i is computed so that the amount of service is proportional to number of traffic flows under service, weighted based on their priorities:

$$\phi_i = \frac{\sum\limits_{j \in \mathcal{A}} w_j \cdot I_i(j)}{\sum\limits_{j \in \mathcal{A}} w_j} \tag{1}$$

where $\mathcal{A}$ is the set of all active traffic flows served by this node, j is an active flow with priority w_j, and $I_i(j)$ is an indicator function which equals 1 if j is under service at queue i, 0 otherwise. Since each traffic flow is under service at exactly one queue, $\sum_i \phi_i = 1$.

Requesting and granting active queues are then served in a round-robin fashion: at each round, queue i is entitled to serve $\phi_i F_{\mathrm{RR}}$ bytes, where F_{RR} is a system parameter, called *target round duration*. Specifically, when defining the content of the MSH-DSCH message, each granting queue i is entitled to grant up to $\phi_i F_{\mathrm{RR}}$ bytes to neighbor i, while each requesting queue i is entitled to request up to $\phi_i F_{\mathrm{RR}}$ bytes from neighbor i. If the number of bytes requested from (granted to) neighbor i is smaller than $\phi_i F_{\mathrm{RR}}$, the queue is removed from the active list after service. F_{RR} is set to the smallest number of bytes such that $\phi_i F_{\mathrm{RR}}$ is greater than or equal to the MAC Maximum Transfer Unit (MTU), for any queue i. This way the chance that SDUs are fragmented is reduced, hence saving some MAC overhead. In any case, FEBA does not require all nodes to have the same F_{RR}, which is a local parameter of nodes.

There are cases when a queue i, though in the active list, is not eligible for service. More specifically, a granting queue i is not eligible for service when all the slots in the grant horizon are busy, i.e., it is not possible to grant any slots to neighbor i. On the other hand, a requesting queue i is not eligible for service when $req_i^{out} - cnf_i^{out} > pending_{\max}$, i.e., the requester demands cannot be satisfied. Ineligible queues are not removed from the active list. However, we store in the variable lag_i the number of bytes that queue i could not consume while it was ineligible. In a subsequent turn when queue i eventually becomes eligible, it will receive an extra service equal to lag_i bytes. To prevent queue i from not being served at all, lag_i is bounded by a threshold, $lag_{\max}$ which is set to 2 F_{RR}. Note that the notion of "lagging" queue is known in the scheduling literature [21] where it is used for fair-queueing in wireless networks. However, those results cannot be directly applied to the context of IEEE 802.16 mesh networks.

C. Schedule Horizon

The procedure described in the previous section is used by each node to share bandwidth among its neighbors by scheduling requests and grants. In this section we describe how to select the time interval when grants can take place, which we call the *schedule horizon*.[2] More formally, the schedule horizon is defined as follows.

Definition 2 (Schedule Horizon): The schedule horizon between a granter node G and a requester node R is defined as the range of frames where G is allowed to grant slots for data to be transmitted by R.

Since FEBA works by allocating requests and grants dynamically based on the current status of the traffic load and physical transmission rates, the schedule horizon should be as small as possible in order to closely follow these variations. However, its duration should be enough to allow the granter to reserve the channel capacity entirely to the requester. Otherwise,

[2]We used the term "horizon" instead of the more common "period" so as to stress that the schedule is neither periodic nor it provides long-term reservations, as is often the case with TDMA MAC protocols for WMNs.

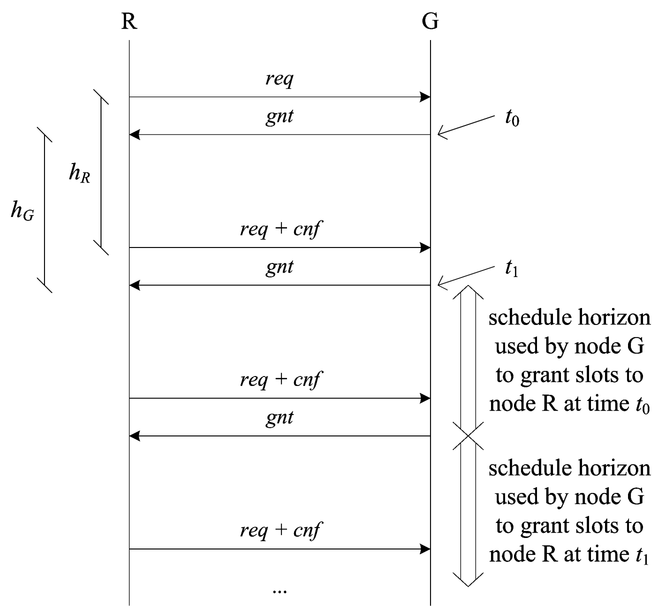

Fig. 2. Schedule horizon of node G at times t_0 and t_1.

the full channel utilization might not be achieved. The schedule horizon duration is thus equal to the interval h_G between two consecutive turns for node G to transmit MSH-DSCH messages, i.e., to grant slots to any neighbor. Furthermore, the schedule horizon should not begin before the requester has the chance to confirm the slots that it is granted. Otherwise, these unconfirmed slots cannot be used for data transmission according to the IEEE 802.16 MAC protocol. Therefore, the granter should refrain from granting slots before a time interval equal to h_R, i.e., the interval between two consecutive turns for node R to transmit MSH-DSCH messages. By summarizing, the schedule horizon at time t computed by the granter is $[t+h_R, t+h_R+h_G]$. As a result of the MAC protocol being based on TDMA, the schedule horizon is expressed in terms of frames.

An example of the computation at node G of the schedule horizon to grant slots to node R at times t_0 and t_1, respectively, is illustrated in Fig. 2. In the example it is assumed that node R confirms the granted slots in the same MSH-DSCH used for requesting further bandwidth. It is worth noting that in a real IEEE 802.16 mesh network, the access to the control sub-frame is determined by the distributed election procedure, which does not guarantee perfectly periodic access. Instead, even in ideal channel conditions, the interval between two consecutive transmissions of MSH-DSCH messages varies over time. Furthermore, on the average, nodes with less two-hop neighbors access the control sub-frame more often than the nodes with a larger two-hop neighborhood [12].

The derivation of the schedule horizon thus requires the knowledge of the interval h between two consecutive transmissions of MSH-DSCH messages in the neighborhood which, however, is not available in practice. In fact, the perfect prediction of the timing of MSH-DSCH messages would require the exchange of network-wide information that is not specified by the IEEE 802.16 standard. Therefore, for the purpose of setting the schedule horizon, h needs to be estimated. This can be done in a simple, though effective, way by means of computing an exponentially weighted moving average for each neighbor: any node j estimates the average h_i, for any neighbor i, as follows:

$$h_i^+ = \alpha \cdot h_i^- + (1-\alpha) \cdot h_i^{\text{sampled}} \quad (2)$$

where h_i^{sampled} is the last sampled interval between two consecutive transmissions of MSH-DSCH messages by node i, h_i^+ and h_i^- are the new and old estimates, respectively, and α and $(1-\alpha)$ are used to weigh the old and new estimates, respectively. Based on our extensive simulations, the value for $\alpha = 0.1$ produces the most accurate estimations.

D. Grant Allocation

During the request/grant procedure described in Section III-B, a node can grant a number of slots to one of its neighbors. To comply with the schedule horizon discussed in the previous section, these slots are allocated, i.e., the actual grant IE is added to the MSH-DSCH message, as described below. Assume that node G has to grant n slots to its neighbor R. The space where these slots will be granted can be visualized as a three-dimension matrix $B = \{0,1\}^{S \times F \times C}$, where S is the number of slots per frame, F is the number of frames that fall within the schedule horizon for node G to grant slots to node R, and C is the number of available channels. An element $b(s,f,c)$ of matrix B is equal to 1 if it is not possible for G to allocate that slot to R, i.e., if any of the following is true: (i) G (or R) transmits/receives in the s-th slot of the f-th frame of the schedule horizon on any channel; (ii) a neighbor of R transmits in the s-th slot of the f-th frame of the schedule horizon on channel c. Otherwise, it is 0.

The grant allocation thus refers to the problem of "flipping" n elements of B from 0 to 1. Each "flipped" element is a slot granted by node G to R. To reduce the number of grant IEs in the MSH-DSCH messages, the time-contiguous slots should be allocated. In fact, the MSH-DSCH capacity is limited, hence employing too many grant IEs might cause premature termination of the request/grant procedure.[3] This can eventually lead to under-utilization of the channel. Additionally, when R will use the granted slots for transmitting data, after confirmation, it will have to add a physical preamble to each burst of PDUs, i.e., to each time-contiguous allocation of slots. Recall that physical preambles do not carry data, but are only used for synchronizing the granter and the receiver nodes, and can thus be considered as overhead.

It is worth noting that there are some resemblances between the problem of allocating grants described above and that of allocating memory pages in an Operating System (OS), e.g., [22]. In both domains, higher fragmentation entails lower memory/channel utilization. Unfortunately, the existing approaches in the OS literature to reduce fragmentation cannot be directly applied, because it is broadly assumed that pages can be re-arranged, though at some cost, which is not possible in the context of IEEE 802.16. Therefore, we specifically devised an algorithm for the grant allocation problem in IEEE 802.16, whose

[3]We have verified that in all the scenarios simulated, whose results are reported in Section IV, the MSH-DSCH limit has never been reached.

```
A ← {} // allocated slots
f ← 1 // frame under consideration
while ( f ≤ F and #A < n ) {
    C ← {1, ..., C} // all channels
    while ( #C > 0 and #A < n ) {
        c ← random_uniform(C)
        C ← C \ {c}
        for i  ( 1 ... S ) {
            if ( B(i, f, c) ≡ 0 ) A ← A ⋃ B(i, f, c)
            if ( #A ≡ n ) break
    }
    f ← f + 1
}
```

Fig. 3. Pseudo-code of the algorithm to allocate n slots.

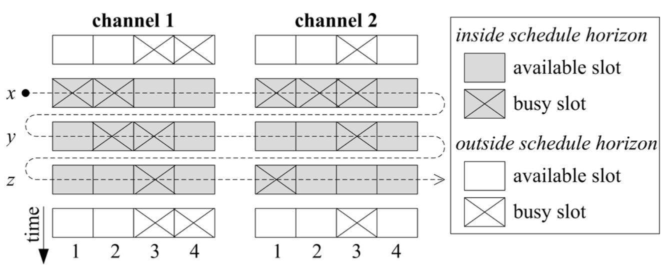

Fig. 4. Example of allocating three slots, with $S = 4, F = 3, C = 2$.

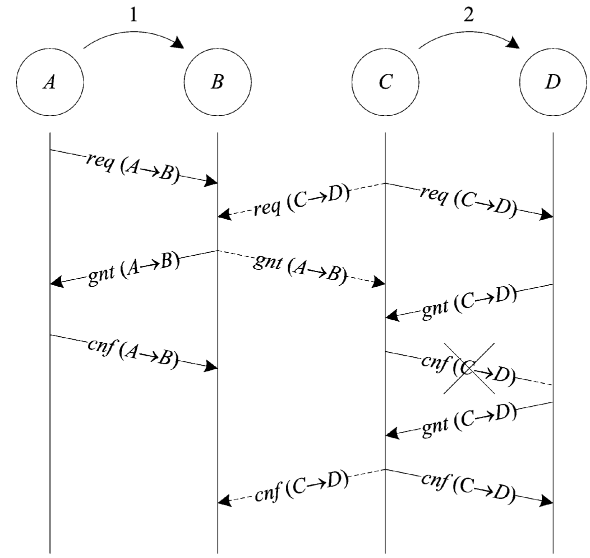

Fig. 5. Information asymmetry example in a chain topology with four nodes. Transmissions overheard by a node, but not directly addressed to that node, are represented as dashed lines.

pseudo-code is reported in Fig. 3.[4] The basic idea of the algorithm is to visit slots in temporal order, while the channel selection is random. The former is aimed at reducing the bandwidth negotiation latency by granting slots as soon as possible. In the latter case, the random channel selection reduces the chance that nodes two hops away, which cannot hear each other, grant the same set of slots to a middle node. This is done by considering one frame at a time (f) from the beginning to the end of the grant horizon, which spans over F frames. Slots are then allocated one at a time into the list $\mathcal{A}$ by finding the earliest slot i of frame f into channel c that has $B(i, f, c)$ equal to zero, where c is drawn randomly among the available channels in a uniform manner.

A simple example of this procedure is illustrated in Fig. 4, where the dashed line represents the visiting order of slots, and the crossed boxes represent slots which cannot be granted because they have already been allocated by the granter or the granter's neighbors. Assume that the granter needs to assign three slots to a neighbor. It first randomly selects one channel between the two available ones, i.e., channel 1 in Fig. 4. Then, it visits the grant horizon from the earliest frame, i.e., x in Fig. 4. Slots 3 and 4 in channel 1 are available, thus, a two-slot grant is issued. Since there is still one slot to be granted, the granter continues searching for available slots. Note that slot 4 in channel 2 cannot be granted anymore, since it overlaps in time with slot 4 in channel 1. Thus, slot 1 of frame y in channel 1 is granted instead, which completes the procedure. To summarize, the following two grant IEs are added to MSH-DSCH: $\langle [3, 4], [x, x], 1 \rangle$ and $\langle [1, 1], [y, y], 1 \rangle$.

[4]Implementation details have been left out for the sake of explanation. For instance, in a real system, nodes need some time to switch from one channel to another, which should be taken into account by nodes when allocating slots.

E. Hidden Node Problem

We described the procedure for scheduling grants and requests in the previous sections. However, as outlined in Section II, the confirmations are also needed for bandwidth negotiation as part of the three-way handshake of the IEEE 802.16 mesh MAC. It is straightforward to assume that nodes should confirm grants whenever possible, otherwise transmission cannot occur in the granted slots. However, a node must refrain from confirming a grant (or part thereof) when another neighbor has already granted, or confirmed, the same set of slots, in such a way that interference would happen on the receiver node. We define this event as *grant withdrawal*. As quantified through simulation in Section IV-B, the grant withdrawal can lead to a significant degradation of the performance, in terms of fairness among traffic flows. This produces an effect that is similar to the "hidden node" problem in CSMA MAC protocols, such as IEEE 802.11 [23]. Therefore, we use the same terminology, even though the cause of the problem in IEEE 802.16 is different from that in CSMA MAC protocols.

Let us consider the example in Fig. 5, where four nodes are arranged in a chain topology, i.e., node A can hear only the transmissions from node B, which in turn can hear only the transmissions from nodes A and C, and so on. Assume that there are two flows: flow 1 from node A to node B, and flow 2 from node C to node D. With a CSMA-based MAC protocol, flow 2 achieves a significantly greater amount of bandwidth than the flow 1. This is because when node A accesses the channel, it is not aware of any ongoing transmissions between C and D. Therefore, node A transmissions are very likely to collide, which reduces flow 1 throughput because of retransmissions and backoff periods. The very same problem does not exist in IEEE 802.16 because the grant/confirm mechanism avoids collisions.

However, let us consider the time diagram of IEEE 802.16 control messages in Fig. 5. Nodes B and D, i.e., the granter nodes, cannot hear each other. It is, thus, possible that their grant allocation procedures select the same slots to be granted to A

and C, respectively, because the scheduling is completely distributed. Should this happen, node C would have to withdraw its grant, as illustrated in Fig. 5, so as to avoid collision with node B. Therefore, the information asymmetry also exists in IEEE 802.16 networks, though in the opposite direction of IEEE 802.11 networks: flow 2 is penalized instead of flow 1.

In FEBA we introduce the *regranting* procedure to mitigate the "hidden node" problem. Whenever a granter node (D in the example) has granted slots to a requester node (C in the example), but the latter does not confirm them entirely in its subsequent MSH-DSCH message, the granter node re-schedules the unconfirmed slots to the requester. The following differences exist between grants and regrants: (i) regrants are provided in an unsolicited manner, without any additional request from the receiver; (ii) regrants are allocated before requests and grants, so that they have a lower chance of not being included in the MSH-DSCH message due to its limited size; and, most importantly, (iii) the schedule horizon for regranting is not overlapping with the schedule horizon, so that regranting does not impair "regular" grant allocation: we define the regranting schedule horizon at time t as $[t + h_R + h_G + 1, t + 2h_R + h_G + 1]$.

In any case, even without regranting, the negative effect due to "hidden nodes" is not as prominent as in CSMA MAC protocols. In fact, the following limiting factors exist. Firstly, sporadic transmissions reduce the chance of the grant withdrawal. Consider again Fig. 5. If node C has the chance of confirming its grant from node D *before* node B issues its grant, then the latter will grant a set of non-overlapping slots to node A, based on the grant allocation procedure described in the previous section. Secondly, grant withdrawals can be reduced by means of advertising availabilities into MSH-DSCH messages, which is optional in the IEEE 802.16 standard. Finally, if multiple channels are available, the random channel selection (see Section III-D) further reduces the chance that neighbors two hops away schedule overlapping grants. For instance, in the example illustrated in Fig. 5, if two channels were available, then node D would have 0.5 probability of scheduling a grant to node C overlapping with that from node B to node A, even though nodes D and B allocated the grant at the same time.

F. Packet Scheduling

After a grant has been confirmed, its slots can be used by the requester for transmitting data. Since grants are issued on a per-node basis, it is necessary that specific packet scheduling policies are implemented so as to provide per-flow fairness. In particular, it is very important that separate transmission buffers are kept for different traffic flows, as confirmed through simulation in Section IV-A. Among several existing packet scheduling algorithms in the literature, we chose the deficit round robin (DRR) [24] algorithm because it achieves fair queueing for variable length packets, can operate at $\mathcal{O}(1)$ complexity, and its implementation is easy. With DRR, each flow is assigned a *quantum* (ψ), which is the amount of bytes, on the average, that it is scheduled at each round when it is backlogged. By assigning different *quanta* it is possible to differentiate services. Specifically, we compute the *quantum* of traffic flow i (ψ_i) as

$$\psi_i = \frac{w_i}{\sum_j w_j} F_{\mathrm{DRR}}$$

where F_{DRR} is the target DRR duration and is set like the target round duration F_{RR} of the request/grant procedure described in Section III-B. The ψ values have to be updated whenever the set of active flows changes. Finally, note that while only one request/grant procedure is instanced per node, each node has one packet scheduler per logical link/neighbor.

G. Implementation Issues

First, FEBA requires that all nodes keep track of the flows that are currently active. This increases the spatial and temporal computational complexity of the algorithm, which is a classical problem of flow-based architectures. However, wireless mesh routers are expected to have high computation capabilities and the number of traffic flows traversed by any node is expected to be relatively small. Therefore, we do not consider keeping the state of all active traffic flows to be an issue for a real implementation of IEEE 802.16 devices. Instead, this issue makes FEBA hardly suitable for the application in other networks, where either the number of active flows is high, e.g., backbone Internet routers, or the computational capabilities of nodes are low, e.g., wireless sensor networks.

Another non-trivial real-time task that needs be performed by nodes is computing the matrix B for each grant that is assigned to a neighbor (see Section III-D). However, this operation can be done efficiently via *grant bitmaps*, which are updated at the reception of any MSH-DSCH message. A grant bitmap keeps track of all grants and confirmations that are listened by a node, even though the node itself is not involved in the bandwidth negotiation. Since the data sub-frame is divided into fixed duration slots, this is as simple as directly setting one or more entries to 1 in a vector of bits, where each element represents a forthcoming slot for data transmission. Note that one grant bitmap per channel is required. The matrix B is then derived when assigning grants by performing a logical OR operation on the receiver's neighbors, which can be executed very efficiently even without specialized hardware.

In regard to assigning priority levels to traffic flows, this can be easily achieved in IEEE 802.16 by means of the priority field that is specified in the standard MAC header. The priority field consists of 3 bits, which allows up to eight different traffic flows for each node. If each priority is mapped to a different type of service, the space of priorities is large enough to accommodate existing classes of services, even larger than that allowed by the competing WMN technologies, such as IEEE 802.11s.[5] Furthermore, since the IEEE 802.11d standard [25] specifies eight priority levels, it is possible to bridge Quality of Service (QoS) enabled heterogeneous LANs through an IEEE 802.16 WMN.

[5]The draft IEEE 802.11s specifies the Enhanced Distributed Channel Access (EDCA) as the default medium access function for differentiated service, which allows up to four classes of service.

Finally, recall that the three-way handshake procedure of the IEEE 802.16 mesh MAC guarantees collision-free transmissions only when the "protocol-model" assumption holds. While this assumption can be acceptable in planned WMNs, where the objective of the network operators is to deploy wireless mesh routers in order to maximize spatial re-use while limiting interference, still counter-measures should be devised to deal with those cases where the assumption does not hold.

A possible solution is to provide the ad-hoc routing agent with cross-layer information about MAC layer measurements in terms of the links that are more subject to interference than the others. Since the IEEE 802.16 MAC enables reliability by means of automatic repeat request (ARQ) mechanisms, this can be as simple as estimating the success probability of PDUs for each link. The existence of "weak" links can then be conveyed to the routing agent, which will assign low weight or reliability values to them, depending on which path selection algorithm is used. Cooperation between the routing and MAC layers can also provide other benefits, e.g., broadcast messages could be sent in control slots so as to ensure that they are transmitted with the most efficient MCS and that all the nodes in a neighborhood receive them, even though multiple channels are used. However, we consider these routing protocol issues to be outside the scope of this work.

In any case, we note that the links that are more likely to experience high interference are those employing less efficient MCSs, which should be avoided anyway by the path selection algorithm due their low transmission efficiency. Recall that any node in the IEEE 802.16 WMN is required to employ a procedure to adapt the MCS used for data transmission on each of its links. While the exact algorithm is not specified by the standard, it is clear that the procedure should be aimed at using the most efficient MCS, while guaranteeing the packet error rate to be smaller than a given threshold. The used MCS can then be considered as a function of the interference, e.g., measured in terms of the Signal-to-Noise Ratio (SNR): the greater is the SNR at a receiver node, the more efficient is the MCS selected by the sender for transmission. Instead, the control transmissions are required to use the most robust MCS, i.e., the QPSK modulation with code rate 1/2. Since the logical links are created by means of control messages, neighborhood relationships are established among nodes even though they experience the smallest SNR values, e.g., because they are far away from each other. Therefore, by ensuring that no simultaneous transmissions from neighbors occur in the same channel while a node is receiving data, the three-way handshake procedure implicitly protects ongoing transmissions at a high degree with more efficient MCS than that used for control messages.

IV. Performance Evaluation

In this section we present extensive simulation results obtained by varying several network parameters and workload configurations. The values of the network parameters used in simulations are those specified in the IEEE 802.16 standard as system profile *profP3_10*[11]. Specifically, the channel bandwidth is 10 MHz, with a frame duration of 4 ms, including both control and data sub-frames. The *XmtHoldoffExponent* has been set to 0. Unless otherwise stated, there are four control slots per frame, and the nodes employ the lowest modulation, i.e., QPSK-1/2, which yields a raw physical bandwidth of about 6.5 Mb/s per channel. We do not take channel errors into account, which allows us to focus specifically on the system performance at the MAC layer.

We have implemented the IEEE 802.16 mesh mode with coordinated scheduling in the *ns2* network simulator.[6] The source code of our implementation is publicly available as *open source* software.[7] Unless otherwise specified, each traffic flow has a separate 100 kB buffer. The simulation output evaluation has been carried out using the method of independent replications [26]. For each scenario, the number of replications, the initial warm-up period and the simulation duration were tuned to produce accurate estimations, i.e., whose 95% t-student confidence interval is smaller than or equal to 1/10 of the estimated value [27]. Confidence intervals are not drawn whenever negligible.

A. Bandwidth Sharing

Here we compare FEBA to a *Greedy* approach where each node requests and grants as much bandwidth as possible at each turn. Additionally, we compare the DRR algorithm for packet scheduling to a First-In-First-Out (FIFO) scheduler where all the SDUs with the same next-hop are enqueued into the same buffer. For this purpose we simulate a network with an increasing number of nodes, from 2 to 10, arranged in a *chain* topology. Each node has one traffic flow directed to the chain end-point node, carried with a constant bit-rate stream of 1000 bytes packets emulating infinite bandwidth demands.

We first consider the single-channel case. In Fig. 6 we show the *end-to-end throughput* (or *throughput*, for short), which is defined as the number of bits received by the destination node per second for a given traffic flow, without any MAC overhead. As it can be seen, the throughput steeply decreases as the number of nodes increases, regardless of the scheme adopted. This is because an increasing fraction of the channel capacity is employed to relay packets at intermediate nodes. For instance, with three nodes the end-to-end FEBA/DRR throughput is about 2/3 of the available raw bandwidth: 1/3 is consumed by the traffic flow that is one hop from the destination, and 2/3 are consumed by the other one that has a length of two hops. This way they both achieve the same throughput. Moreover, FEBA/DRR achieves the highest throughput compared to other schemes which are demonstrated in Fig. 6. With Greedy, most wireless resources are employed to transmit data to neighbors. This creates many bottlenecks along the network, which obstruct those traffic flows that are farther from the destination node, hence severely degrade their throughput performance. In fact, a large amount of data transmitted over the wireless channel is dropped by intermediate nodes due to buffer overflow. This phenomenon has been observed in the context of IEEE 802.11 WMNs also, e.g., in [5]. Such an undesirable situation is prevented by FEBA which has a flow-based architecture. However, the bandwidth allocation alone is effective only when the number of nodes is smaller than six. This limit is overcome by coupling FEBA with DRR packet scheduling,

[6] http://www.isi.edu/nsnam/ns/

[7] http://cng1.iet.unipi.it/wiki/index.php/Ns2mesh80216

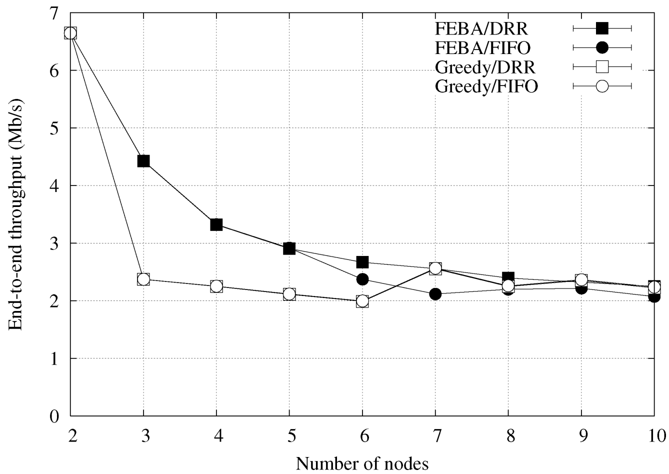

Fig. 6. *Bandwidth sharing.* Sum of the throughput of all traffic flows.

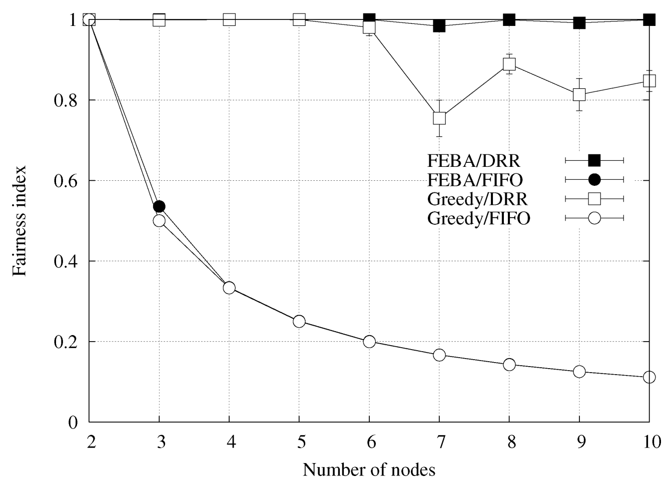

Fig. 7. *Bandwidth sharing.* Throughput fairness index.

which in fact over-performs FEBA/FIFO in the full range of nodes considered in this scenario.

We now consider the fairness of the bandwidth allocation, by means of the index defined in [28] as the ratio $\left(\sum_{i=1}^{n} x_i\right)^2 / \left(n \sum_{i=1}^{n} x_i^2\right)$, where n denotes the number of traffic flows, and x_i the throughput of the i-th traffic flow. Note that, by definition, the fairness index ranges between $1/n$ (worst case, when only one traffic flow is assigned a positive amount of bandwidth) and 1 (best case, when the bandwidth is partitioned evenly among all traffic flows). Fig. 7 shows the fairness index computed over time windows of 100 ms. As it can be seen, FEBA/DRR is the only combination to achieve almost perfect fairness among all the traffic flows in all cases. On the other hand, Greedy/DRR is only fair while the number of nodes is relatively small, i.e., below seven. Finally, the packet scheduler FIFO exhibits very poor fairness even with three nodes, regardless of the bandwidth allocation algorithm. This is because SDUs that need to be relayed are enqueued at the same buffer that stores local SDUs, which are generated at a rate much higher than the arrival rate from neighbors.

We now repeat the same scenario above, with nodes employing 16-QAM-1/2 and 64-QAM-2/3 MCSs and with two channels. The sum of the throughput of all flows is given in

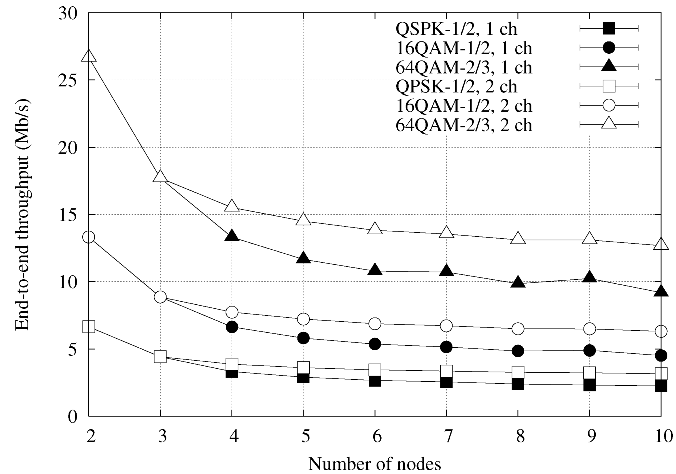

Fig. 8. *Bandwidth sharing.* Sum of the throughput of all traffic flows using different modulations, with a single channel (1 ch) and two channels (2 ch).

Fig. 8. In the single-channel case the throughput is proportional to the MCS efficiency: the QPSK-1/2 throughput is about half of the 16-QAM-1/2 one, which in turn is about half of that with 64-QAM-2/3. On the other hand, adding a second channel does *not* double the throughput. This is because all nodes have a single radio, thus data transmissions between different pairs of nodes at the same time, on different channels, can only occur between disjoint pairs. For example, with three nodes only, it is not possible to exploit the second channel because the middle node always takes part in the communication. Adding further channels, in this case, does not improve the system performance significantly.

B. Hidden Node Effect

In this section we quantitatively evaluate the "hidden node" effect in IEEE 802.16. To this aim we analyze the reference scenario illustrated in Fig. 5 (see Section III-E), i.e., four nodes arranged in a chain topology labeled from A to D, with one flow established from A to B and another from C to D. In this scenario transmission from node C to D interferes with transmission from node A to B, while the opposite is not true. The performance obtained employing the regranting procedure described in Section III-E is compared to that in the case where regranted slots are allocated in the same schedule horizon used for "regular" grant scheduling (labeled as 'regranting*'). The results without regranting are also reported. In this case we also show the performance gain obtained by employing an additional channel and by advertising availabilities in MSH-DSCH messages, respectively.

In Fig. 9 we show the fairness index of throughput when bandwidth demands of the two flows increase from 1 Mb/s to 8 Mb/s each, by means of constant rate generation of 1000 bytes packets. Since there are only two flows, the fairness index ranges from 0.5 (lowest) to 1 (highest). As it can be seen, with regranting the fairness is close to 1, even at high loads. This desirable property is partly achieved by assigning regranted slots into a separate schedule horizon. In fact, the regranting* curve falls below 0.7 at high loads, i.e., when the offered load is greater than or equal to 10 Mb/s. This can be explained as follows. At

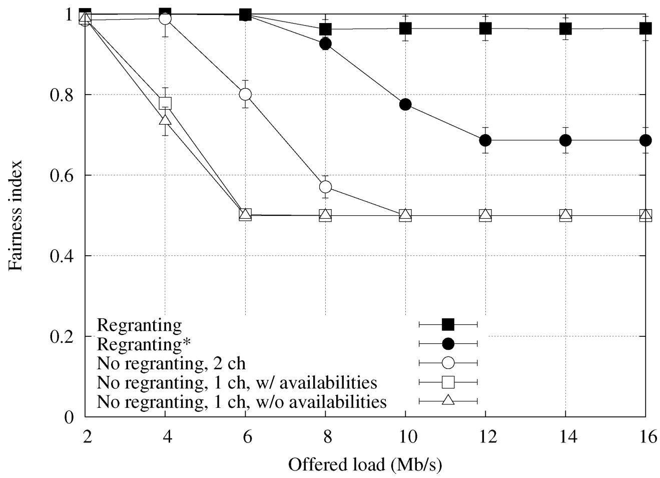

Fig. 9. *Hidden node effect.* Throughput fairness index.

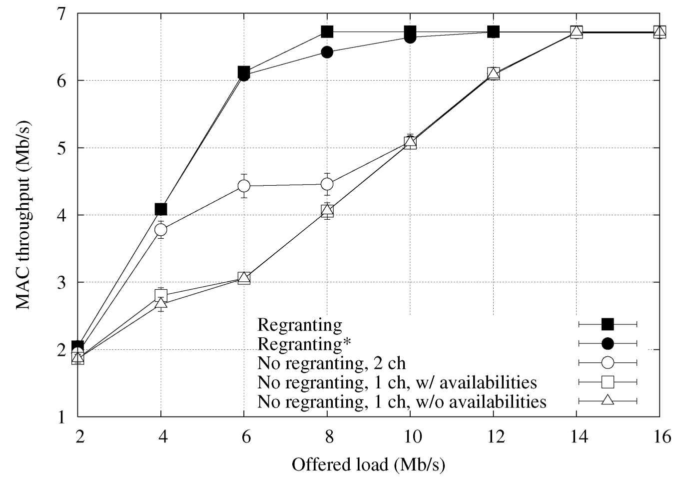

Fig. 10. *Hidden node effect.* Sum of the throughput of all traffic flows.

low offered loads there are, on the average, several slots in the data sub-frame not used for data transmission. These slots can be exploited effectively by node D for regranting to node C (see Fig. 5). When the network becomes saturated, i.e., the offered load increases beyond the channel capacity, the number of unused slots in the schedule horizon decreases significantly, which increasingly starves the regranting process at node D. This effect is substantially mitigated by using a separate horizon for regranting, which in fact achieves almost perfect fairness.

In any case, the "hidden node" effect becomes most noticeable without regranting, which reaches lowest fairness as soon as the offered load becomes equal to 6 Mb/s, which is close to the channel capacity. Furthermore, while, in principle, advertising availabilities in MSH-DSCH messages can mitigate the "hidden node" effect, in practice the performance gain is negligible. Finally, adding a second channel for data transmission enhances the fairness, but only up to a limited degree.

We now show that the fairness improvement with regranting is not gained at the expense of the throughput, which is reported in Fig. 10. As a matter of fact, the regranting curve lies above all others: the higher is the fairness among the two traffic flows, the greater is the throughput of the traffic flows. Therefore, there is no trade-off between throughput and fairness, as it is often the case in WMNs, because regranting allows bandwidth to be saved, with respect to alternative approaches, by decreasing the number of slots that are left unallocated due to unwise granting.

C. Multi-Channel

In this scenario we evaluate the performance improvement, in terms of throughput, due to the use of multiple channels in a densely populated network. Specifically, we consider a network with 9 nodes, with an increasing number of neighbors, from 2 to 8. Each node has a traffic flow with infinite demands towards one of its neighbors.

In Fig. 11 we show the sum of the throughput of all traffic flows, with one to four available channels. As it can be seen, all the curves decrease when the network density increases, because the spatial reuse decreases. In fact, the more dense is a network, the higher is the number of nodes that compete for granting bandwidth in interfering links. However, in this

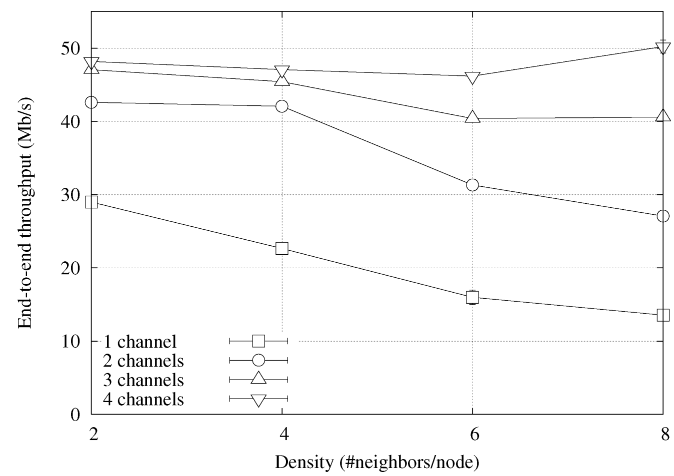

Fig. 11. *Multi-channel.* Sum of the throughput of all traffic flows, from 1 to 4 channels.

scenario, employing multiple channels greatly improves the network throughput, since it allows nodes to exploit frequency reuse. This is especially true with a completely connected network, i.e., 8 neighbors/node, where every node receives all the MSH-DSCH messages advertised by neighbors and regranting, thus, never occurs.

D. Bursty Traffic

We now evaluate the isolation among traffic flows with bursty traffic, in terms of the average end-to-end delay. Specifically, we define the end-to-end delay (or *delay*) of a packet as the time interval between the arrival time of this packet at the network layer of the sender node, and the time when this packet is completely delivered to the network layer at the destination node. Packet generation of each traffic flow is based on the procedure proposed in [29] as the reference traffic model for characterizing the performance of IEEE 802.16 networks: a best-effort traffic flow is built from the super-imposition of four Interrupted Poisson Processes (IPPs), generating packets with a constant size of 192 bytes. The average offered load for each traffic source is about 125 kb/s. The network under investigation consists of 19 nodes arranged in a tree topology. Let a traffic flow entering (leaving)

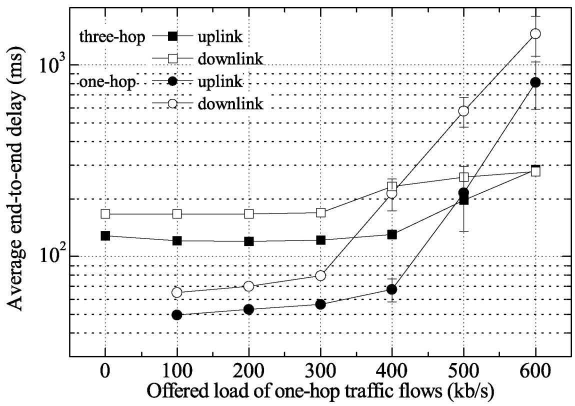

Fig. 12. *Bursty traffic.* Average end-to-end delay.

the root be a *downlink* (*uplink*) flow. Each neighbor of the root node has a bi-directional traffic flow towards the root at a variable offered load, while nodes three hops away from the root have a bi-directional traffic flow at 100 kb/s established with it.

In Fig. 12 we show the average delay of one-hop and three-hop traffic flows, in the uplink and downlink direction, respectively. We start from a scenario in which there are three-hop flows only and increase the network load by injecting one-hop flows at an increasing offered load of 100 kb/s up to 600 kb/s.

While the network is lightly loaded, i.e., the offered load of one-hop traffic flows is below 400 kb/s, the average delay of three-hop traffic flows is almost constant, but significantly higher than that of one-hop flows. This is due to the three-way bandwidth negotiation at each hop. Both for one- and three-hop flows, the uplink delays are below the downlink delays, which is due to the root node being heavily loaded. When the network becomes overloaded, i.e., the offered load of one-hop flows is above 400 kb/s, then the links between the root and its neighbors become bottlenecks, which slow down both the one-hop and the three-hop flows and cause longer delays, as shown in Fig. 12. Note that the one-hop delays increase steeply when the offered load is higher than 400 kb/s, which is caused by the channel being occupied, i.e., the bandwidth exhausted, and queues being created accordingly.

E. *Differentiated Service*

We now evaluate the ability of IEEE 802.16 with FEBA to provide multimedia streams with differentiated services with respect to best-effort (BE) traffic flows. For the latter we use the same traffic model described in Section IV-D, while two types of multimedia streams are considered: Voice over IP (VoIP) and Video on Demand (VoD), which are characterized below. The same network topology is used with both VoIP and VoD, which consists of a grid of 5×5 nodes, where each edge node also has a logical link with the node at the opposite side of the grid. Therefore, the resulting connectivity graph is a toroid, which guarantees that all nodes have the same number of neighbors and thus prevents edge effects from showing in results. In each scenario, the number of real-time (RT) traffic flows, either VoIP or VoD, established is equal to 10, but the source and destination nodes are selected randomly. Instead, the number of best effort flows is increased by 10 flows at a time to inflate the inter-class interference. Like multimedia traffic, source and destination nodes of BE traffic flows are drawn randomly.

VoIP is analyzed first, with QPSK-1/2 modulation used on all links. A VoIP call is simulated by means of an ON/OFF process, with silence and talkspurt periods distributed according to Weibull distributions as specified in [30]. The packet size and generation interval are selected according to the GSM AMR specifications, which is one of the most employed codec in wireless cellular networks and yields an offered load of 12.8 kb/s during talkspurts. For VoIP traffic, we used the Mean Opinion Score (MOS), between 1 (unbearable quality) and 5 (best quality), as the performance index [31]. To remove the effect of imperfect playout buffering at the VoIP receiver, an optimal (non-causal) algorithm is implemented in the simulator to select the playout time of VoIP frames.[8]

In Fig. 13 we show the average MOS of VoIP calls when the offered load of BE traffic increases from 0 to 14 Mb/s. Measures were collected separately for VoIP calls that have different path lengths. Only the limit cases of one- and four-hop VoIP calls are reported. Two- and three-hop calls experienced intermediate quality. Three different ratios $w_{\mathrm{RT}}/w_{\mathrm{BE}}$ between the RT and BE weights are considered, from 1 to 100. Results when VoIP calls are conveyed by traffic flows with the same priority level as BE are also reported as a reference. When there are only VoIP traffic flows in the network, i.e., the BE offered load is 0, the call quality does not depend on the differentiation policy. However, without differentiation, as soon as BE traffic flows are injected into the network, the MOS significantly decreases, which is especially true for four-hop VoIP calls, whose quality is totally unacceptable with 5.6 Mb/s or higher BE load. This is because VoIP traffic flows that are further from the destination have to contend more often with BE traffic than those with smaller path lengths. VoIP performance slightly improves when RT and BE flows are conveyed in separate traffic flows, because of buffer separation at the MAC packet schedulers (see Section III-F). However, even better quality is achieved by increasing $w_{\mathrm{RT}}/w_{\mathrm{BE}}$ beyond 1. In fact, when the ratio is equal to 100, the VoIP quality of one-hop flows is almost insensitive to the amount of BE traffic load in the network, and that of four-hop flows degrades only slightly.

Moreover, in Fig. 14 we report the sum of the throughput of BE traffic flows, i.e., the BE carried load. As it can be seen, the curves almost coincide. This means that providing prioritized service to VoIP traffic does not impair significantly the ability of the network to serve BE traffic. Therefore, tuning the ratio between the traffic flow weights in FEBA provides an efficient mechanism to provide VoIP calls with differentiated service.

We conclude this section by analyzing VoD streaming. In this scenario we configure nodes to employ the 16-QAM-3/4 modulation for data transmission, otherwise only a very small number of VoD traffic flows would have been allowed due to its higher bit-rate than VoIP. VoD traffic is simulated via a pre-encoded MPEG4 trace of the movie Jurassik Park [32], with 33 ms frame generation interval and average (peak) rate equal to 1 Mb/s (6 Mb/s). The per-flow MAC buffer size was also in-

[8]URL: http://cng1.iet.unipi.it/wiki/index.php/Ns2voip.

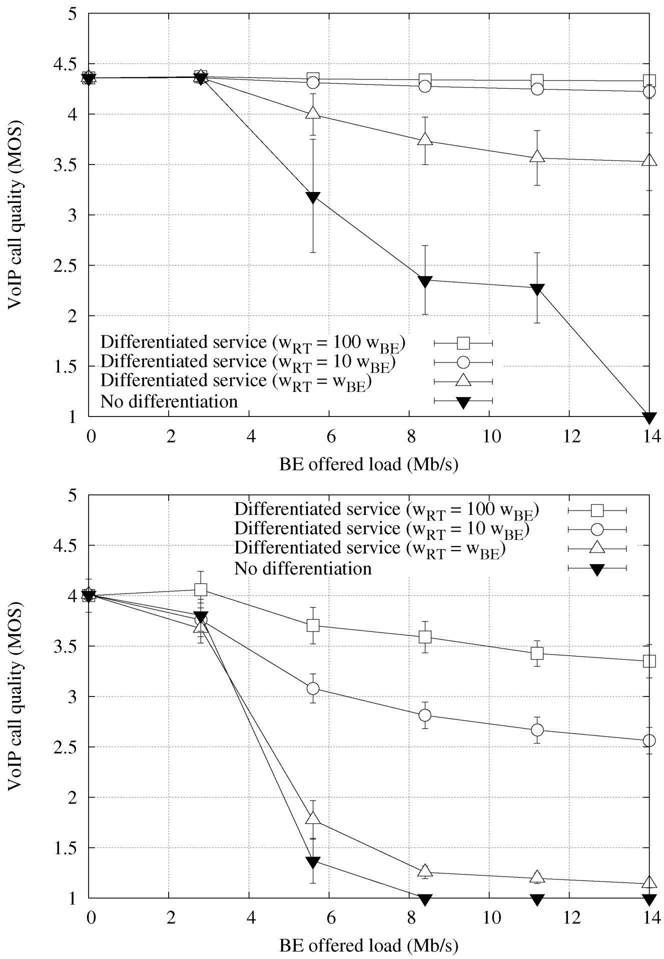

Fig. 13. *Differentiated service.* Mixed VoIP and BE traffic. Average MOS of VoIP calls (top: one-hop length calls; bottom: four-hop length calls).

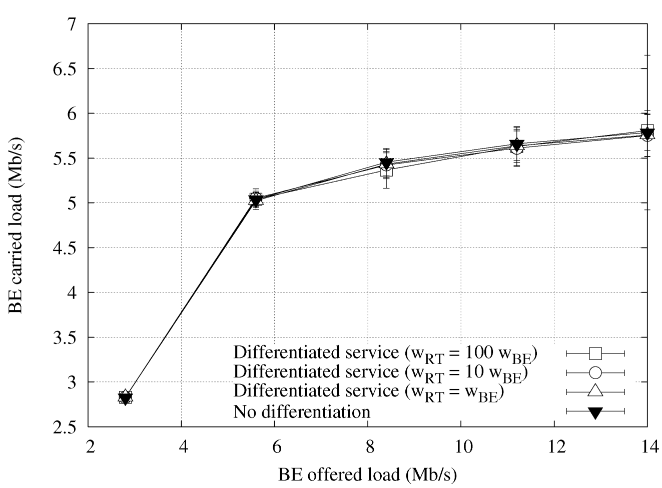

Fig. 14. *Differentiated service.* Mixed VoIP and BE traffic. Sum of the throughput of BE traffic flows.

creased to 300 kB to avoid unnecessary buffer overflows. In the following we measure the VoD performance by means of the *cell outage* metric, defined as the ratio between the number of VoD traffic flows that experience a frame loss greater than or equal to 0.05 and the total number of VoD traffic flows established in the WMN. In turn, the *frame loss* is defined as the ratio between the

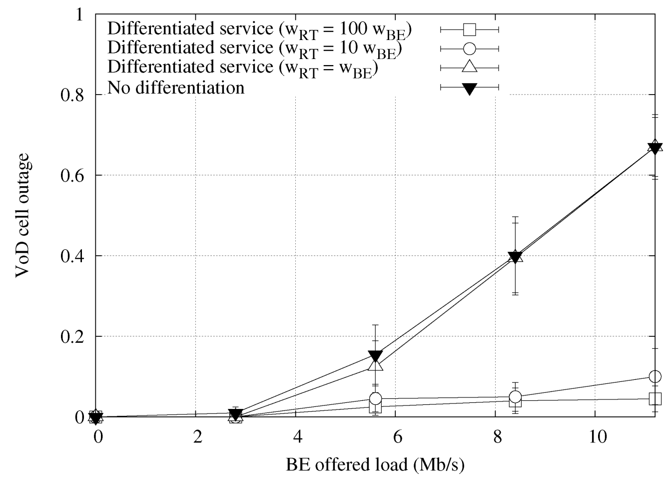

Fig. 15. *Differentiated service.* Mixed VoD and BE traffic. Cell outage with 5% frame loss threshold.

number of video frames correctly decoded and the total number of frames generated. Note that not all received video frames can be actually decoded, because prediction dependencies of video frames have to be met, e.g., a B-type video frame cannot be decoded if the preceding P- or I-frame is lost.

In Fig. 15 we show the cell outage when the BE offered load increases from 0 to 11.2 Mb/s. As it can be seen, when the BE offered load becomes greater than or equal to 5.6 Mb/s the curves without differentiation and with $w_{\mathrm{RT}}/w_{\mathrm{BE}} = 1$ increase sharply, while the others are relatively low. With 11.2 Mb/s, about 4% of the VoD traffic flows experience a frame loss greater than 5%, while this number grows up to 70% without differentiating between VoD and BE traffic flows.

However, unlike with VoIP, the performance gain of VoD flows comes at the expenses of the throughput of BE flows. In fact, as shown in Fig. 16, the BE carried load is greater for those cases where the VoD performance is worse. This is because the high cell outage is due to video frames getting lost due to buffer overflows, which enables the BE traffic to use more wireless resources. This behavior was not noticeable with VoIP because the latter consumes a negligible amount of bandwidth (about 165 kb/s) compared to the offered load of VoD (about 10 Mb/s).

V. Related Work

Even though the mesh mode of IEEE 802.16 has been released in 2004, the literature so far lacks substantial work in this context, especially for the distributed coordinated mode, which is the subject of this study. To the best of our knowledge, the existing works in this context are focused only on the distributed election procedure run by nodes to access the control slots. In [12] a model is proposed for the distributed election procedure under simplified assumptions, while a simulation study was carried out in [33]. Based on these results, the interval between two consecutive accesses to the control slots can become fairly large in dense WMNs, which can eventually lead to poor network utilization. A solution to this problem is proposed in [34], based on dynamic adaptation of the *XmtHoldoffExponent* system parameter.

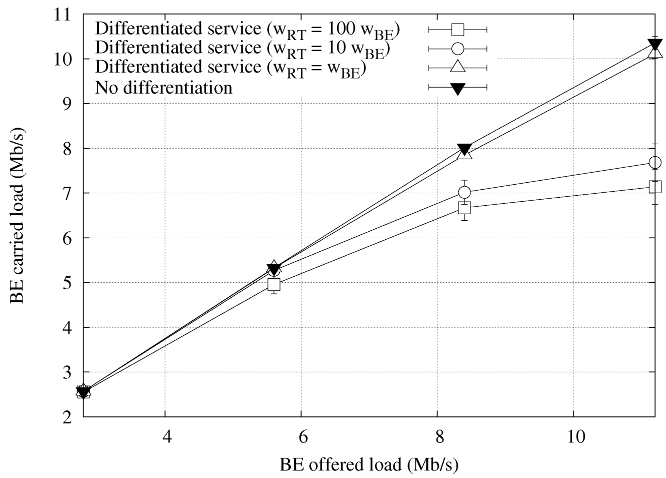

Fig. 16. *Differentiated service.* Mixed VoD and BE traffic. Sum of the throughput of BE traffic flows.

On the other hand, there are some works addressing the scheduling problem in WMNs based on the IEEE 802.16 mesh MAC with centralized scheduling. The scheduling problem is reduced to a linear programming problem in [35], where the goal is to maximize the network throughput while enforcing fairness among traffic flows. An efficient solution to solve the problem is also described and evaluated via simulation. Another solution has been proposed in [36], where resources are scheduled in two phases: first the demands of the nodes are collected by the BS and flooded throughout the WMN, then each node performs local link allocation enforcing transmissions to be collision-free in any neighborhood. A joint routing and scheduling algorithm is devised in [37], which takes into account the number of interfering nodes of each link. Finally, in [38] the distributed scheduling features of IEEE 802.16 have been shown to enhance the overall performance when used in combination with centralized scheduling, for which two baseline algorithms are provided.

Furthermore, using the point-to-multipoint (PMP) mode of IEEE 802.16 to deploy WMNs has been suggested in [39], which proposes a framework for activating wireless links so as to limit interference. Efficient centralized scheduling algorithms and an admission control scheme have been proposed in the same context in [40].

VI. Conclusion

In this paper we have presented FEBA, a distributed algorithm for bandwidth balancing in multi-channel IEEE 802.16 WMNs. FEBA is specifically tailored to solve the problem of unfairness among traffic flows with different path lengths, which otherwise affects WMNs. Also, differentiated services, in terms of throughput, are provided to traffic flows with different priorities. To this aim, bandwidth requesting and granting is carried out in a round-robin fashion, where the amount of service at each round is proportional to the number of incoming or outgoing flows, appropriately weighted according to the flows' priorities. Additionally, we substantially mitigate the "hidden node" problem by means of *regranting*.

We have shown that FEBA proves to be effective in providing end-to-end traffic flows with fair medium access, with respect to alternative approaches which assign bandwidth in a greedy manner and/or employ a FIFO algorithm to schedule packets. Furthermore, we have shown that the frequency diversity is exploited by FEBA to increase the network capacity, hence the achievable throughput. The effect of multi-channel is especially significant in densely populated networks, with low spatial reuse. Finally, we have simulated scenarios with mixed best-effort and real-time traffic, with realistic VoIP and VoD traffic models. Results have shown that service differentiation can be provided by FEBA by appropriately tuning the relative weights of service classes. This eventually leads to multimedia streams being provided with adequate QoS in IEEE 802.16 WMNs.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Comput. Netw.*, vol. 47, pp. 445–487, Mar. 2005.

[2] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering wireless systems with multiple radios," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 39–46, Oct. 2004.

[3] A. Raniwala and T. Chiueh, "Architecture and algorithms for and IEEE 802.11-based multi-channel wireless mesh network," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 2223–2234.

[4] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "IEEE 802.11b ad hoc networks: Performance measurements," *Cluster Computing*, vol. 8, pp. 135–145, Jul. 2005.

[5] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-end performance and fairness in multihop wireless backhaul networks," in *Proc. ACM MobiCom*, Philadelphia, PA, Sep. 2004, pp. 287–301.

[6] T. Salonidis and L. Tassiulas, "Distributed on-line schedule adaptation for balanced slot allocation in wireless ad hoc networks," in *Proc. IEEE IWQoS*, Montreal, Canada, Jun. 2004, pp. 20–29.

[7] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint asynchronous congestion control and distributed scheduling for multi-hop wireless networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12 (DOI: 10.1109/INFOCOM.2006.210).

[8] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ESS Mesh Networking. IEEE 802.11s/D1.06, 2007.

[9] R. Zhao, B. Walke, and G. R. Hiertz, "An efficient IEEE 802.11 ESS mesh network supporting quality-of-service," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2005–2017, Nov. 2006.

[10] X. Wang and A. O. Lim, "IEEE 802.11s wireless mesh networks: Framework and challenges," *Ad Hoc Netw.*, 2007, Doi:10.1016/j.adhoc.2007.09.003.

[11] Air Interface for Fixed Broadband Wireless Access Systems. IEEE 802.16d, 2004.

[12] M. Cao, W. Ma, Q. Zhang, and X. Wang, "Analysis of IEEE 802.16 mesh mode scheduler performance," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1455–1464, Apr. 2007.

[13] C. Cicconetti, I. F. Akyildiz, and L. Lenzini, "Bandwidth balancing in multi-channel IEEE 802.16 wireless mesh networks," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 2108–2116.

[14] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 388–404, Mar. 2000.

[15] Q. Dong, S. Banerjee, and B. Liu, "Throughput optimization and fair bandwidth allocation in multi-hop wireless LANs," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12 (DOI 10.1109/INFOCOM.2006.247).

[16] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[17] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 9, pp. 1024–1039, Sep. 1991.

[18] Y. Xue, B. Li, and K. Nahrstedt, "Optimal resource allocation in wireless ad hoc networks: A price-based approach," *IEEE Trans. Mobile Comput.*, vol. 5, no. 4, pp. 347–364, Apr. 2006.

[19] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *Proc. IEEE INFOCOM*, Jun. 2002, pp. 763–772.

[20] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," *IEEE/ACM Trans. Networking*, vol. 15, no. 1, pp. 133–144, Feb. 2007.

[21] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 473–489, Aug. 1999.

[22] M. A. Nichols, H. J. Siegel, H. G. Dietz, R. W. Quong, and W. G. Nation, "Eliminating memory for fragmentation within partitionable SIMD/SPMD machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 2, no. 7, pp. 290–303, Jul. 1991.

[23] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proc. ACM MobiCom*, Boston, MA, Aug. 2000, pp. 87–98.

[24] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," *IEEE/ACM Trans. Networking*, vol. 4, no. 3, pp. 375–385, Jun. 1996.

[25] Media Access Control (MAC) Bridges. IEEE 802.1d, 2004.

[26] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*. New York: McGraw-Hill, 2000.

[27] C. Cicconetti, E. Mingozzi, and G. Stea, "An integrated framework for enabling effective data collection and statistical analysis with ns-2," in *Proc. WNS2*, Pisa, Italy, Oct. 2006, http://doi.acm.org/10.1145/1190455.1190466.

[28] R. Jain, W. Hawe, and D. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corp., New York, Tech. Rep. DEC-TR-301, 1984.

[29] C. R. Baugh, J. Huang, R. Schwartz, and D. Trinkwon, "Traffic model for 802.16 TG3 MAC/PHY simulations," Tech. Rep., IEEE 802.16 Broadband Wireless Access Working Group, 2001.

[30] C.-N. Chuah and R. H. Katz, "Characterizing packet audio streams from Internet multimedia applications," in *Proc. IEEE ICC*, New York, Apr. 2002, pp. 1199–2203.

[31] Transmission Impairments due to Speech Processing. ITU-T Recommendation G.113, 2001.

[32] F. H. P. Fitzek and M. Reisslein, "MPEG4 and H.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, Nov. 2001.

[33] C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi, "Performance evaluation of the mesh election procedure of IEEE 802.16/WiMAX," in *Proc. IEEE MSWiM*, Chania, Crete, Greece, Apr. 22–26, 2007, pp. 323–327.

[34] N. Bayer, B. Xu, V. Rakocevic, and J. Habermann, "Improving the performance of the distributed scheduler in IEEE 802.16 mesh networks," in *Proc. IEEE VTC-Spring*, Apr. 22–25, 2007, pp. 1193–1197.

[35] M. Cao, V. Raghunathan, and P. Kumar, "A tractable algorithm for fair and efficient uplink scheduling of multi-hop WiMAX mesh networks," in *Proc. IEEE WiMesh*, 2006, pp. 93–100.

[36] D. Kim and A. Ganz, "Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems," in *Proc. Broadnets*, Oct. 3–7, 2005, pp. 833–839.

[37] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. Haas, "Interference-aware IEEE 802.16 WiMax mesh networks," in *Proc. IEEE VTC-Spring*, May-Jun. 30–1, 2005, vol. 5, pp. 3102–3106.

[38] S.-M. Cheng, P. Lin, D.-W. Huang, and S.-R. Yang, "A study on distributed/centralized scheduling for wireless mesh network," in *Proc. IWCMC*, Vancouver, BC, Canada, Jul. 2006, pp. 599–604.

[39] G. Narlikar, G. Wilfong, and L. Zhang, "Designing multihop wireless backhaul networks with delay guarantees," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12 (DOI 10.1109/INFOCOM.2006.319).

[40] D. Ghosh, A. Gupta, and P. Mohapatra, "Admission control and interference-aware scheduling in multi-hop WiMAX networks," in *Proc. IEEE MASS*, Oct. 8–11, 2007, pp. 1–9.

Claudio Cicconetti (M'07) graduated in computer systems engineering from the University of Pisa, Pisa, Italy, in October 2003. He received the Ph.D. degree in information engineering from the same University in July 2007.

His main research interests are quality of service support at the MAC layer in cellular and ad hoc wireless networks and network simulation tools.

Dr. Cicconetti served as co-chair of NSTools'07 and QoSim'08 and as workshop co-chair of Valuetools'08.

Ian F. Akyildiz (M'86–SM'89–F'96) received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981, and 1984, respectively.

Currently, he is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, and Director of Broadband Wireless Networking Laboratory. Since June 2008, he is an Honorary Professor with the School of Electrical Engineering at the Universitat Politecnico de Catalunya, Barcelona, Spain. His current research interests are in wireless mesh networks, cognitive radio networks, wireless sensor networks, and nano-communication networks.

Dr. Akyildiz is the Editor-in-Chief of *Computer Networks (COMNET) Journal* (Elsevier) as well as the founding Editor-in-Chief of both the *Ad Hoc Networks Journal* (Elsevier) and *Physical Communication (PHYCOM) Journal* (Elsevier). He received the Don Federico Santa Maria Medal for his services to the Universidad of Federico Santa Maria in 1986. From 1989 to 1998, he served as a National Lecturer for ACM and received the ACM Outstanding Distinguished Lecturer Award in 1994. He received the 1997 IEEE Leonard G. Abraham Prize Award (IEEE Communications Society) for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architectures" published in the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS (JSAC) in January 1996. He received the 2002 IEEE Harry M. Goode Memorial Award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking." He received the 2003 IEEE Best Tutorial Award (IEEE Communication Society) for his paper entitled "A Survey on Sensor Networks," published in *IEEE Communications Magazine*, in August 2002. He also received the 2003 ACM Sigmobile Outstanding Contribution Award with the citation "for pioneering contributions in the area of mobility and resource management for wireless communication networks." He received the 2004 Georgia Tech Faculty Research Author Award for his "outstanding record of publications of papers between 1999–2003." He also received the 2005 Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech. He has been a Fellow of the Association for Computing Machinery (ACM) and IEEE Fellow since 1996.

Luciano Lenzini (M'07) received the degree in physics from the University of Pisa, Pisa, Italy.

He joined CNUCE, an institute of the Italian National Research Council (CNR) in 1970. From 1973 to 1974, he worked on computer networks at the IBM Scientific Center, Cambridge, MA. He has since directed several national and international projects including (in chronological order) RPCNET, the first Italian packet switching network, STELLA, the first European broadcasting satellite packet network, and OSIRIDE, the first Italian OSI compliant network. At the end of the 1970s, in cooperation with Robert Kahn and Vint Cerf (the inventors of TCP/IP), he led the team that installed the first Internet nodes in Italy (the second country in Europe after the U.K.). The interconnection of the Italian section of Internet with the U.S. and U.K. ones was done via a SATNET gateway. In 1994, he joined the Department of Information Engineering of the University of Pisa as a Full Professor. His current research interests include the design and performance evaluation of MAC protocols for wireless networks, architectures and protocols for mesh networks, and the Quality of Service provision in integrated and differentiated services networks.

Dr. Lenzini is currently on the Editorial Boards of *Computer Networks* and the *Journal of Communications and Networks* as area editor for wireless networks. He served as general chairman for the 1992 IEEE Workshop on Metropolitan Area Networks, for the 2002 European Wireless Networks, and for the First International Conference on Performance Evaluation Methodologies and Tools (ValueTools) which was held in Pisa in October 2006, of which he is a co-founder. He has served as a guest editor of several journals. He has been involved in several national projects and European projects of the 5th and 6th framework program.