End-to-End Bandwidth Reservation in IEEE 802.16 Mesh Networks

Claudio Cicconetti[1], Alessandro Erta[2], Vanessa Gardellin[3], Luciano Lenzini[1], Enzo Mingozzi[1]
[1] *Dipartimento di Ingegneria dell'Informazione, University of Pisa, Italy, {c.cicconetti|l.lenzini|e.mingozzi}@iet.unipi.it*
[2] *IMT Lucca Institute for Advanced Studies, Lucca, Italy, a.erta@imtlucca.it*
[3] *Student, Faculty of Engineering, University of Pisa, Italy*

Abstract

IEEE 802.16 mesh does not include support to traffic flows with strict Quality of Service requirements. In this paper, we propose an End-to-end Bandwidth Reservation Protocol (EBRP) in the backhaul of a Wireless Mesh Network using IEEE 802.16 mesh. The distinctive feature of EBRP is that it is carried out at the Medium Access Control (MAC) layer. Therefore, EBRP not only makes the resource reservation process extremely rapid, it also allows the available resources to be allocated efficiently by exploiting technology-specific information available at the MAC. We present EBRP as part of a framework which also includes the support for performing distributed Call Admission Control (CAC). Preliminary simulations results obtained with VoIP traffic and nodes arranged in a grid topology are presented to show the effectiveness of EBRP, with ideal CAC computation.

1. Introduction

Due to the recent advances of communication technology and ubiquitous deployment of wireline and wireless infrastructures, users are becoming more passionate about multimedia interactive applications, such as Voice over IP (VoIP) and gaming. However, the latter require the underlying network to meet a specified set of Quality of Service (QoS) requirements so as to satisfy user demands. In wireline networks, capacity over-provisioning is one viable —and often implemented— solution, as this resulted much less expensive than deploying complex QoS-aware network architectures. However, this approach is hardly suitable with technologies based on wireless communications, e.g. GPRS/UMTS and IEEE 802.11, since the available network capacity is much scarcer than in competing wireline technologies, e.g. xDSL. This problem is exacerbated in the case of Wireless Mesh Networks (WMNs), due to the presence of multi-hop traffic flows, which consume an even higher amount of wireless capacity than standard point-to-multipoint wireless networks. Still, the interest of industry and academy in WMNs has been increasing during the last few years, and real-life deployments have spread all over the world [1].

So far, most research efforts have been directed to supporting best-effort access in WMNs based on IEEE 802.11 networks [12]. However, the IEEE published in 2004 an amendment to the IEEE 802.16 standard [9], which adds features to the point-to-multipoint MAC protocol [8] in order to support the *mesh* mode. According to the latter, the MAC protocol is based on Time Division Multiple Access (TDMA) and communications occur on a time frame basis. Each frame consists of a control sub-frame and a data sub-frame. Control sub-frames are partitioned into slots of fixed duration (hereafter, *control slots*), up to a maximum of 16 per frame. Likewise, data sub-frames consist of a fixed number of mini-slots, up to 256. The control slots are accessed by means of a distributed procedure, i.e. the *mesh election procedure*, which provides nodes with a collision-free access to the control sub-frame. Further details and performance evaluation of the mesh election procedure can be found in [5]. The control slots are used by the nodes to transmit control messages which can be employed for several purposes. For instance, in [7], the authors exploited the standard three-way handshake request/grant/confirm message exchange to develop a scheduling strategy for best effort data aimed at providing a fair sharing of the network resources among the traffic flows. In this paper, we exploit the control messages to solve the problem of providing traffic flows with strict real-time requirements. Specifically, we propose a framework for end-to-end bandwidth reservation called End-to-end Bandwidth Reservation Protocol (EBRP), which is integrated with the IEEE 802.16 MAC through the addition of some information elements to the MAC control messages. Our framework includes a call admission control procedure which requires all the nodes to have complete knowledge of the network topology.

The latter can be achieved by extending existing routing protocols.

Basically, EBRP consists of two phases. First, a PATH message is forwarded from the node that requests the admission of a new traffic flow to its destination. During this phase resources are not reserved. In the second phase, a RESV message proceeds backwards, hop-by-hop, from the destination node to the node that originated the request along the path laid down by the corresponding PATH message.

Our solution provides evident benefits. First, the time required to open a flow is expected to be decreased with respect to any other protocol running over IP, such as RSVP, since reservation messages are encapsulated into MAC control messages. Furthermore, optimized and efficient fine-grained resource allocation strategies can be envisaged exploiting information present at the MAC layer. A similar approach has been already proposed in the literature [6], though in the context of IEEE 802.11 mesh networks, which however cannot be straightforwardly applied to IEEE 802.16 mesh networks.

2. End-to-end Bandwidth Reservation Protocol

In this section we present the main contribution of this work. In sub-section 2.1 we describe the protocol for end-to-end bandwidth reservation at the MAC layer of an IEEE 802.16 mesh network, namely EBRP, while the procedure for admission control at each node is reported separately in sub-section 2.2.

2.1. EBRP messages and procedures

The End-to-end Bandwidth Reservation Protocol draws its basic concepts from standard protocols for resource reservation in IP networks, such as RSVP [3], from which we broadly reused the lexicon. However, EBRP works at the MAC layer of an IEEE 802.16 network, which makes it more efficient than an IP-based reservation protocol due to decreased overhead, both in terms of bandwidth consumed and time spent during the reservation. More specifically, the messages that are used to establish traffic flows are conveyed as Information Elements (IEs) of Mesh Distributed Scheduling (MSH-DSCH) MAC control messages, which are advertised periodically by all nodes.

We assume that the procedure for bandwidth reservation is triggered by an admission control request from upper layers, which also conveys the traffic flow's specifications, or TSPEC. Since this work is dealing with CBR type of traffic, TSPEC reduces to the nominal packet size (in bytes) and period (in seconds).

First, the node that originates the traffic flow (hereafter *originator*) conveys the admission request to the final destination node (hereafter *recipient*) through a PATH message, which is relayed by the intermediate nodes along the path. In addition to the traffic flow specifications, a PATH message includes the list of: i) the originator and recipient node IDs; ii) the intermediate node IDs[1] (i.e. the traversed nodes); iii) the TSPEC; and iv) the traffic flow identifier (TFID). The originator is required to select the TFID of any new flow in such a way that the triple ⟨originator, recipient, TFID⟩ uniquely identifies the traffic flow in the network. A comprehensive example illustrating the reservation procedure along a four hops path is reported in Fig. 1.

In the second phase, the recipient sends back to the originator a RESV message. The latter is routed through the same path that has been covered by the PATH message. This is achieved by using the list of intermediate node IDs included in the PATH message. On reception of the RESV message, each node performs the CAC to check that enough resources are available in the link towards the recipient. The CAC procedure and exact definition of "resource allocation" are detailed in the next section. Note that, in general, the system performance, in terms of (e.g.) the maximum number of flows that can be supported by the network, depends on the route, i.e. the sequence of nodes, selected from the originator to the recipient. However, in this work we do not investigate this issue, which is left as a future work.

Resources remain allocated until the traffic flow is dropped, which happens if nodes do not receive packets belonging to that flow for a specified amount of time, called T_F, which is a system parameter. Note that some applications may generate packets with an ON/OFF pattern. This is especially the case of interactive multimedia applications, such as Voice over IP, videoconference streaming, and gaming. Therefore, in order to prevent premature termination of the traffic flow during an OFF period, the originator generates *dummy* packets to guarantee a constant refresh of the established traffic flow state on each node in the path. Dummy packets are special messages, which only contain the information about the traffic flow to which they belong, and are silently discarded by recipient's MAC layer. The generation interval of dummy packets is a system parameter, which must be smaller than T_F.

[1] In an IEEE 802.16 mesh network each node is uniquely identified by means of a 16-bit *node ID*.

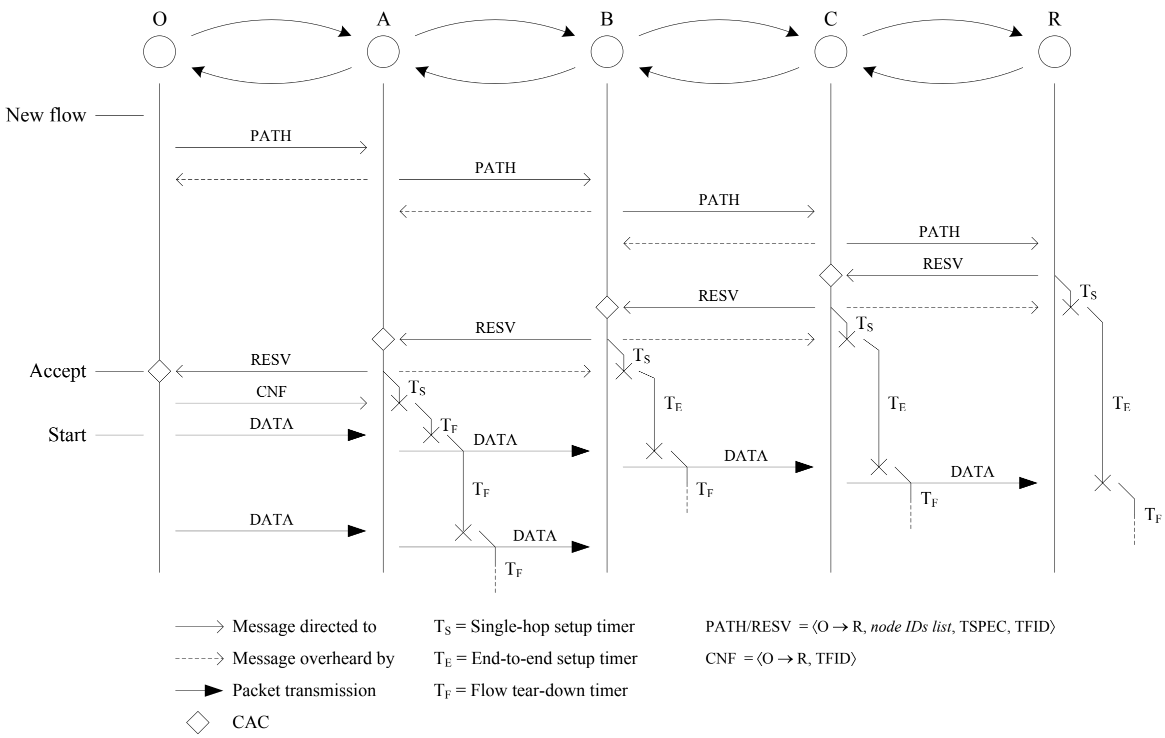

Figure 1. Four hops end-to-end bandwidth reservation example.

Note that sending dummy packets only consumes resources that have been already reserved for the traffic flow in the data sub-frame. Thus, we adopted this solution instead of an RSVP-like approach, where periodic refresh messages, e.g. PATH, are sent to maintain the state along the reserved path. This way, control slots are only used for the purpose of opening new traffic flows.

While the T_F timer is employed by nodes to detect call drops due to user session termination, two additional timers are used by nodes during the setup phase of the traffic flow to detect traffic flow termination due to CAC failure. Both these timers need be configured appropriately to detect a possible CAC failure as early as possible, thus freeing the allocated resources promptly. It is worth noting that, due to the broadcast nature of the control messages, there is no need for explicitly notifying the CAC success/failure to the node that sent the RESV. In fact, CAC status can be inferred by checking whether the RESV message is forwarded further or not (dashed lines in Fig. 1). Clearly, the only exception to this is represented by the originator which must explicitly notify its neighbor of the CAC success through a confirmation (CNF) message.

2.2. Call Admission Control

We now describe the CAC procedure employed by a node to check whether there are sufficient resources to admit a traffic flow over one link. The main idea is to partition the data sub-frame into a number of time intervals of equal duration, each consisting of a *group* of links. In IEEE 802.16, the term *link* has a very precise meaning. In fact, a (logical) link is set up between two nodes by means of a link establishment procedure, provided that they can communicate directly with each other, i.e. they are in the transmission range of one another. In this paper, we will use this definition of link. Each group is built so that all the links that belong to it can be activated simultaneously, i.e. simultaneous transmissions on the links of the same group do not interfere with each other and therefore can be successfully decoded by the receivers. Therefore, any node of the network can allocate resources for one of its links in a distributed manner. For instance, consider the example of groups depicted in Fig. 2. Node A and node D can both independently reserve slots for the transmission of packets towards B and C, respectively, since links AB and DC belong to the same group. This approach has evident benefits, since it allows nodes to act in a distributed manner when performing CAC.

However, the following two issues should be con-

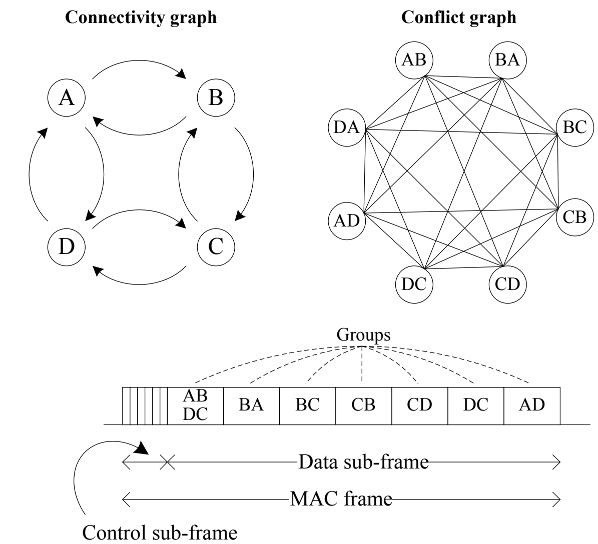

Figure 2. Connectivity/conflict graphs of a sample four-node networks.

sidered. First, it is pretty obvious that all nodes must use the same algorithm to compute the groups over the same input data (i.e. the connectivity graph), so as to obtain a consistent group allocation, since, in general, many different group partitioning can be envisaged from the same set of links. In any case, this requires network-wide information to be distributed among all the nodes. This activity might be performed by any routing protocol properly enhanced[2]. Note that the IEEE 802.16 standard does not specify any mandatory protocol to be used for routing. The second, most critical, issue is how to state that any two links interfere with one another when activated simultaneously. This problem is regarded as one of the most important open issues of WMNs, and is still under investigation, while some preliminary works have been published in the literature [4]. Since the focus of this work is the bandwidth reservation protocol, we solved this issue in a simplistic manner, as described in the following.

We assume that a node is not able to decode transmissions received on a channel from a neighbor node if either (i) the receiving node is transmitting on the same slot, or (ii) another neighbor of the receiving node is transmitting on the same slot. In other words, the total interference caused by the nodes lying outside the one-hop neighborhood of the receiving node is assumed to be negligible. Under this assumption, each node only needs to know the set of links to any of its neighbors. Based on that, it is straightforward to derive in a distributed manner the *connectivity graph* of the network and its relative *conflict graph*, which are widely used means to represent the network topology and to characterize the link interference, respectively [13]. In the former, vertices represent nodes of the network and edges links. In the latter, vertices represent links and edges mutual interference.

After the conflict graph has been computed, each node must use the same algorithm to partition the links into groups. Note that having as few groups as possible is a desirable condition, since this leads to having more slots for each link, i.e. a higher capacity available to accommodate a greater number of traffic flows. Unfortunately, finding the minimum number of groups requires finding the maximum independent set of the conflict graph, which is known to be NP-hard [13]. It is worth noting that, in practice, the interference among links can change over time, due to the time-varying properties of the wireless channel. Whenever this happens, the links need be re-partitioned into a new set of groups, which makes it impractical to use exhaustive search algorithms in most WMN deployments. However, several heuristics are present in the literature of graph theory [2] and WMNs [14]. Furthermore, since nodes in a WMN are usually assumed to be fixed, the link quality and the interference conditions can be considered stable for reasonably long periods of time. This considerably reduces the chance of re-computing the set of groups. The selection of the "best" heuristic algorithm, e.g. in terms of the time required to find a fractional sub-optimum solution, and, in general, the adoption of more sophisticated techniques to characterize the interference and to calculate the groups are not considered in this study and left for future investigation.

3. Performance evaluation

We first define the settings under which the performance evaluation is carried out in Section 3.1, which also reports the metrics used to assess the performance. Then, we present a set of simulation results in Section 3.2.

3.1. Simulation environment

The parameters used in simulations are those specified in the IEEE 802.16 standard [9]. The frame duration has been set to 4 ms, and the modulation used for data transmission was QPSK, with code rate 3/4. We have implemented the IEEE 802.16 mesh in the *ns2* network simulator [11], including the standard distrib-

[2] Note that, in principle, any routing protocol, including cross-layer solutions, which provides all the nodes with a complete knowledge of the network topology, can be extended so as to support EBPR.

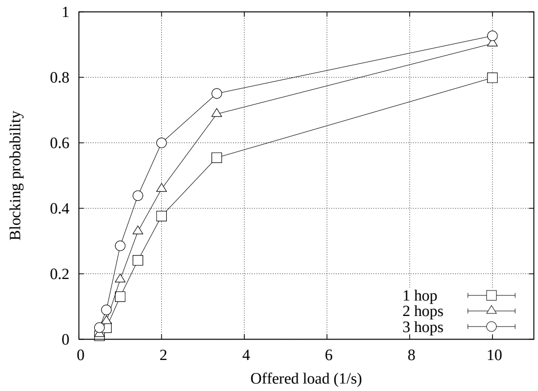

Figure 3. Blocking probability versus offered load with 8 control slots per frame.

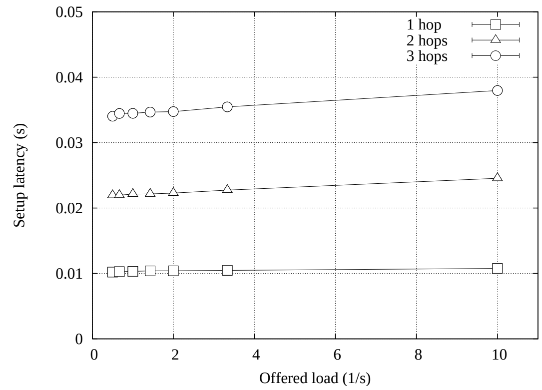

Figure 4. Setup latency versus offered load with 8 control slots per frame.

uted election procedure to access control slots. The WMN is assumed to operate in a steady state, where links are not established or removed over time, and transmission rates are stable.

The following performance indices have been considered for the purpose of assessing the end-to-end bandwidth reservation framework performance:

- The *blocking probability* is computed as the ratio between the number of traffic flows that are refused by the CAC over the number of total traffic flows that have requested to be established.
- The *setup latency* is the interval between the time when the request to establish a new flow (issued by an application) is received at the originator and the time when the traffic flow reservation message is received by the originator.

The simulation scenario consists of a grid of 5×5 nodes. We distinguish between two types of nodes: gateways and access points. A gateway is the egress point of the WMN, while an access point serves an area with a number of mesh clients. Traffic flows are only established between an access point and its nearest gateway. The four vertices of the grid are gateways, while the remaining nodes are access points. We assume that all the access points have the same workload, which consists of G.711 VoIP calls [10] of constant duration equal to 2 s, requesting admission at exponentially distributed intervals, with average *T*.

3.2. Simulation results

We first assume that there are eight control slots per frame, which entails a moderate amount of MAC overhead due to transmission of control messages. In fact, about one third of the channel capacity is consumed by the control sub-frame.

Let us consider the blocking probability, which is reported in Fig. 3, when the offered load increases. The latter is expressed as $1/T$ where T is the average interval between two consecutive calls at the same node and increases from 0.1 s to 2 s. Three different curves are reported, depending on the distance of the access point from its nearest gateway. As expected, the blocking probability increases when the offered load increases. However, what is most interesting is that the blocking probability is not affected by the path length significantly. This is because the probability that there are not enough resources to accept a call decreases with the distance from the recipient, since links that are farther from the gateway tend to be less loaded due to spatial reuse, which is captured by the CAC procedure described in Sec. 2.2.

We now evaluate the setup latency, which is shown in Fig. 4. Since in this case the control part of the MAC frame is overprovisioned, the setup latency is only affected by the propagation delay of the PATH and RESV messages. Therefore, unlike the blocking probability, this metric is greatly affected by the path length. Nonetheless, the absolute value of the setup latency is very small for all distances of the access points from the gateway, due to the forwarding of EBRP messages at the MAC layer. When the number of control slots increases, however, the setup latency decreases significantly. This is shown in Fig. 5, which reports the setup latency when the number of control slots increases from 2 to 16, which yields a decreasing MAC overhead from ~8% to ~66%. However, even though the minimum overhead is achieved, the setup latency is still in the order of hundreds of ms in all cases, which makes EBRP a suitable candidate for realistic applications, e.g. VoIP calls.

To conclude, in Fig. 6, we report the blocking probability versus the number of control slots per frame. As can be seen, having a smaller number of control slots per frame entails a lower blocking prob-

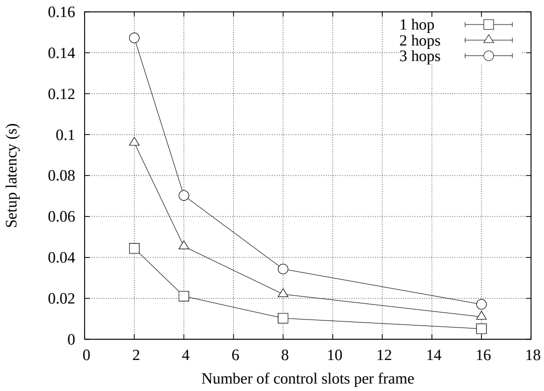

Figure 5. Setup latency versus number of control slots per frame.

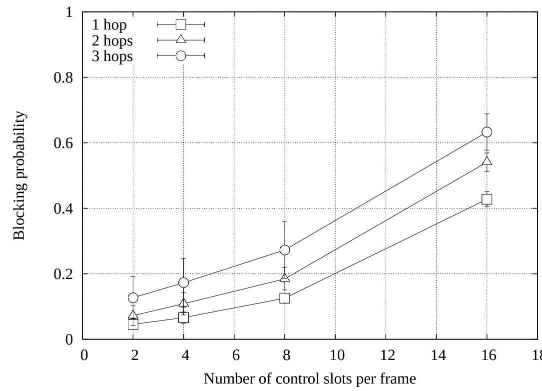

Figure 6. Blocking probability versus number of control slots per frame.

ability. In fact, the smaller the number control slots per frame, the higher the capacity available for data transmission, hence, the greater the number of flows which can be admitted on average. Note the trade-off between the setup latency and the blocking probability reported in Fig. 5 and Fig. 6, respectively. Reducing the number of control slots per frame provides a higher available capacity for data transmission, but, at the same time, increases the setup latency.

4. Conclusions

In this paper we have proposed an end-to-end bandwidth reservation scheme, called EBRP, to be employed at the MAC layer of an IEEE 802.16 mesh network. Specifically, we have devised a resource reservation protocol, inspired from RSVP, and a procedure to perform distributed CAC of traffic flows. We have assessed the effectiveness of EBRP through simulation in a reference scenario consisting of a 5×5 grid topology with VoIP traffic. Results have shown that the latency to establish traffic flows is relatively small and significantly depends on the distance of the originator node from the gateway. On the other hand, the spatial bias does not affect significantly the probability that a traffic flow is rejected, due to spatial reuse.

5. References

[1] I. F. Akyildiz, X. Wang, "A survey on wireless mesh networks," *IEEE Communications*, vol. 43, no. 9, Sept. 2005, pp. 23–30.

[2] B. S. Baker. "Approximation algorithms for NP-complete problems on planar graphs," *Journal of the ACM*, vol. 41, no. 1, Jan. 1994, pp. 153–180.

[3] R. Braden, L. Zhang, S. Berson, S. Jamin. "RFC 2205, Resource Reservation Protocol (RSVP)," Sep. 1997.

[4] G. Brar, D. M. Blough, P. Santi. "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," proc. *ACM MobiCom 2006*, Los Angeles, USA, Sep. 23–26, pp. 2–13.

[5] M. Cao, W. Ma, Q. Zhang, X. Wang, W. Zhu. "Modelling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode," proc. *ACM MobiHoc 2005*, Urbana-Champaign, USA, May 25–28, 2005, pp. 78–89.

[6] E. Carlson, C. Prehofer, C. Bettstetter, H. Karl, A. Wolisz. "A distributed end-to-end reservation protocol for IEEE 802.11-based Wireless Mesh Networks," *IEEE JSAC*, vol. 24, no. 11, Nov. 2006, pp. 2018–2027.

[7] C. Cicconetti, I. F. Akyildiz, L. Lenzini. "Bandwidth balancing in multi-channel IEEE 802.16 wireless mesh networks," proc. *IEEE INFOCOM 2007*, Anchorage, Alaska, USA, May 6–12, 2007.

[8] C. Cicconetti, C. Eklund, L. Lenzini, E. Mingozzi. "Quality of service support in IEEE 802.16 networks," *IEEE Network*, vol. 20, no. 2, March 2006, pp. 50–55.

[9] IEEE 802.16-2004. "IEEE standard for Local and Metropolitan Area Networks – Part 16," October 1, 2004.

[10] ITU-T. "Recommendation G.711: Pulse code modulation (PCM) of voice frequencies," Nov. 1988.

[11] http://www.isi.edu/nsnam/ns/, last version 2.29, October 2005.

[12] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, J. Zahorjan. "Measurement-based models of delivery and interference in static wireless networks," proc. *ACM SIGCOMM 2006*, Pisa, Italy, Sep. 11–15, pp. 51–62.

[13] G. Sharma, R. R. Mazumdar, N. B. Shroff. "On the complexity of scheduling in wireless networks," proc. *ACM MobiCom 2006*, Los Angeles, USA, Sep. 23–26, pp. 227–238.

[14] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, O. Frieder. "Efficient interference-aware TDMA link scheduling for static wireless networks," proc. *ACM MobiCom 2006*, Los Angeles, USA, Sep. 23–26, pp. 262–273.