Performance Evaluation of the IEEE 802.16 MAC for QoS Support

Claudio Cicconetti, Alessandro Erta, Luciano Lenzini, and Enzo Mingozzi

Abstract—The IEEE 802.16 is a standard for broadband wireless communication in Metropolitan Area Networks (MAN). To meet the QoS requirements of multimedia applications, the IEEE 802.16 standard provides four different scheduling services: Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), non-real-time Polling Service (nrtPS), and Best Effort (BE). The paper is aimed at verifying, via simulation, the effectiveness of rtPS, nrtPS, and BE (but UGS) in managing traffic generated by data and multimedia sources. Performance is assessed for an IEEE 802.16 wireless system working in Point-to-Multipoint (PMP) mode, with Frequency Division Duplex (FDD), and with full-duplex Subscriber Stations (SSs). Our results show that the performance of the system, in terms of throughput and delay, depends on several factors. These include the frame duration, the mechanisms for requesting uplink bandwidth, and the offered load partitioning, i.e., the way traffic is distributed among SSs, connections within each SS, and traffic sources within each connection. The results also highlight that the rtPS scheduling service is a very robust scheduling service for meeting the delay requirements of multimedia applications.

Index Terms—IEEE 802.16, broadband wireless access, MAC protocols, quality of service, scheduling algorithms, performance evaluation.

1 Introduction

During the last few years, commercial and residential users have witnessed a rapid growth of new services based on multimedia applications, such as Voice over IP (VoIP), video conferencing, Video on Demand (VoD), massive online gaming, and peer-to-peer. The most important driving factor behind this dramatic rise is the increasing availability of broadband access, based on leased lines using fiber optic links, cable modems, and digital subscriber line (xDSL) access networks. At the same time, users have become familiar with personal devices, such as laptops, palmtops, and cellular phones, and are thus reliant on ubiquitous service. Industry and research communities are consequently investing considerable effort in the convergence of multimedia services and ubiquitous instant access, which by necessity depends on the use of Broadband Wireless Access (BWA) technologies [1]. Standards for BWA are being developed within IEEE project 802, Working Group 16, often referred to as 802.16. The IEEE 802.16 standard is also known in the trade press as Worldwide Interoperability for Microwave Access (WiMAX). The current version of the standard was published in 2004 [12], though the standardization process is still ongoing [13].

The 802.16 standard specifies two modes for sharing the wireless medium: Point-to-Multipoint (PMP) and Mesh (optional). In the PMP mode, the nodes are organized into a cellular-like structure, where a base station (BS) serves a set of subscriber stations (SSs) within the same antenna sector in a broadcast manner, with all SSs receiving the same transmission from the BS. Transmissions from SSs are directed to and coordinated by the BS. On the other hand, in Mesh mode, the nodes are organized ad hoc and scheduling is distributed among them.

In this paper, we focus on the PMP mode. In the IEEE 802.16 standard, uplink (from SS to BS) and downlink (from BS to SS) data transmissions are frame-based, i.e., time is partitioned into subframes of fixed duration. Since the transmission is broadcast, all SSs listen to the data transmitted by the BS in the downlink subframe. However, an SS is only required to process data that are directed to itself or that are explicitly intended for all the SSs. In the uplink subframe, on the other hand, the SSs transmit data to the BS in a Time Division Multiple Access (TDMA) manner. Downlink and uplink subframes are duplexed using one of the following techniques: Frequency Division Duplex (FDD), where downlink and uplink subframes occur simultaneously on separate frequencies, and Time Division Duplex (TDD), where downlink and uplink subframes occur at different times (i.e., they alternate to each other) and usually share the same frequency. SSs can be either full-duplex, i.e., they can transmit and receive simultaneously, or half-duplex, i.e., they can transmit and receive at nonoverlapping time intervals.

This paper is aimed at verifying, via simulation, the ability of IEEE 802.16 MAC to manage traffic generated by multimedia applications, with strict QoS requirements, and by data applications, which do not pose such constraints. Conclusions are drawn for an IEEE 802.16 wireless system working in Point-to-Multipoint (PMP) mode, with Frequency Division Duplex (FDD), and with full-duplex Subscriber Stations (SSs). The target air interface is WirelessMAN-OFDM, based on Orthogonal Frequency Division Multiplexing (OFDM).

- *C. Cicconetti, L. Lenzini, and E. Mingozzi are with the Dipartimento Ingegneria dell'Informazione, via Diotisalvi 2, I—56122, Pisa, Italy. E-mail: {c.cicconetti, l.lenzini, e.mingozzi}@iet.unipi.it.*
- *A. Erta is with the IMT Lucca Institute for Advanced Studies, via S. Micheletto 3, 55100, Lucca, Italy. E-mail: alessandro.erta@imtlucca.it.*

Manuscript received 25 Nov. 2005; revised 16 Mar. 2006; accepted 4 Apr. 2006; published online 15 Nov. 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0347-1105.

To the best of our knowledge, this type of analysis has not yet been made for the 802.16 technology. More specifically, previous performance evaluation work on IEEE 802.16 focuses on specific aspects. In [5], we described the QoS framework of 802.16 and discussed simulation results in specific application scenarios, i.e., last mile Internet access for residential and small and medium-sized enterprises users. A packet scheduling algorithm with QoS support has been proposed in [26]. The mechanisms for supporting the Automatic Repeat reQuest (ARQ) optional feature of 802.16, which provides error recovery at the MAC layer, have been evaluated in [9]. An efficient algorithm for scheduling uplink grants to SSs with VoIP traffic has been proposed in [17]. In [11], the author performed a hybrid analytic-simulative analysis of the effect on the system performance of several MAC mechanisms, including the fragmentation of Service Data Units (SDUs) and the padding of OFDM symbols. The performance with TDD mode has been analyzed in [4], [10]. Finally, in [8], the authors analyzed the performance of WiMAX systems from the perspective of a physical layer.

The paper is organized as follows: In Section 2, we describe the 802.16 standard, focusing on the MAC layer. We describe in detail the implementation choices in our instance of the 802.16 standard in Section 3. We also characterize the workload and denote the measures of interest. An extensive performance evaluation is assessed in Section 4 both for data and multimedia traffic. Finally, conclusions are drawn in Section 5.

2 IEEE 802.16

In this section, we briefly introduce the IEEE 802.16 MAC, focusing on those features that are specifically relevant to this paper—see [6] for more details.

The MAC protocol is connection-oriented: All data communications, for both transport and control, are in the context of a unidirectional connection. SSs medium access is coordinated by the BS. At the beginning of each downlink subframe, the BS broadcasts the uplink and downlink MAP messages, UL-MAP and DL-MAP, respectively. These maps notify the SSs of the start and the end times of their uplink/downlink grants. The uplink subframe is delayed with respect to the downlink subframe by a fixed amount of time, called the *uplink allocation start time*, so as to give SSs enough time to decode the UL-MAP and take appropriate decisions. At the beginning of the downlink subframe, the BS transmits a sequence of physical preambles to let the SSs regain synchronization after the uplink subframe. A physical preamble consists of one OFDM symbol[1] and carries a well-known bit sequence and synchronization information. In order to synchronize the BS's receiver, each 802.16 SS transmits a physical preamble in the uplink direction before transmitting data. The 802.16 MAC layer encapsulates the Service Data Units (SDUs) generated by applications in Protocol Data Units (PDUs). If needed, the MAC layer can fragment an SDU into multiple variable length PDUs. Each MAC PDU begins with a 6 byte fixed-length MAC header.

SSs notify the BS of the amount of bytes (i.e., the backlog) to be sent by a connection through specific MAC headers. While bandwidth is requested by an SS per each connection, the BS grants uplink bandwidth to an SS as a whole. Due to this hybrid nature of the request/grant mechanism (i.e., requests per connection, grants per SS), an SS also has to implement locally a scheduling algorithm to redistribute the granted capacity to all of its connections. The bandwidth request can be *incremental* or *aggregate*. If it is aggregate, the SS indicates the whole connection backlog. Whereas, if it is incremental, the SS indicates the difference between its current backlog and the one carried by its last bandwidth request. There are several bandwidth request mechanisms: *unsolicited requests, unicast polls, broadcast/multicast polls,* and *piggybacking*.

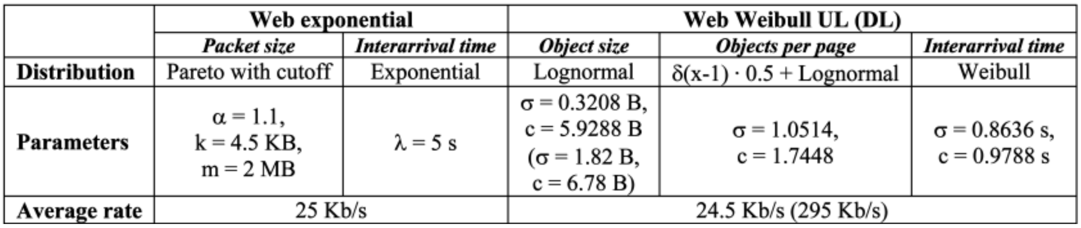
Since it would not be feasible to address the QoS requirements of all of the applications foreseen for an IEEE 802.16 network, their functionality are grouped by the standard into a small number of classes named *scheduling services* based on the commonality of their: 1) QoS service requirements (e.g., real-time applications with stringent delay requirements, best effort applications with minimum guaranteed bandwidth), 2) packet arrival pattern (fixed/variable-size data packets at periodic/aperiodic intervals), and 3) mechanisms to send bandwidth requests to the BS. Thus, each scheduling service is tailored to support a specific class of applications. In the following, we describe the IEEE 802.16 scheduling services by focusing on the supported targeted applications and related bandwidth request mechanisms (uplink only).

Unsolicited Grant Service (UGS) is designed to support real-time applications, with strict delay requirements, which generate fixed-size data packets at periodic intervals, such as T1/E1. Therefore, UGS is defined so as to closely follow the packet arrival pattern. Grants occur on a periodic basis. The base period and the grant size are specified during the connection setup phase. After that, SSs never request bandwidth for UGS connections. For these reasons, we did not find this scheduling service interesting from a MAC standpoint, and so its performance is not assessed in this paper.

Real-time Polling Service (rtPS) is designed to support real-time applications with less stringent delay requirements, which generate variable-size data packets at periodic intervals, such as Moving Pictures Expert Group (MPEG) video and VoIP with silence suppression. Unlike UGS-tailored applications, the size of arriving packets with rtPS is not fixed, thus SSs are required to explicitly make a request for bandwidth from the BS. The standard provides that the BS periodically sends unicast polls to rtPS connections. The base period can be specified during the connection setup. Specifically, it is possible to set the polling period to the interval at which packets are expected to be generated by the application. A unicast poll consists of an uplink allocation from the BS to the polled SS of the bandwidth needed to transmit a bandwidth request PDU.

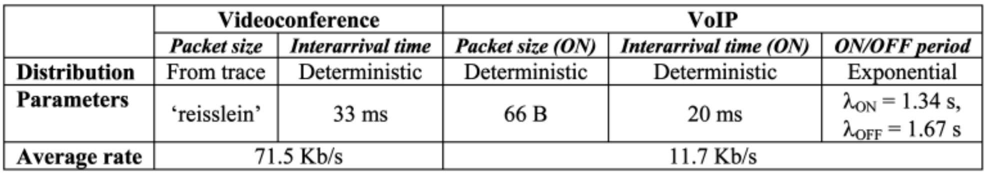
[1] An OFDM symbol is made up from subcarriers, the number of which determines the Fast Fourier Transform (FFT) size used. The standard specifies an FFT size of 256. Part of the OFDM symbol duration, named the Cyclic Prefix duration, is used to collect multipath. The interested reader can find a technical introduction to the OFDM system of the IEEE 802.16 in [15].

TABLE 1
Workload Characterization (Web Sources)



	Web exponential		**Web Weibull UL (DL)**		
	Packet size	***Interarrival time***	***Object size***	***Objects per page***	***Interarrival time***
Distribution	Pareto with cutoff	Exponential	Lognormal	$\delta(x-1) \cdot 0.5$ + Lognormal	Weibull
Parameters	$\alpha = 1.1$, $k = 4.5$ KB, $m = 2$ MB	$\lambda = 5$ s	$\sigma = 0.3208$ B, $c = 5.9288$ B ($\sigma = 1.82$ B, $c = 6.78$ B)	$\sigma = 1.0514$, $c = 1.7448$	$\sigma = 0.8636$ s, $c = 0.9788$ s
Average rate	25 Kb/s		24.5 Kb/s (295 Kb/s)		

TABLE 2
Workload Characterization (Multimedia Sources)



	Videoconference		**VoIP**		
	Packet size	***Interarrival time***	***Packet size (ON)***	***Interarrival time (ON)***	***ON/OFF period***
Distribution	From trace	Deterministic	Deterministic	Deterministic	Exponential
Parameters	'reisslein'	33 ms	66 B	20 ms	$\lambda_{ON} = 1.34$ s, $\lambda_{OFF} = 1.67$ s
Average rate	71.5 Kb/s		11.7 Kb/s		

Unlike UGS and rtPS scheduling services, *non-real-time Polling Service* (nrtPS) and *Best Effort* (BE) are designed for applications that do not have specific delay requirements. The main difference between them is that nrtPS connections are reserved a minimum amount of bandwidth (by means of the Minimum Reserved Traffic Rate parameter). Additionally, the BS grants unicast polls to nrtPS connections on a large time-scale. The IEEE 802.16 standard specifies this scale to be one second or less. Both nrtPS and BE uplink connections typically use contention-based bandwidth requests. Such requests are sent in response to broadcast/multicast polls, which are advertised by the BS in the UL-MAP. The BS is free to use any algorithm to decide which uplink subframe portion is reserved for broadcast/multicast contention slots on a frame-by-frame basis. The main drawback of this mechanism is that a collision occurs whenever two or more SSs access the medium in the same contention slot to send a bandwidth request. A bandwidth request is considered lost (i.e., a collision occurred) if the transmitting SS does not receive the related data grant within a specified timeout (50 ms, in our analysis). To reduce the likelihood of this event, a collision avoidance scheme is used. SSs randomly select a number in the backoff window (see [12]) which indicates the number of contention slots the SSs must defer before transmitting. When collisions occur, a truncated binary exponential backoff algorithm is employed to increase the backoff window. Consequently, this polling mechanism is tailored to serve traffic with no specific delay requirements, such as bursty Web traffic.

In addition, an SS can issue an unsolicited bandwidth request for one of its non-UGS backlogged connections by consuming part of the grant that it was allocated for the transmission of data. Optionally, incremental unsolicited bandwidth requests can be piggybacked to PDUs by means of a specific 2 bytes MAC subheader. In Section 3.3.2, we describe the procedures employed by the BS and SSs for bandwidth request/granting, which have been left unspecified by the 802.16 standard.

3 Simulation Environment

In this section, we explain the simulation environment in detail. First, we characterize the traffic workload and define the performance metrics of interest, and then we describe the design choices which are deliberately not specified in the standard and are thus left up to each manufacturer. Specifically, we justify our choice of the scheduling algorithms running on the BS and SSs, and then we continue by showing the way we manage bandwidth requests mechanisms. The simulations were carried out by means of a prototypical simulator of the IEEE 802.16 MAC protocol. The simulator is event-driven and was developed using C++. The MAC layer of SSs and the BS are implemented, including all procedures and functions for uplink/downlink data transmission and uplink bandwidth request/grant.

3.1 Traffic Models

Different types of traffic sources are used in the simulation scenarios. The data traffic is modeled as a Web source. We used two different Web source models, namely, Web *exponential* [20] and Web *Weibull* [18]. Table 1 shows the characterization of the two traffic models. Multimedia traffic is evaluated by means of Videoconference and VoIP sources. Their characterizations are reported in Table 2. Specifically, VoIP is modeled as an ON/OFF source with Voice Activity Detection (VAD). Packets are generated only during the ON period. The duration of the ON and OFF periods is distributed exponentially [2]. On the other hand, videoconference traffic is based on a preencoded MPEG4 trace from a real-life lecture [7].

3.2 Performance Metrics

We have specified several metrics to assess the performance of the 802.16 MAC protocol. The following traffic-related metrics have been defined:

1. *gross subframe utilization* (hereafter, *utilization*), the ratio between the OFDM symbols utilized in a subframe for data transmission (including physical

preambles) over the total number of OFDM symbols contained in a subframe;

2. *throughput*, the overall amount of net user data (i.e., data purged from the MAC header and trailer overheads other than the physical preambles and Forward Error Correction (FEC) overhead) carried out by the system in the unit of time;
3. *transfer delay* (hereafter, *delay*), the time interval between when a packet arrives at the MAC connection buffer of the source node (SS/BS) and when this packet is completely delivered to the next protocol layer at the destination node (BS/SS); and
4. *number of SSs served per frame*, the number of SSs which receive an uplink grant by the BS.

On the other hand, bandwidth request mechanisms are assessed by means of the following metrics:

5. N_C, number of contention-based bandwidth requests received by the BS per uplink subframe;
6. N_P, number of piggybacked bandwidth requests received by the BS per uplink subframe;
7. *backlog gap*, difference between the BS's estimate of the backlog of a connection (as acquired via bandwidth requests) and the actual backlog of that connection on the SS; and
8. *notification delay*, the time interval between the time instant at which a new SDU is received by an SS and the time instant at which the BS receives a bandwidth request for this SDU.

3.3 Simulation Choices

In this section, we describe the implementation choices related to our instance of the 802.16 standard used for simulation.

3.3.1 BS and SS Schedulers

At the beginning of each frame, the BS is responsible for broadcasting the uplink and downlink schedules through the UL/DL-MAP messages. UL/DL-MAPs must be produced frame by frame, taking into account the QoS requirements of each connection. However, the 802.16 standard clearly states that the scheduling algorithm running on the BS (as well as that one running on SSs) is left up to the manufacturer. Many scheduling algorithms have been put forward in the literature to support QoS in wired and wireless networks [3]. Since a minimum reserved rate is the basic QoS parameter negotiated by a connection within an 802.16 scheduling service, the class of latency-rate [23] scheduling algorithms is particularly suited for implementing the schedulers in the 802.16 MAC. Specifically, within this class, we selected Deficit Round Robin (DRR) as the downlink scheduler to be implemented at the BS [22], since it combines the ability of providing fair queuing, in the presence of variable length packets, with the simplicity of implementation. In fact, it can exhibit $O(1)$ complexity, provided that specific allocation constraints are met. In particular, DRR requires a minimum rate to be reserved for each connection being scheduled. Therefore, although not required by the 802.16 standard, BE connections should also be guaranteed a minimum rate. This opportunity can be taken either to avoid BE traffic starvation in overloaded scenarios or to let BE traffic take advantage of the excess bandwidth which is not reserved for the other scheduling services. DRR assumes that the size of the head-of-line packet is known at each packet queue, thus it cannot be used by the BS to schedule transmissions in the uplink direction. In fact, with regard to the uplink direction, the BS is only able to estimate the overall amount of backlog of each connection, but not the size of each backlogged packet. Therefore, we selected Weighted Round Robin (WRR) [14] as the uplink scheduler in our 802.16 simulator. Like DRR, WRR belongs to the class of rate-latency scheduling algorithms.

To enforce QoS support at a connection level, connections are not grouped together based on the SS they belong to, but they are served independently with both the DRR and the WRR scheduling algorithms. We provide only one instance of the DRR/WRR scheduler on the BS for all the downlink/uplink connections, irrespective of the connection scheduling service. Before building the downlink and uplink MAPs, the BS groups the grants addressed to the same SS. Since a physical preamble must be prepended to each uplink grant, this reduces the number of uplink grants per subframe and, thus, the overhead.

Any SS, on receiving a grant from the BS, must share it among the backlogged connections according to an established policy. As for the BS, we decided to adopt the DRR scheduler at the SSs. The uplink capacity, which is assigned by the BS on a frame-by-frame basis, is thus shared fairly by the connections of each SS proportionally to their minimum reserved rates.

3.3.2 Bandwidth Requests Management

Even though the 802.16 standard provides the SSs with several mechanisms for requesting bandwidth, the actual procedure defining what mechanism an SS shall use and when, in order to inform the BS of its bandwidth requirements, is not specified. In this section, we describe how we managed to implement such a procedure.

An SS sends a contention-based bandwidth request to the BS for a BE or nrtPS connection when it becomes busy.[2] As soon as the SS receives an uplink grant, that connection becomes eligible for service by the DRR packet scheduler at the SS. It may happen that new SDUs are buffered at a connection while it is busy. In this case, an SS may issue an incremental bandwidth request by means of a specific MAC header or by piggybacking the request to the transmitted connection PDUs. In our implementation, SSs only use piggybacking for busy connections. Moreover, SSs always request bandwidth when needed. We reserve in each uplink subframe a minimum amount of contention slots, namely, BW_{min} for broadcast polls. BW_{min} remains constant during the whole simulation run. However, the uplink subframe capacity that is not scheduled as uplink grants to SSs is made available as broadcast polls. The impact of these choices on the performance, in terms of throughput and delay, is discussed in Section 4.2.

[2] In the following, we define a connection as *busy* (or backlogged) when it has one or more buffered SDUs awaiting transmission. Connections that are not busy are said to be *idle*.

Finally, our implementation supports rtPS connections by means of a static allocation of periodic unicast polls. The polling period of each connection is equal to the SDU interarrival time (i.e., videoconference: 33 ms; VoIP: 20 ms). On the other hand, we provide nrtPS connections with unicast polls every 500 ms.

4 Performance Evaluation

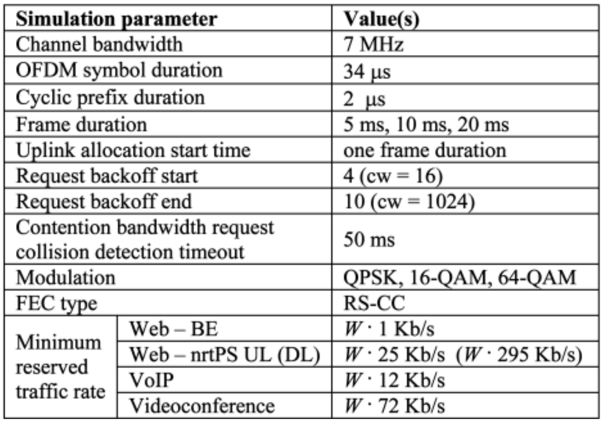
In this section, we report and discuss simulative results of an extensive performance analysis of the IEEE 802.16 operated with the WirelessMAN-OFDM air interface in FDD mode. First, we analyze the 802.16 performance with data traffic only, i.e., with traffic which does not have specific QoS requirements. Specifically, we estimate the throughput and the average delay under several traffic scenarios and system parameters values. Second, to gain insight into the MAC protocol, we investigate and compare the effectiveness of the 802.16 bandwidth request mechanisms. Finally, we assess the performance of 802.16 with several multimedia traffic scenarios, which typically pose stringent delay requirements. The target scheduling service for data traffic is BE, whereas for multimedia traffic, it is rtPS. Moreover, we evaluate the nrtPS scheduling service with data and multimedia traffic.

We assume that all SSs have full-duplex capabilities, thus the whole downlink (uplink) subframe duration can be used by each SS to receive (transmit) as notified by the BS through MAPs. Furthermore, we assume ideal channel conditions, i.e., no packet corruption is due to the wireless channel impairment. This allows us to get insight into the mechanisms that are provided by the 802.16 MAC to manage data and multimedia traffic, regardless of any specific assumptions on the physical characteristics. Finally, we analyze the system while in a steady-state, where the set of admitted connections does not change over time.[3]

System parameters used in the simulation analysis are reported in Table 3. Specifically, the physical layer parameters are those envisaged by the WiMAX forum in [25] and currently employed by manufacturers producing 802.16-compliant devices (e.g., [21]). Furthermore, since 802.16 takes on adaptive modulation and coding to adjust data transmission to different channel conditions,[4] in our simulation scenarios, we consider a mix of SSs employing different modulation schemes. Specifically, based on the results presented in [11], which was derived by assuming that SSs are uniformly distributed in a circular cell, with the BS placed in the center, the number of SSs employing QPSK modulation is assumed to be twice as much as the number of SSs employing 16-QAM modulation, which is in turn twice as much as the number of SSs employing 64-QAM modulation.

As far as workload is concerned, we assume in all scenarios that each connection carries aggregate traffic from a number W of identical basic data sources, whose specific type—Web, Videoconference, VoIP—and characterization, e.g., in terms of rate, depend on the actual simulation scenario. Furthermore, we assume that each SS supports in a given scenario a fixed number C of connections per direction. By denoting with S the overall number of stations, the system offered load (hereafter, offered load) per direction can therefore be expressed as a number $N = S \times C \times W$ of elementary traffic sources.

TABLE 3
Simulation Parameters



Simulation parameter		Value(s)
Channel bandwidth		7 MHz
OFDM symbol duration		34 μs
Cyclic prefix duration		2 μs
Frame duration		5 ms, 10 ms, 20 ms
Uplink allocation start time		one frame duration
Request backoff start		4 (cw = 16)
Request backoff end		10 (cw = 1024)
Contention bandwidth request collision detection timeout		50 ms
Modulation		QPSK, 16-QAM, 64-QAM
FEC type		RS-CC
Minimum reserved traffic rate	Web – BE	$W \cdot 1$ Kb/s
	Web – nrtPS UL (DL)	$W \cdot 25$ Kb/s ($W \cdot 295$ Kb/s)
	VoIP	$W \cdot 12$ Kb/s
	Videoconference	$W \cdot 72$ Kb/s

As mentioned above, scheduling algorithms have been selected so as to provide each connection with a minimum guaranteed rate. Specifically, based on well-known results related to DRR and WRR parameters' configuration [14], [22], scheduling parameters are set so as to reserve a minimum rate to each type of traffic as reported in Table 3. More specifically, the minimum reserved rate of a VoIP connection is computed as the sum of the VoIP sources' peak rates. On the other hand, Videoconference connections are provided with a minimum reserved rate equal to the sum of the sources's average rates. Finally, in regard to Web traffic, the minimum reserved rate depends on the scheduling service employed: nrtPS connections are guaranteed a minimum reserved rate equal to the sum of the sources' average rates; the minimum reserved rate of BE connections, on the other hand, is set to a nominal value of 1 Kb/s for each source.

The simulation analysis was carried out by using the method of independent replications [16]. Specifically, the simulation of each scenario was repeated 20 times. The duration of each run was 1,200 s, with a warmup period of 360 s, during which measures were not collected. In all the simulation runs, we estimated the 95 percent confidence interval[5] of each performance measure.

4.1 Throughput and Delay Analysis

In the following performance evaluation study, we identified a number of key factors that might affect the data traffic performance: the arrival process of traffic sources, scheduling service (uplink only), BWmin value, frame duration, and offered load partitioning. In this scenario, the minimum traffic unit is 147 Kb/s, as derived from the aggregation of six Web sources. Thus, the offered load is $N \times 147$ Kb/s.

To evaluate how the arrival process of traffic sources affects the performance, we ran all simulation scenarios

[3] We do not assess the performance of the signaling protocol between the BS and SSs for establishing new connections and the admission control procedures at the BS.

[4] Channel conditions may depend on a number of factors, including path loss, shadow and multipath fading, and interference from nearby SSs.

[5] The confidence interval is not drawn whenever it is negligible.

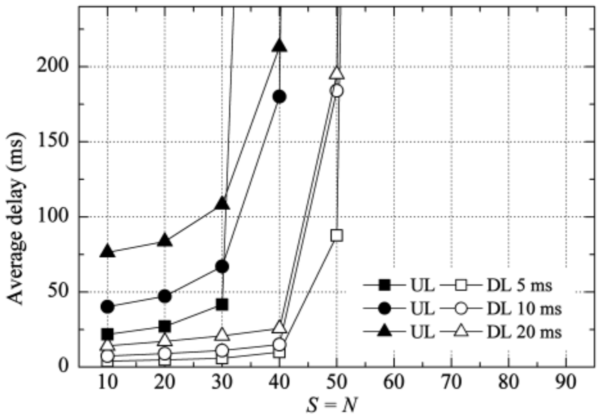


Fig. 1. Average delay versus number of SSs.

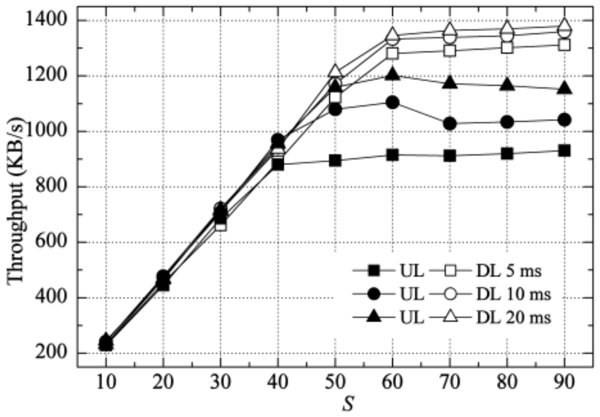


Fig. 2. Throughput versus number of SSs.

with both *exponential* Web sources and *Weibull* Web sources, as defined in Section 3.1. The simulation results in the two cases do not differ (in a statistical sense) from each other. Thus, in this section, we report only the results obtained with the *exponential* Web source. Furthermore, for uplink connections, we found that the nrtPS and BE scheduling services perform substantially the same. We discuss this counterintuitive result at the end of Section 4.2. In the remainder of this section, we assume that uplink connections are served with the BE scheduling service. Finally, in this section, we set BW_{min} to 7. The rationale behind this choice is given in Section 4.2.

We start the analysis by setting up a scenario with an increasing number of SSs (S). Each SS has one connection, which is loaded with one traffic source, i.e., $C = 1$, $W = 1$, and $N = S$. Fig. 1 shows the average delay of downlink and uplink connections versus the number of SSs for three different frame durations (i.e., 5ms, 10ms, and 20ms). As expected, the average delay increases with the offered load. In fact, the time needed for the BS scheduler to "serve" a downlink (uplink) busy connection depends on the overall (estimated) amount of backlogged data from the various SSs' connections. Furthermore, the average delay of uplink connections is much higher than that of downlink connections. This is because the transmission of uplink SDUs requires the SSs to request bandwidth from the BS, thus incurring an additional delay. When the system is lightly loaded, the average delay increases with the frame duration, in both the uplink and the downlink directions. In such conditions, since the connections buffers are empty most of the time, the main component of the delay is the time between the packet arrival and the beginning of the forthcoming frame. In particular, note that the gap between any two downlink curves is almost equal to the difference between the respective frame durations. Moreover, in the uplink case, the shorter the frame duration, the sooner the system gets overloaded. In fact, when the system is underloaded, the average number of SSs served per frame is the same in all three cases (not shown). Consequently, the shorter the frame duration, the higher the overhead due to physical preambles and, thus, the lower the available bandwidth for data traffic. On the other hand, as soon as the system gets overloaded, there is a sharp increase in the average delay, whose main component becomes the queuing delay of SDUs at their connection buffers given that the connections buffers are full most of the time.

Furthermore, as shown in Fig. 2, shorter frames have a drawback from a throughput standpoint. As can be seen, the throughput increases linearly with the offered load as long as the system is underloaded ($N \leq 40$ in the uplink case, $N \leq 50$ in the downlink case). The throughput then reaches an almost constant value, which depends on the frame duration. Among the downlink curves, there is a small throughput gain with longer frames. This is because the MAC overhead due to the transmission of MAPs decreases slightly when the frame duration increases. On the other hand, the uplink throughput improvement with longer frames is much more evident. In any case, all the downlink curves lie significantly above the uplink ones. This is because uplink data transmission incurs both in the additional delay due to requesting bandwidth and the overhead of prepending physical preambles to burst of PDUs.

To summarize, there is a trade-off between average delay and throughput with respect to frame duration. However, the dependence of the downlink performance on the frame duration is weaker than that of the uplink performance.

We now evaluate how the offered load partitioning affects the system performance in terms of throughput. This evaluation was carried out for all the three frame durations considered previously (i.e., 5 ms, 10 ms, 20 ms). Results are shown for the 5 ms frame duration since they scale according to the frame duration as previously shown for the throughput and average delay. As reported in Table 4, we first carry out a set of simulations in which the offered load N increases from 10 to 90 in steps of 10 units by varying only one factor (S, C, W) at a time. Unlike the frame duration, the offered load partitioning does not significantly affect the performance of downlink connections in terms of throughput since: 1) there is no need to use bandwidth request mechanisms, and 2) there is no physical preambles

TABLE 4
Offered Load Partitioning—First Set of Simulations

Scenario identifier	S	C	W	N	Offered load (Kb/s)
SS	10 → 90	1	1	10 → 90	147 → 13,230
conn	10	1 → 9	1	10 → 90	147 → 13,230
source	10	1	1 → 9	10 → 90	147 → 13,230

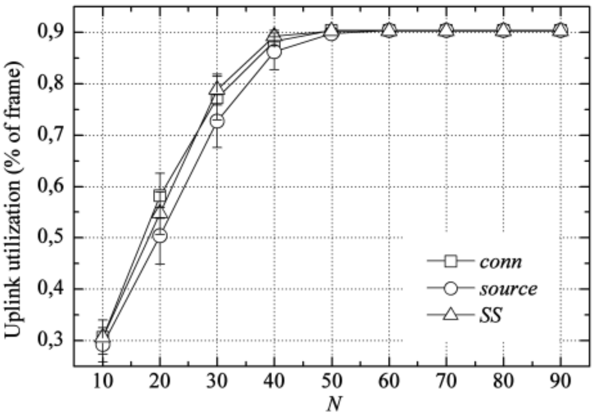


Fig. 3. Utilization versus offered load.

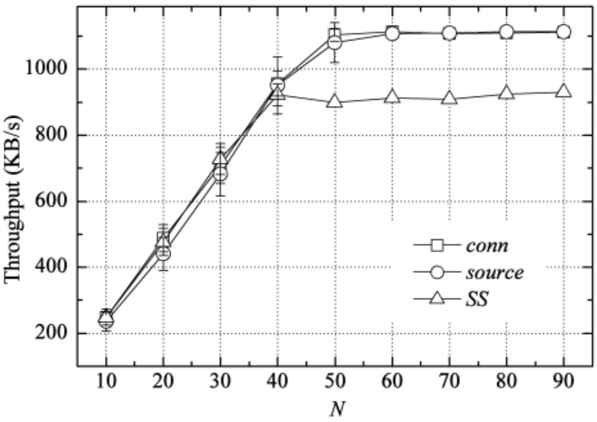


Fig. 4. Throughput versus offered load.

overhead due to the multiple transmitters, hence, the BS can exploit the overall downlink bandwidth irrespectively of the way traffic is shared. Thus, for the rest of this section, we only consider uplink traffic.

Fig. 3 shows the uplink utilization versus the offered load and highlights that the utilization does not change significantly for values of N greater than or equal to 50. This is because, with $N = 50$, the system is overloaded, i.e., all connections are almost always backlogged and there is packet loss (not shown) due to buffer overflow. Note that the utilization asymptotically reaches the value of 0.90. In fact, a portion of the uplink subframe is always allocated to contention slots.[6] Thus, when the system is overloaded, the uplink subframe is busy at the maximum possible extent, regardless of how the offered load is partitioned.

Fig. 4 shows the uplink throughput versus the offered load. As expected, there is no packet loss when the system is underloaded ($N < 50$). On the other hand, the offered load partitioning significantly affects the throughput when the system is overloaded ($N \geq 50$). Specifically, the *SS* throughput is significantly lower than the *conn* and *source* throughput. To explain this behavior, we analyze the average number of SSs served per frame. The results are shown in Fig. 5. The *conn* and *source* curves lie below the value of 10 (which is the number of SSs in the system) for any value of N. Instead, in the *SS* case, there are on average up to 32 SSs served per frame, i.e., in the *SS* case, the average number of SSs served per frame is approximately 20 percent higher than in the *conn* and *source* cases. Thus, the number of physical preambles used for transmission is on average 20 percent higher in the *SS* case compared to the other two cases. This explains why the *SS* throughput is lower than that of *conn* and *source* cases.

Furthermore, Fig. 5 shows a small difference between the *conn* and *source* curves. This can be explained as follows: An SS requests bandwidth for each connection independently. The BS keeps track of the busy connections of each SS, and it serves them individually. In fact, in the *source* case, the SS requests for its connection an amount of bandwidth which increases with N. In the *conn* case, the amount of bandwidth requested is independent of N, given that a connection traffic load does not change. Thus, in the *source* case, the uplink grants are likely to be larger, which entails a slightly smaller average number of SSs served per frame.

The achievable uplink throughput thus seems to depend on how the offered load is partitioned in the system. More specifically, the throughput depends on the number of SSs, whereas it does not depend on both the offered load per connection and the number of connections per SS.

In the results discussed so far, we have increased the offered load by varying the value of only one of the S, C, and W system parameters at a time. We now vary a combination of those parameters. Specifically, we first increase the number of SSs from five to 20, while mixing the number of connections per SS and the number of sources per connection such that their product is constant and equal to $C \times W = 6$. Then, we increase the number of SSs up to 60, while the other two parameters (C, W) are set to (2, 1) and (1, 2), respectively. As a consequence, the offered load N varies in the range [30, 120]. The parameter values of this second set of simulations are summarized in Table 5. Fig. 6 shows the throughput against the offered load. The throughput in the case where $C \times W = 6$, represented with open symbols, is much higher than the throughput in the case where $C \times W = 2$, represented with solid symbols. This is because, in the former case, the number of SSs is much lower than in the latter case. Thus, for the cases where $C \times W = 6$, the curves of the throughput

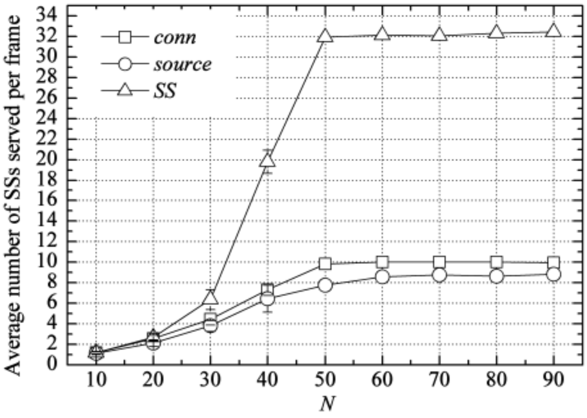


Fig. 5. Average number of SSs served per frame versus offered load.

[6] The number of OFDM symbols needed for $BW_{min} = 7$ contention slots is 14, which is about the 10 percent of the total number of OFDM symbols in a frame (147).

TABLE 5
Offered Load Partitioning—Second Set of Simulations

Scenario identifier	*S*	*C*	*W*	*N*	Offered load (Kb/s)
(1, 6)	5 → 20	1	6	30 → 120	183.8 → 2205.0
(2, 3)	5 → 20	2	3	30 → 120	183.8 → 2205.0
(3, 2)	5 → 20	3	2	30 → 120	183.8 → 2205.0
(6, 1)	5 → 20	6	1	30 → 120	183.8 → 2205.0
(1, 2)	15 → 60	1	2	30 → 120	183.8 → 2205.0
(2, 1)	15 → 60	2	1	30 → 120	183.8 → 2205.0

versus the offered load are almost the same as those reported in Fig. 4 for the *conn* and *source* cases, at least in the range [30, 90] of N. On the other hand, for low values of C and W, the main component which contributes to the offered load is the number of SSs, which lowers the throughput. This second set of simulations is further confirmation of the fact that the throughput is mostly affected by the overhead due to the transmission of physical preambles, which increases with the number of SSs.

4.2 Bandwidth Request Analysis

In this section, we investigate the relative effectiveness of the bandwidth request mechanisms with data traffic. Again, the scenario under consideration is reported in Table 4. Since the BE scheduling service is used, the SSs request bandwidth from the BS both by sending contention bandwidth requests and by piggybacking bandwidth requests on outgoing PDUs. Thus, the following analysis is aimed at understanding under what conditions (if any) one mechanism takes over from the other.

Fig. 7 shows the average number of bandwidth requests received by the BS per uplink subframe, both in response to a broadcast poll (N_C) and piggybacked on PDUs (N_P), versus the offered load N. When the system is underloaded (i.e., $N < 20$), most incoming SDUs at each connection are served before the application generates a new SDU. Thus, connections are often found idle by SDU arrivals and this leads to higher values of N_C compared to N_P. On the other hand, when N increases beyond 20, the probability that connections are found idle by SDU arrivals decreases while the piggybacking mechanism tends to take over from the contention mechanism. Although, for N greater than 50, the confidence intervals are very high, the N_C curves are very close to zero, i.e., in all cases, for $N > 50$, N_C is almost negligible.

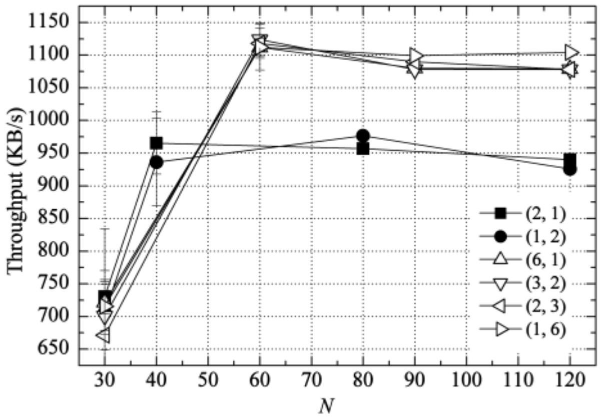


Fig. 6. Throughput versus offered load.

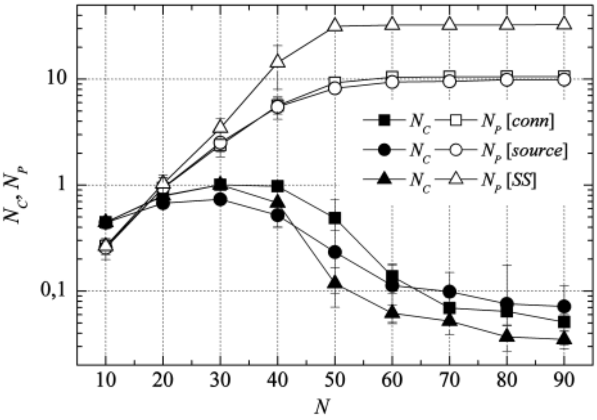


Fig. 7. Number of bandwidth requests (contention-based and piggybacked) per uplink subframe versus offered load.

Note that the curves in Fig. 7 exhibit a behavior which depends on how the offered load is partitioned. Let us first consider the N_C curves for $N < 50$ (for N greater than 50, the confidence intervals are so high that it does not make any sense to make comparisons among the various curves). Note that the *source* curve lies below the other ones. This can be explained as follows: SSs only request bandwidth for idle connections. In the *source* case, irrespectively of N, the number of connections is constant and equal to 10, one for each SS. On the other hand, in the *SS* and *conn* cases, the number of connections is proportional to N. Thus, the number of connections that are idle is lower on average in the *source* case than in the other two cases. With regard to the *SS* and *conn* curves, they almost coincide until $N < 40$. Afterward, the N_C *SS* curve has a sharper drop than the *conn* curve because the system gets overloaded earlier in the former case (see Fig. 4). As far as N_P is concerned, all the curves are almost constant when the system is overloaded. In this condition, the connections' buffers are nearly always full, thus almost each burst of PDUs from the same connection carries a piggybacked bandwidth request. The *SS* curve lies significantly above the other curves because the number of uplink grants per subframe is much higher (see Fig. 5). To summarize, when the system is lightly loaded, the most commonly used mechanism for requesting bandwidth is contention. Once the system gets overloaded, the piggyback mechanism takes over from the contention one and is then exploited in a greedy fashion.

We now evaluate the impact of BW_{min} on the performance, in terms of average delay and throughput. Specifically, we first analyze the average delay with the system underloaded ($N \leq 30$). Then, we focus on the throughput when the system is overloaded ($N = 90$). Fig. 8 shows the average delay with $N = 10$, 20, and 30, when BW_{min} increases from 0 to 12. Since, for these values of N and BW_{min}, the traffic partitioning does not significantly affect the results, for the sake of brevity, we only show the *SS* case. Regardless of the number of SSs, increasing the BW_{min} value reduces the probability of collision among bandwidth requests initially. Hence, the average delay first decreases, and then it tends to increase

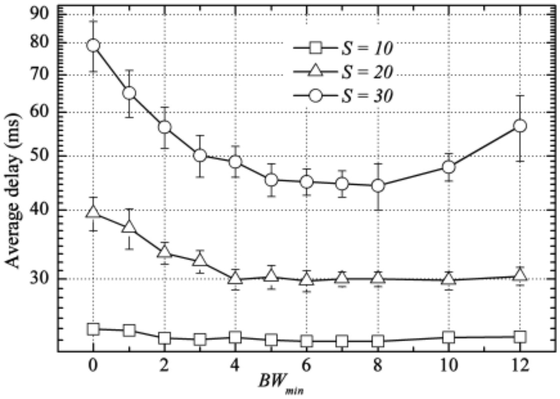


Fig. 8. Average delay versus BW_{min}.

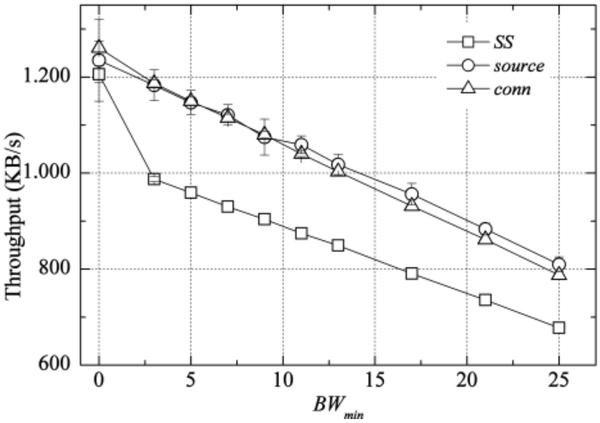


Fig. 9. Throughput versus BW_{min}.

as the bandwidth available for data transmissions decreases.[7] Note that the capacity reserved for contention bandwidth requests is almost entirely wasted when the system tends to become overloaded since piggybacking takes over from contention as highlighted by Fig. 7. Note also that $BW_{min} = 7$ minimizes the average delay, and this justifies the assumption made at the beginning of Section 4.

So far, we have analyzed the impact of BW_{min} on the average delay. Let us now analyze the impact of the BW_{min} on the throughput when the system is overloaded. Fig. 9 reports the throughput in the *SS*, *conn*, and *source* cases with $N = 90$ when BW_{min} increases from 0 to 25. The throughput decreases when BW_{min} increases, regardless of how the offered load is partitioned. However, the curves related to the *source* and *conn* cases lie above the *SS* curve. This was justified in Section 4.1. Furthermore, while the *source* and *conn* curves overlap almost perfectly for BW_{min} values up to approximately 11, for $BW_{min} > 11$, the *source* throughput is slightly higher than the *conn* throughput. This is because, as shown in Fig. 5, the *conn* case requires a slightly higher number of physical preambles per uplink subframe. Note that $BW_{min} = 0$ is the only value for which the *SS* throughput is the same as in the *conn* and *source* cases. We showed that the prominent bandwidth request mechanism when the system is overloaded is piggybacking. However, this mechanism comes into operation when connections move from the idle state to the backlogged state and this is achieved when SSs request bandwidth using contention slots for idle connections. Since no capacity is reserved for contention access and the system is overloaded, contention slots (which are acquired from the unused slots in the current uplink subframe) are very sporadic (we measured one contention slot every 12.6 frames, on average, in the *SS* case). Therefore, it is hard for an SS to request bandwidth using contention, and this significantly reduces the average number of SSs served per frame. Thus, with $BW_{min} = 0$, the *SS* case does not require a higher number of physical preambles than those required in the other two cases. This explains the same throughput value in the *SS*, *conn*, and *source* cases for $BW_{min} = 0$.

The results presented in this section have an immediate consequence. As we mentioned at beginning of the section, the MAC mechanisms for supporting the nrtPS scheduling service do not significantly improve the performance with respect to the BE scheduling service. Recall that, from the MAC mechanisms standpoint, the difference between the nrtPS and BE services is that the BS provides nrtPS connections with periodic unicast polls on a time-scale of one second or less. However, we showed that, with the system overloaded, the connections almost always exploit piggybacking to request bandwidth. On the other hand, with the system underloaded, the time-scale of the unicast polls to nrtPS connections is larger than the time needed for requesting bandwidth using contention. We confirmed this behavior by rerunning the whole set of simulations with the nrtPS scheduling service. For all the metrics evaluated, although not reported here for the sake of brevity, the difference with BE is negligible.

4.3 Evaluation of Multimedia Traffic

In this section, we evaluate the system performance with multimedia traffic, i.e., with traffic that poses stringent delay requirements. First, we evaluate how the frame duration affects the performance in terms of delay. Then, we investigate how the offered load partitioning affects the performance of uplink connections. Finally, we compare the performance of rtPS and nrtPS, when they coexist within the same SS. We ran scenarios with videoconference traffic, as described in Section 3. Thus, the minimum traffic unit is 71.5 Kb/s. Furthermore, in this section, we use the same notation previously introduced for data traffic. Since SSs never request bandwidth on a contention basis, we assume that $BW_{min} = 0$.

We evaluate first how the frame duration affects the delay of uplink and downlink connections. To this aim, we set up a scenario with a variable number of SSs increasing from 10 to 90. Each SS has one uplink and one downlink connection, respectively, each carrying a videoconference source, i.e., $C = 1$, $W = 1$. Fig. 10 shows the 95th percentile of the delay. As expected, the uplink curves are much higher than the downlink ones. In fact, uplink connections experience the additional delay of requesting bandwidth to the BS. However, the curves are almost constant when the

[7] The higher the BW_{min}, the lower the bandwidth available for the transmission of uplink data, since each broadcast poll consists of two OFDM symbols.

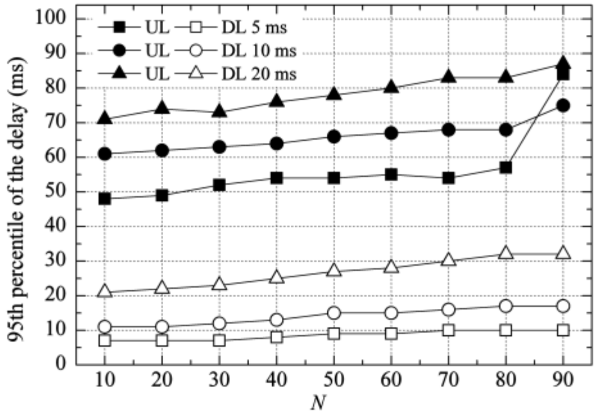


Fig. 10. Uplink/downlink 95th percentile of the delay versus offered load.

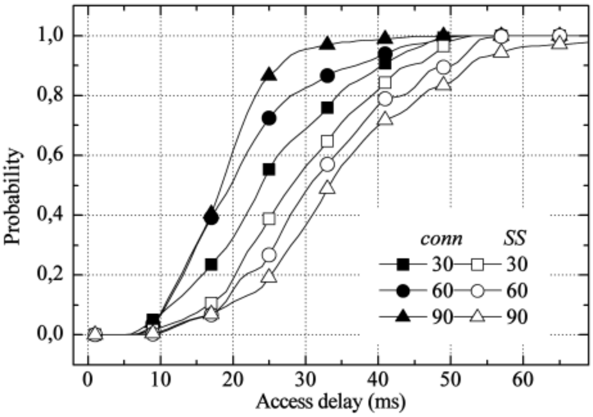


Fig. 11. CDF of the delay in the *conn* and *SS* cases with 30/60/90 videoconference sources.

offered load increases because the BS schedules a unicast poll for each connection on a periodic basis, with the period equal to the interarrival time of videoconference SDUs (i.e., 33 ms). Furthermore, the longer the frame duration, the higher the curves in both the uplink and the downlink cases. This can be explained as follows: Since scheduling is performed at the beginning of each frame, the higher the frame duration, the longer (on average) the SS has to wait before using the related grant. In other words, with longer frames, the BS is less responsive to the SSs' bandwidth requests. Furthermore, the 95th percentile of the delay of downlink connections at low offered loads is almost equal to the frame duration. However, as with data traffic, the lower delay with shorter frames entails a higher overhead due to a higher number of physical preambles (uplink) and longer MAPs (downlink). Therefore, the offered load that the system is able to serve (i.e., the carried load) decreases with the frame duration. This behavior can be seen in Fig. 10, in which the uplink 5 ms curve increases sharply for $N = 90$.

We then evaluate for uplink traffic how the offered load partitioning affects the system performance, in terms of delay, with a frame duration equal to 5 ms. For the above two choices (i.e., uplink traffic and 5 msec frame duration), the same remarks made for the data analysis still hold. Table 6 reports the details of the scenarios which led to the curves reported in Fig. 11 and Fig. 12, showing the cumulative distribution functions (CDFs) of the delay for the *conn* versus *SS* and *source* versus *SS* cases, respectively, with $N = 30$, 60, and 90. Regardless of the offered load, both the *conn* and *source* cases perform better than the *SS* case. This is because the *SS* case incurs more overheads due to the transmission of a higher number of physical preambles compared to the *conn* and *source* cases. Furthermore, in the *conn* and *source* cases, the higher the offered load, the lower the delay. This counter-intuitive behavior can be explained as follows: Let us consider the case of an SDU enqueued at connection i. If the connection is busy, the related SS piggybacks a bandwidth request to the next outgoing packet from the same connection i. This is done regardless of how the offered load is partitioned. However, such an event is more likely to occur in the *source* case because there are multiple traffic sources multiplexed into connection i. In fact, the larger the value of W, the higher the probability that an arriving SDU observes a nonempty buffer. This accounts for the result in Fig. 12. On the other hand, if connection i is idle, in the *SS* and *source* cases, the SS has to wait for its next unicast poll from the BS. Instead, in the *conn* case, the BS might schedule an uplink grant to another connection j before the unicast poll to connection i is due. In this case, the SS is able to transmit a bandwidth request for connection i stealing (part of) connection j's bandwidth. The larger the value of C, the higher the probability that such an event occurs, which accounts for the result in Fig. 11.

In order to gain insight into the behavior exhibited by the curves in Fig. 11 and Fig. 12, we report the notification delay in Fig. 13. In the *conn* and *source* cases, the notification delay decreases when the offered load increases, whereas, it is almost constant (i.e., decreases slightly) in the *SS* case. This confirms the apparent anomalies highlighted by Fig. 11 and Fig. 12. Basically, the higher the number of videoconference

TABLE 6
Multimedia Traffic

Scenario ID	*S*	*C*	*W*	*N*	Offered load (Kb/s)
SS	10 → 90	1	1	10 → 90	715 → 7,150
conn	10	1 → 9	1	10 → 90	715 → 7,150
source	10	1	1 → 9	10 → 90	715 → 7,150

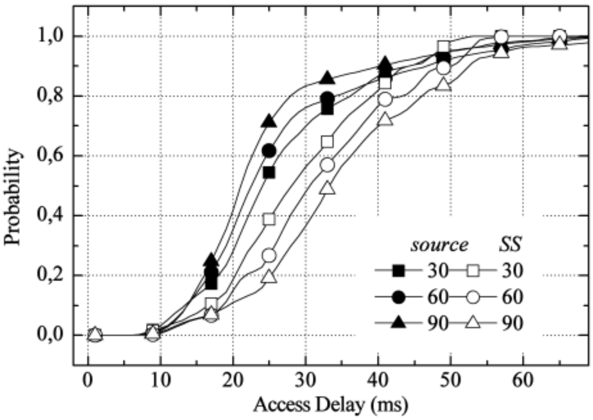


Fig. 12. CDF of the delay in the *source* and *SS* cases with 30/60/90 videoconference sources.

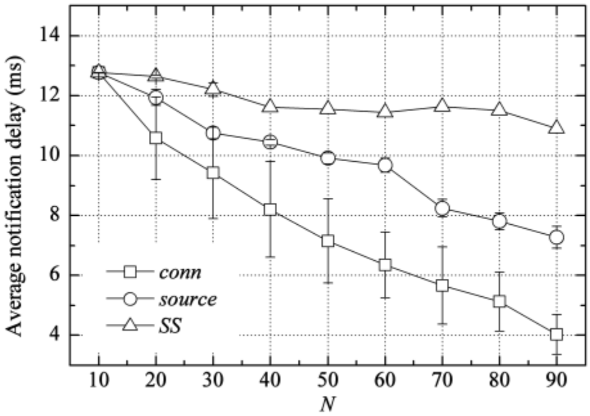


Fig. 13. Notification delay versus offered load.

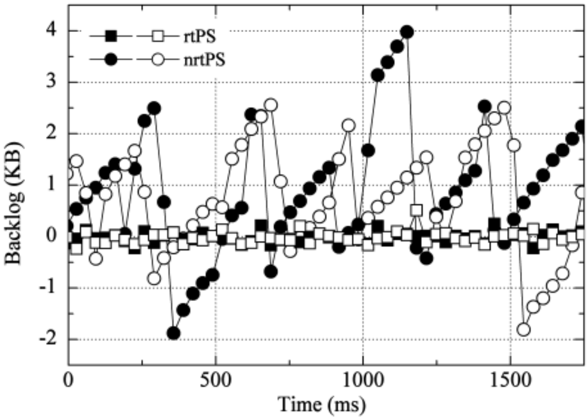


Fig. 15. Backlog error versus time with 160 videoconference sources.

sources per SS, the lesser the 95th percentiles of the delay. In fact, SSs benefits from the statistical multiplexing of multiple videoconference traffic sources because they can exploit the piggybacking/bandwidth stealing mechanisms to request bandwidth before they are issued a unicast poll.

We now evaluate the effectiveness of the rtPS and nrtPS when both scheduling services are employed at each SS to serve videoconference traffic. To this aim, we set up a scenario with a variable number of connection pairs for each SS. The number of SSs is fixed and equal to 10. Each connection pair consists of an rtPS connection and an nrtPS connection, each loaded with a videoconference source. We set the minimum reserved rate to the same value for both the rtPS and nrtPS connections. Thus, rtPS and nrtPS connections only differ in how they request bandwidth. Fig. 14 shows the 95th percentile of the delay against the number of videoconference sources, which increases from 20 to 160. The rtPS curve decreases slightly when the offered load increases. This behavior is due to the multiplexing of multiple videoconference sources into each SS and was thoroughly investigated in the first part of this subsection. The nrtPS curve instead increases slightly when the system is underloaded, whereas it increases sharply as soon as 100 videoconference sources are served. In fact, when the system is underloaded, the interarrival time of videoconference SDUs is almost always greater than the sum of: 1) the time needed for the SS to transmit a contention bandwidth request, plus 2) the time for the BS to schedule enough bandwidth to entirely serve that SDU. On the other hand, when the system is heavily loaded, nrtPS connections are not able to request bandwidth on time. Unlike nrtPS, the high offered load does not affect the notification delay of rtPS connections and, thus, the delay, which are polled on a periodic basis.

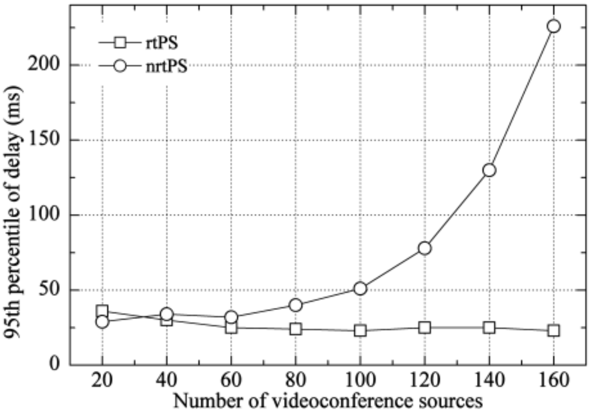


Fig. 14. Ninety fifth percentile of the delay versus offered load.

To conclude the comparison between nrtPS and rtPS scheduling services, we evaluate the backlog gap, as defined in Section 3.2, when the system is heavily loaded (i.e., 160 videoconference sources). Fig. 15 reports the backlog gap against time with 160 videoconference sources. More specifically, the curves represent the behavior of two connection pairs that belong to two randomly chosen SSs. Since the BS is not immediately aware of SDUs that arrive at the connections of an SS, it usually underestimates the backlog (i.e., bandwidth requirements) when scheduling uplink grants to SSs. However, as can be seen in Fig. 15, there are cases in which the BS overestimates the backlog. This is due to the fact that the BS allocates bandwidth to the SS as a whole, whereas the SSs request bandwidth for specific connections and this may lead to bandwidth stealing (as explained earlier in this section). In any case, the lower the notification delay, the smaller the estimation error of the BS. This is confirmed in Fig. 15, which shows that the backlog gap with rtPS is much smaller than that with nrtPS.

In conclusion, rtPS outperforms nrtPS in terms of delay in the simulated scenarios. This is especially true when the system is heavily loaded. On the other hand, nrtPS connections are provided with a (slightly) better service than rtPS connections only when the system is extremely underloaded. We also ran simulation scenarios with VoIP traffic (as defined in Section 3) and with a mix of VoIP and videoconference traffics multiplexed into different connections at each SS. Even though the results are quantitatively different, because VoIP and videoconference traffic have inherently different traffic characterizations, it is possible to draw from them the same conclusions as in the case of videoconference traffic alone. Again, for the sake of brevity, we have not reported these results in the paper.

5 Conclusions

This paper presents a simulation study of the IEEE 802.16 MAC protocol operated with the WirelessMAN-OFDM air interface and with full-duplex stations. We have evaluated the system performance under different traffic scenarios and by varying the values of a set of relevant (from an engineering standpoint) system parameters.

With regard to data traffic, we concluded that there is a trade-off between average delay and throughput with respect to frame duration. Specifically, the longer the frame durations, the higher the average delays (the lower the throughput). Furthermore, we found that the overhead due to the transmission of physical preambles increases with the number of SSs. Hence, when the system is overloaded, the achievable uplink throughput decreases when the number of SSs increases. Finally, we have shown that SSs are able to request uplink bandwidth to the BS efficiently using piggybacked bandwidth requests, unless the system is lightly loaded. For this reason, under the considered scenarios, we proved that the nrtPS scheduling service does not improve the performance of uplink connections with respect to the BE scheduling service in terms of throughput and average delay.

As far as traffic with QoS requirements, we have found that the performance of uplink connections, in terms of delay, is highly dependent on the delay introduced by the bandwidth request mechanism. Specifically, having shorter frame duration entails lower delays, even though it increases the MAC overhead, thus reducing the throughput. Moreover, SSs might effectively exploit piggybacking and bandwidth stealing to improve the delay performance. This can only be done if there are multiple traffic sources in the same SS, either multiplexed into the same connection or carried by separate connections. Finally, we have shown that rtPS outperforms nrtPS in terms of delay, at least under the considered scenarios.

Acknowledgments

The results reported in this paper were obtained in the framework of a project financed by Nokia to the Dipartimento di Ingegneria dell'Informazione of the University of Pisa.

References

[1] D.I. Axiotis, T. Al-Gizawi, K. Peppas, E.N. Protonotarios, F.I. Lazarakis, C. Papadias, and P.I. Philippopoulos, "Services in Interworking 3G and WLAN Environments," *IEEE Wireless Comm.*, vol. 11, no. 5, pp. 14-20, Oct. 2004.

[2] P.T. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way Conversation," *Bell System Technical J.*, vol. 48, pp. 2445-2472 Sept. 1969.

[3] Y. Cao and O.K. Li, "Scheduling Algorithms in Broad-Band Wireless Networks," *Proc. IEEE*, vol. 89, no. 1, pp. 76-87 Jan. 2001.

[4] D.-H. Cho, J.-H. Song, M.-S. Kim, and K.-J. Han, "Performance Analysis of the IEEE 802.16 Wireless Metropolitan Area Network," *Proc. First Int'l Conf. Distributed Frameworks for Multimedia Applications (DFMA '05)*, pp. 130-137, Feb. 2005.

[5] C. Cicconetti, C. Eklund, L. Lenzini, and E. Mingozzi, "Quality of Service Support in IEEE 802.16 Networks," *IEEE Network Magazine*, vol. 20, no. 2, Mar. 2006.

[6] C. Eklund, R.B. Marks, K.L. Stanwood, and S. Wang, "IEEE Standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," *IEEE Comm. Magazine*, vol. 40, no. 6, pp. 98-107, June 2002.

[7] F.H.P. Fitzek and M. Reisslein, "MPEG4 and H.263 Video Traces for Network Performance Evaluation," *IEEE Network Magazine*, vol. 15, no. 6, pp. 40-54 Nov. 2001.

[8] A. Ghosh, D.R. Wolter, J.G. Andrews, and R. Chen, "Broadband Wireless Access with WiMax/802.16: Current Performance Benchmarks and Future Potential," *IEEE Comm. Magazine*, vol. 43, no. 2, pp. 129-136, Feb. 2005.

[9] O. Gurbuz and E. Ayanoglu, "A Transparent ARQ Scheme for Broadband Wireless Access," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '04)*, pp. 423-429, Mar. 2004.

[10] O. Gusak, N. Oliver, and K. Sohraby, "Performance Evaluation of the 802.16 Medium Access Control Layer," *Lecture Notes on Computer Science*, vol. 3280, pp. 228-237, 2004.

[11] C. Hoymann, "Analysis and Performance Evaluation of the OFDM-Based Metropolitan Area Network IEEE 802.16," *Computer Networks*, vol. 49, no. 3, pp. 341-363, Oct. 2005.

[12] *IEEE 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE, Oct. 1,2004.

[13] *IEEE P802.16/Cor1/D2, Corrigendum to IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE, Apr. 2005.

[14] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip," *IEEE J. Selected Areas in Comm.*, vol. 9, no. 8, pp. 1265-1279, 1991.

[15] I. Koffman and V. Roman, "Broadband Wireless Access Solutions Based on OFDM Access in IEEE 802.16," *IEEE Comm. Magazine*, vol. 40, no. 4, pp. 96-103, Apr. 2004.

[16] A.M. Law and W.D. Kelton, *Simulation Modeling and Analysis*, third ed. McGraw-Hill, 2000.

[17] H. Lee, T. Kwon, and D.-H. Cho, "An Efficient Uplink Scheduling Algorithm for VoIP Services in IEEE 802.16 BWA Systems," *Proc. IEEE Vehicular Technology Conf. (VTC '04)*, pp. 3070-3074, Sept. 2004.

[18] M. Molina, P. Castelli, and G. Foddis, "Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering through HTML-REDUCE," *IEEE Network Magazine*, pp. 46-55 May 2000.

[19] J.F. Mollenauer, J. Klein, and B. Petry, "An Efficient Media Access Control Protocol for Broadband Wireless Access Systems," *IEEE 802.16 Broadband Wireless Access Working Group*, Oct. 1999.

[20] Motorola, *Evaluation Methods for High Speed Downlink Packet Access (HSDPA)* TSG-R1 document, TSGR#14(00)0909, 2000.

[21] Redline Communications, "Redmax Base Station Datasheet AN-100U," http://www.redlinecommunications.com/, 2005.

[22] M. Shreedhar and G. Varghese, "Efficient Fair Queueing Using Deficit Round Robin," *IEEE Trans. Networking*, vol. 4, no. 3, pp. 375-385, June 1996.

[23] D. Stiliadis and A. Varma, "Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms," *IEEE/ACM Trans. Networking*, vol. 6, pp. 675-689, Oct. 1998.

[24] WiMAX Forum, http://www.wimaxforum.org/, 2005.

[25] WiMAX Forum, "Initial Certification Profiles and the European Regulatory Framework," WiMAX Forum Regulatory Working Group, Sept. 2004.

[26] K. Wongthavarawat and A. Ganz, "Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems," *Int'l J. Comm. Systems*, vol. 16, no. 1, pp. 81-96, Feb. 2003.

Claudio Cicconetti graduated in computer systems engineering from the University of Pisa, Italy, in October 2003. He is currently pursuing the PhD degree at the same university. His research interests include quality of service in wireless networks, medium access protocols for mobile computing, and mesh networks. He is involved in the EuQoS (End-to-end Quality of Service support over heterogeneous networks) project, which participates in the EU Information Society Technologies (IST) Program.

Alessandro Erta graduated (cum laude) in computer systems engineering from the University of Pisa, Italy, in February 2005. He is currently a PhD student at IMT Lucca, Institute for Advanced Studies. His research interests include quality of service in wireless networks, the design and performance evaluation of MAC protocols, and scheduling algorithms for wireless networks.

Luciano Lenzini received a degree in physics from the University of Pisa, Italy. He joined CNUCE, an institute of the Italian National Research Council (CNR), in 1970. 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 and the Quality of Service provision in integrated and differentiated services networks. He is currently on the editorial boards of *Computer Networks* and the *Journal of Communications and Networks*. He served as chairman for the 1992 IEEE Workshop on Metropolitan Area Networks and for the 2002 European Wireless (EW '02) conference. He has directed several national and international projects in the area of computer networking.

Enzo Mingozzi received the Laurea (cum laude) degree and the PhD degree in computer systems engineering in 1995 and 2000, respectively, from the University of Pisa. He has been an associate professor with the faculty of engineering at the University of Pisa, Italy, since January 2005. His research activities span several areas, including design and performance evaluation of multiple access protocols for wireless networks, QoS provisioning, and service integration in IP networks. He has been involved in several national (FIRB, PRIN) and international (Eurescom, IST) projects, as well as research projects supported by private industries (Telecom Italia Lab, Nokia). He actively took part in the standardization process of HIPERLAN/2 and HIPERACCESS networks in the framework of the ETSI project BRAN (Broadband Radio Access Networks).