An Efficient Cross Layer Scheduler for Multimedia Traffic in Wireless Local Area Networks with IEEE 802.11e HCCA*

Claudio Cicconetti, **Luciano Lenzini**, **Enzo Mingozzi**, **Giovanni Stea**
c.cicconetti,l.lenzini,e.mingozzi,g.stea@iet.unipi.it
Dipartimento di Ingegneria dell'Informazione, University of Pisa, Italy

This paper proposes a scheduling algorithm, namely Wireless Timed Token Protocol (WTTP), for the Hybrid Coordination Function (HCF) Controlled Channel Access (HCCA) in IEEE 802.11e. WTTP provides traffic streams with a minimum reserved rate, as required by the standard, and it accounts for two types of traffic streams simultaneously, depending on the corresponding application: constant bit rate, which are served according to their rate, and variable bit rate traffic streams. The latter are guaranteed a minimum rate, but they are also allowed to exploit unused bandwidth, which preserves small access delays in case of bursty arrivals. Additionally, WTTP shares the capacity which is not reserved for QoS traffic streams transmissions among traffic flows with no specific QoS requirements. We also propose a strategy for the QoS Access Point to infer the idle/busy status of the uplink transmission buffers, based on cross layer information made available at the MAC by the application layer, which allows channel capacity to be saved. WTTP exhibits O(1) per packet computational complexity. We evaluate the performance of WTTP via simulation under different traffic conditions, and we investigate its resilience to variations of different system parameters.

I. Introduction

In recent years Wireless Local Area Networks (WLANs) have become very popular due to the increasing interest of residential and office customers in a ubiquitous service. Within this framework, the IEEE 802.11 has established itself as the worldwide standard in indoor and outdoor wireless LANs [19]. On the other hand, the high level of performance provided by the wired networks is driving users towards an emerging set of applications with Quality of Service (QoS) requirements, such as phone or videoconference over IP networks. QoS applications exhibit different traffic arrival patterns, being either Constant Bit Rate (CBR) or Variable Bit Rate (VBR), and they require strict delay bounds. Those applications will coexist with legacy TCP-based applications, which have no specific QoS requirements. In order to support applications with QoS requirements, IEEE recently published an amendment to the IEEE 802.11 standard, namely IEEE 802.11e [20], which adds the Hybrid Coordination Function (HCF). The latter specifies two access mechanisms: Enhanced Distributed Channel Access (EDCA), which is based on a distributed control and enables prioritized channel access, and HCF Controlled Channel Access (HCCA), which on the other hand requires centralized scheduling, and allows the applications to negotiate parameterized service guarantees in the context of Traffic Streams (TSs). The Hybrid Coordinator (HC) provides scheduling for both the QoS Access Point (QAP) and QoS Stations (QSTAs), by dispensing transmission opportunities (TXOPs) of variable size to both downlink and uplink TSs. Downlink TXOPs are granted for transmission to QoS Stations (QSTAs) from the QAP. Uplink TXOPs consist of data messages transmitted by QSTAs in response to individual polling messages sent from the QAP to TSs. Although the IEEE 802.11e standard does not specify a mandatory scheduling algorithm, it mandates TSs to be provided with a minimum reserved rate, under controlled channel conditions.

Providing an efficient schedule of both CBR and VBR traffic simultaneously is a daunting task. While CBR traffic has a regular arrival pattern, VBR traffic rate varies greatly, with a *peak* rate which is often much larger than the *average* rate. Therefore, unlike CBR, VBR traffic cannot be served efficiently when TSs are provided with a fixed service rate. In fact, if a VBR TS is guaranteed a minimum rate equal to its average rate, large bursts are likely to experience high, unfeasible delays. On the other hand, reserving the peak rate would entail a substantial wastage of the reserved capacity. Several HCCA scheduling

*A preliminary version of this paper has appeared in [9]

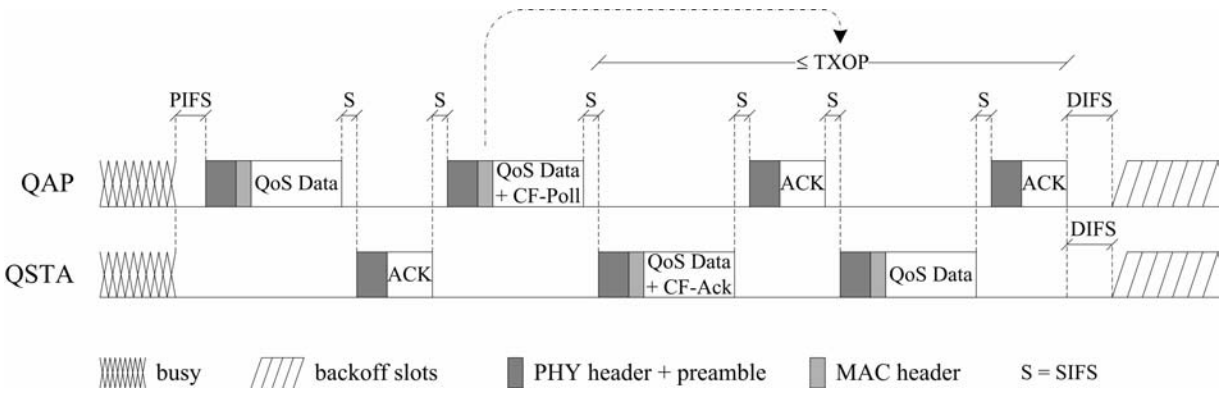

Figure 1: HCCA sample frame exchange sequence

algorithms have been proposed in the literature (see for example [2] [3] [7] [14] [24] [33]). Some of these (e.g., [7] [33]) are explicitly tailored to CBR traffic, and perform poorly with VBR traffic [11]. Others, instead, (e.g., [2] [3] [14] [24]) try to dynamically adjust the TXOP duration to react to variations in the arrival pattern. These algorithms, although more efficient, often exhibit a large computational complexity.

In this paper, we propose a simple scheduling algorithm, namely the Wireless Timed Token Protocol (WTTP), which is explicitly tailored to account for both CBR and VBR traffic simultaneously. More specifically, WTTP provides CBR TSs with a fixed capacity. VBR TSs, instead, are reserved a minimum rate, but they are also allowed to exploit *unused* bandwidth, i.e. the one which is currently not reserved to TSs, to limit the delay of occasional bursts. WTTP shares bandwidth among both uplink and downlink TSs. Unlike downlink TSs, whose queues reside at the QAP, uplink TSs queues reside at the QSTAs. Thus, the QAP does not know their backlog state. When the QAP polls an idle TS, some channel bandwidth is wasted. In order to solve this problem, we devised a strategy to infer the backlog state of uplink TSs at the QAP. The QSTAs piggyback the amount of buffered data on outgoing uplink data messages, which is a standard (although optional) capability of IEEE 802.11e, and advertise the nominal interval at which packets are generated by the applications. Such information is made available at the QAPs MAC by the applications, in a cross-layer approach. Finally, WTTP also strives to minimize the amount of channel capacity scheduled as TXOPs, provided that the minimum reserved rates are guaranteed. In fact, the standard specifies that the channel capacity not consumed by the QAP as downlink/uplink TXOPs may be used by contention-based access functions, either the Distributed Coordination Function (DCF) or EDCA, which are tailored for applications with no specific QoS requirements, which can thus coexist with QoS traffic. Therefore, minimizing the overall length of TXOPs also improves the performance of such kind of traffic.

As its name suggests, WTTP is inspired by the Timed Token Protocol (TTP), in that it applies the same rules to either schedule extra bandwidth allocations to VBR traffic flows or let contention-based traffic access the medium. WTTP has a low computational complexity, requiring $\mathcal{O}(1)$ operations per packet transmission. Therefore, it is more efficient compared to those algorithms proposed in the literature which are based on sorting packet deadlines. Moreover, it does not require complex algorithms to be applied at the QAP to estimate buffer occupancy at QSTAs.

The rest of the paper is organized as follows. Section II introduces the IEEE 802.11 standard and the IEEE 802.11e enhancements. In Section III the WTTP scheduling algorithm is described. The related work is discussed in Section IV. Section V describes the simulation results and conclusions are drawn in Section VI.

II. IEEE 802.11e MAC protocol

The IEEE 802.11e standard [19] describes those enhancements to the MAC services and functions of the IEEE 802.11 standard [20] aimed at enabling Quality of Service provisioning. The IEEE 802.11 specifies two access functions, respectively Distributed Coordination Function (DCF) and Point Coordination Function (PCF). In IEEE 802.11e, two additional access mechanisms are defined: the Enhanced Distributed Channel Access (EDCA) and the HCF Controlled Channel Access (HCCA), both in the context of the Hybrid Coordination Function (HCF). To ensure com-

patibility with legacy devices, the standard allows the coexistence of DCF and PCF with HCF.

EDCA achieves traffic differentiation in a distributed manner [28]. At each QoS Station (QSTA) four Access Categories (ACs) are defined, whose traffic contends for the medium using different access parameters, such as the minimum contention window. Moreover, QSTAs are allowed to transmit multiple frames without contending for medium access within an EDCA transmission opportunity (TXOP), whose maximum duration depends on the AC. In an infrastructure network, the set of parameters for each AC is advertised by the QAP, which also performs the admission control.

On the other hand, HCCA is a centralized access mechanism controlled by the Hybrid Coordinator (HC). For the ease of reading, since the HC is collocated with the QAP, we will not distinguish between them, and assume that the QAP performs bandwidth management as well. Each QSTA may have up to eight established Traffic Streams (TSs). A TS is characterized by a Traffic Specification (TSPEC) negotiated between the QSTA and the QAP. A TSPEC consists of many parameters, most of which optional, which can be used to characterize the application traffic and requirements in several ways. In this work, we consider the subset of six mandatory TSPEC fields only, which are, for TS i: Mean Data Rate (R_i), Nominal Service Data Unit (SDU) Size (N_i), Maximum SDU Size (M_i), Minimum PHY Rate (Γ_i), Delay Bound (D_i), Maximum Service Interval (MSI_i). The first five parameters characterize the TS. The delay bound, in particular, specifies the maximum lifetime of an SDU at the QSTA: if an SDU experiences a delay greater than D_i, it is dropped by the QSTA. MSI_i, instead, is a service requirement, and it corresponds to the maximum time that can elapse between two subsequent polls to the same QSTA. The standard allows one to specify one between D_i and MSI_i, or both. The nominal SDU size field includes one bit (namely, *Fixed Subfield*) that is used to indicate whether the TS generates SDUs of constant size or not. Thereafter, we refer to TSs having constant size SDUs as Constant Bit Rate (CBR) TSs, and to TSs whose SDUs have variable size as Variable Bit Rate (VBR) TSs.

The QAP enforces the negotiated QoS guarantees by scheduling TXOPs during Controlled Access Phases (CAPs). CAPs are initiated by the QAP with a higher priority than that of medium access mechanisms based on contention, i.e. DCF or EDCA. More specifically, the QAP waits for the medium to remain idle for a PCF Inter-Frame Space (PIFS) before accessing the medium, where PIFS is greater than the access interval used by DCF/EDCA transmissions, and smaller than the Short Inter-Frame Space (SIFS), which separates frames within the same exchange. A CAP consists of one or more TXOPs, during which the QAP may either transmit SDUs belonging to established downlink TSs or poll one or more QSTAs by specifying the maximum time that they can occupy the medium. A QSTA is never allowed to exceed the TXOP limit imposed by the QAP, including interframe spaces and acknowledgments. If the TS of a polled QSTA is not backlogged, then the QSTA responds with a QoS CF-Null frame (or *null*, for short). Fig. 1 shows a sample frame exchange sequence during which the QAP transmits two data frames and polls the QSTA, which in turn transmits two data frames. As reported in the figure, the TXOP issued by the QAP by means of the poll piggybacked on the data message must be greater than or equal to the cumulative duration of the data messages from the QSTA, including interframe spaces and acknowledgment messages. The QAP is responsible for scheduling TXOPs so that the negotiated TSPEC parameters of admitted TSs are satisfied, under controlled operating conditions. More specifically, for any time interval $[t_1, t_2]$ greater than a minimum Specification Interval, which is advertised by the QAP, the HCCA function is committed to schedule to TS i a number of TXOPs whose cumulative duration $W_i(t_1, t_2)$ is such that:

$$W_i(t_1, t_2) \geq \left\lceil \frac{t_2 - t_1 - \Delta_i}{N_i/R_i} \right\rceil \cdot tx(N_i, \Gamma_i), \qquad (1)$$

where $tx(N_i, \Gamma_i)$ is the time to transmit an SDU of nominal size N_i at the physical rate Γ_i including MAC control frames and interframe spaces [1], and Δ_i is equal to MSI_i, if the latter is specified, otherwise it is equal to D_i ([20] par. 9.9.3.2). The IEEE 802.11e allows the QSTAs to dynamically change their physical transmission rates so as to improve the performance by following the time-varying wireless link quality. The standard does not specify the algorithm to perform rate switching, which is thus vendor-dependent. Usually, a combination of the number of retransmissions and the measured Signal-to-Noise (SNR) is used [16]. Therefore, it is possible that the current rate γ_i employed for TS i is different from the minimum PHY rate Γ_i advertised in the TSPEC. If $\gamma_i < \Gamma_i$, then

[1] If the Block Acknowledgment mechanism, which is optional in IEEE 802.11e, is enabled, $tx(N_i, \Gamma_i)$ includes the transmission duration of the *BlockAck* and *BlockAckReq* frames only, instead of one *Ack* frame for each SDU [17]

the QAP is not expected to enforce the QoS guarantees of TS i. Otherwise, the bandwidth reserved for TS i according to (1) is overprovisioned.

It is worth noting that the scheduling of TXOPs, i.e. of HCCA traffic streams, also affects the overall capacity left for contention-based access functions, i.e. EDCA and DCF.

Finally, we observe that, in order to allow for an effective bandwidth management, the QAP should know the exact backlog state of uplink TSs, whose queues physically reside at the QSTAs. Thus, the IEEE 802.11e standard defines an optional mechanism to let QSTAs vehiculate such information to the QAP: QSTAs may piggyback the length of their transmission queues on outgoing QoS frames. This is done via a dedicated field of the MAC header, which contains the backlog (in multiples of 256 bytes) of the TS to which the packet belongs.

III. Wireless Timed Token Protocol

In this section we describe the WTTP scheduling algorithm. WTTP is a round robin algorithm. A *round robin list*, whose nodes refer downlink and uplink TSs, is built at the QAP. One special node represents *contention traffic*, i.e. traffic employing contention-based access schemes (i.e. EDCA and DCF). This is shown in Fig. 2. When visiting a node in the list, the scheduler computes the *sojourn time* for that node and either schedules a TXOP of the computed sojourn time (if the node represents an uplink/downlink TS) or refrains from generating CAPs for the computed sojourn time (if the node represents contention traffic). We first describe how the round robin list is managed, and then show how the sojourn time is computed.

The round robin list management is illustrated in Fig. 2. As a general rule, nodes should be in the round robin list only if they actually have traffic to be served. For this reason, downlink TSs — which in fact reside at the QAP — are added (removed) whenever they become backlogged (idle). However, the QAP does not know the backlog state of uplink TSs, which instead reside at the QSTAs. Considering them as always backlogged is obviously not a good solution, since some capacity is wasted when an empty TS is polled. A simple, though effective, alternative is to have each QSTA piggyback the backlog of its TSs on outgoing data messages. This way, the scheduler can remove the node when the associated TS queue is reported to be empty. In order to know when to put the node back in the round robin list, we exploit the fact that many common applications, both CBR and VBR, generate packets at constant intervals (e.g. Motion Picture Expert Group (MPEG) and almost all Voice over IP (VoIP) encoders). We let each TS specify the interval at which the application generates packets. In our cross-layer architecture, this information, which belongs to the application layer, is fed to WTTP. The standard TSPEC field *minimum Service Interval* (mSI) is used to convey such information. An empty uplink TS is therefore added to the list again when a time interval equal to the packet inter-arrival time elapses, so that the QAP can assume that, according to the packet generation interval advertised by the application, at least one SDU has been generated. This way, *null* responses to polling of upstream TSs are drastically reduced. However, there is no way they can be entirely eliminated. In fact, applications can actually have silence periods (e.g., voice streams with voice activity detection). Moreover, even if an application generates packets at known, perfectly regular intervals, operating systems and hardware usually add a random jitter to packets before they are actually inserted into the TS queue, which makes the application-level information unreliable. For this reason, in Section V.F we assess the WTTP resiliency to such irregularities through simulation. However, when a QSTA responds with a *null*, the next node is immediately considered for transmission. Finally, the node representing contention-traffic is always in the round robin list, since the QAP cannot make any assumptions on forthcoming contention traffic.

Hereafter, we describe how the sojourn time is computed under WTTP. In doing this, we recall that WTTP draws its concepts from the *Timed Token Protocol* [15], used as a MAC protocol in FDDI ring networks, whose lexicon is broadly reused. A similar approach, i.e. applying the TTP rules to packet scheduling in a centralized manner, has been used by the Timed Token Service Discipline (TTSD) [26], proposed as an algorithm to schedule packets at a single output wired link for QoS and non-QoS flows.

In WTTP, a *Target Token Rotation Time* ($TTRT$) is selected as a reference round duration. The sojourn time of the QAP scheduler (*server*, hereafter) at a node is given by either one or both of the following components: a *fixed* time, called *synchronous bandwidth*, and a *variable time*, called *asynchronous bandwidth*. The former is a fixed percentage of $TTRT$, while the latter varies from round to round, and it is computed so that — if nodes fully exploit it — the server keeps a steady pace, and every node is visited each $TTRT$. More specifically, each node i has a non negative synchronous bandwidth H_i. Furthermore, a

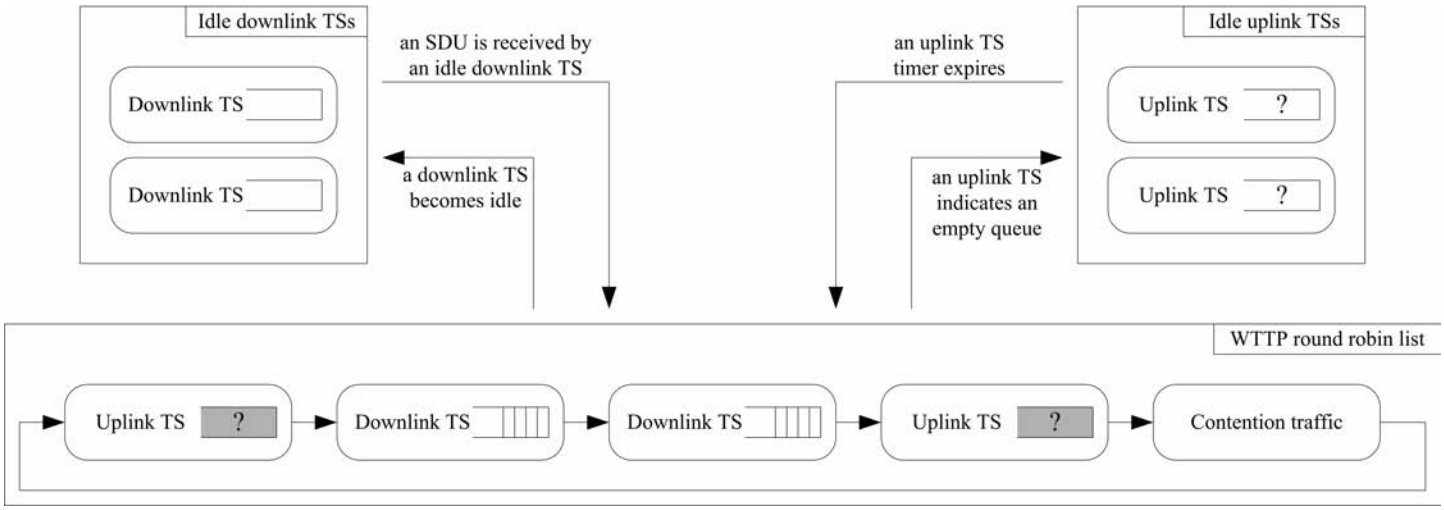

Figure 2: WTTP round robin list management

Token Rotation Timer (TRT_i), initially set to $TTRT$, counts down the time from the last server visit to obtain the maximum fair share of asynchronous bandwidth that node i can exploit. When a node is served, the *asynchronous bandwidth* is computed as follows:

$$a_i = \begin{cases} 0 & TRT_i < 0 \\ \min\{TTRT - H_i, TRT_i\} & TRT_i \geq 0 \end{cases}. \quad (2)$$

Such a mechanism allows nodes to discriminate whether they are served *earlier* than expected ($TRT_i \geq 0$), in which case they can exploit a non null asynchronous bandwidth, or *later* than expected ($TRT_i < 0$), in which case they have to refrain from further delaying the server pace. In the former case, TRT_i is reset to $TTRT$, in the latter to $TTRT + TRT_i$.

It can be easily shown that the above algorithm is the same used by stations running the TTP to regulate the circulation of the token. This implies that the *same* properties that regulate the timing of the token circulation in TTP also regulate the service guarantees in WTTP. More specifically, under the following inequality:

$$\sum_{\text{node } i} H_i + \tau \leq TTRT. \quad (3)$$

$TTRT$ is in fact the average inter-service time for a node, and $2 \cdot TTRT$ is an upper bound on the inter-service time [22]. Therefore, a node that has a synchronous bandwidth equal to H_i is in fact entitled to an average rate equal to $H_i/TTRT$ times the channel speed, and has a bounded medium access time. The term τ in (3) is an overhead which will be described later.

Note that TSs may not need the whole synchronous/asynchronous bandwidth assigned to them at a given round. In the case of a downlink TS, if all data awaiting transmission are sent, then the node is removed from the round-robin list. In the case of an uplink TS, part of the TXOP is left unused and returned to the QAP. On the other hand, since the QAP does not know if one or more stations wish to access the medium using a contention-based function, the capacity allocated to the DCF/EDCA node is always entirely consumed, i.e. the QAP does not start a new CAP until the computed sojourn time has expired. However, it may happen that a station starts the transmission of a frame, using DCF/EDCA, just before the QAP regains the control of the medium to start a new CAP. Thus, contention-based access could actually occupy the medium longer than it was supposed to. However, this time overrun, which we referred to as τ in (3), is upper bounded by the largest duration between: (i) the time to transmit a maximum size frame transmitted at the lowest physical rate with maximum amount of fragments, including MAC control frames (RTS/CTS/ACK) and interframe spaces; and, (ii) the maximum duration of an EDCA TXOP.

We now describe how the above parameters (i.e. $TTRT$ and the synchronous bandwidths H_i) are computed from the TS requirements. We assume that each TS advertises the following TSPEC fields, described in Section II, i.e., R_i, N_i, M_i, D_i. Furthermore, we assume that applications inform the QAP of their traffic generation pattern, once more according to a cross-layer approach: more specifically, they can be either CBR or VBR, i.e. generating either fixed or variable

size packets periodically, respectively. A QoS signaling scheme for IEEE 802.11e HCCA such as the one proposed in [31], which is based on the Reservation Protocol (RSVP), can be used to convey such information. This involves modifying the applications so that they communicate their application-layer QoS requirements to a MAC sub-module, which translates them into the above TSPEC parameters. This module is outside the scope of the IEEE 802.11e standard. If this cross-layer framework cannot be used, e.g. with existing applications that do not implement this kind of resource reservation, the application-layer QoS requirements can be estimated by the MAC sub-module based on traffic observation, e.g. using Kalman filters [1]. The $TTRT$ parameter is instead selected by the QAP according to the TSPEC values negotiated by QSTAs during the admission control phase. Specifically, we set the value of $TTRT$ to half the smallest delay bound, since the round duration is bounded by $2 \cdot TTRT$:

$$TTRT = \frac{1}{2} \min_i \{D_i\} . \quad (4)$$

Note that this guarantees that each TS is granted TXOPs at a sufficient rate. However, this does *not* guarantee that the TXOPs will actually be large enough to clear its backlog, i.e. that all packets will be transmitted within their delay bound. In fact, this depends on the allocated synchronous bandwidth and the available asynchronous bandwidth.

For both CBR and VBR TSs, the synchronous bandwidth is then computed as the minimum capacity such that the mean data rate R_i is guaranteed over $TTRT$, that is

$$H_i = tx(P) \cdot I(i) + \left\lceil \frac{R_i \cdot TTRT}{N_i} \right\rceil \cdot tx(N_i, \Gamma_i), \quad (5)$$

where $tx(P)$ is the time to transmit a QoS CF-Poll (no data) frame and $I(i)$ is an indicator function which is 1 when i is an uplink TS, and 0 otherwise. Moreover, we assume that the minimum PHY rate Γ_i is used when computing the synchronous bandwidth in (5). Therefore the mean data rate QoS requirement is guaranteed in the worst case, i.e. when the data frames of TS i are transmitted at the lowest rate advertised during the TSPEC negotiation Γ_i. When TS i operates at a rate $\gamma_i > \Gamma_i$, it will be able to transmit more bits during H_i. However, if TS i empties its queue, the unused bandwidth is returned to the QAP [2].

Furthermore, if VBR TSs were served according to their average rate, longer packet bursts would experience higher, perhaps unfeasible, delays. On the other hand, assigning them a synchronous bandwidth according to their peak rates would largely overestimate the amount of reserved bandwidth, thus wasting resources. Thus, VBR TSs are also entitled to use asynchronous bandwidth. More specifically, when the server visits a VBR TS i, it computes the sojourn time by summing the synchronous bandwidth H_i (which is fixed, and accounts for the mean rate) and the asynchronous bandwidth computed at that round (which is instead variable from one round to another). This way, VBR TSs are guaranteed a sufficient minimum rate, and they can additionally exploit the available asynchronous bandwidth to reduce the delay of long packet bursts. On the contrary, CBR TSs are not entitled to consume asynchronous bandwidth: therefore, when the QAP schedules a CBR TS, the sojourn time is given by its synchronous bandwidth alone. Finally, the node representing contention traffic is only assigned asynchronous bandwidth. Thus, contention traffic is not guaranteed to receive service on each round. However, it can exploit any bandwidth not reserved or temporarily left unused by HCCA TSs, so as to receive the highest possible level of service, without jeopardizing the negotiated QoS guarantees of HCCA TSs.

The pseudo-code of the procedure **serve**, for serving node i, is reported in Fig. 3, provided that i is in the round robin list. Specifically, the local variable `x` stores the amount of capacity that is granted to the current node during this round. In the case of a CBR TS, `x` is always set to the synchronous capacity of the current TS (line 11). In the case of a VBR TS, instead, the `TRT` variable is used to check the timeliness of the rotation in the function update (lines 1–9), according to (2). The return value of update is then added to the synchronous capacity of a VBR TS (line 15), or it is used to temporarily pause the HCCA scheduler of the QAP (line 17).

Note that the service guarantees of HCCA TSs, regarding the average rate and the upper bound on the medium access time, only hold under inequality (3). Therefore, the QAP must enforce (3) at the admission of new TSs. Specifically, the QAP enforces the fol-

[2] An alternative way of dealing with multi-rate would be to use an *observed* PHY rate in (5) instead of Γ_i, e.g. by keeping track at the QAP of the physical transmission rate employed by all QSTAs during some time window. This has been exploited by Gao *et al.* [13] with the sample scheduler of IEEE 802.11e. While this allows more TSs to be admitted, provided that some employ a physical transmission rate higher than Γ_i, resource overbooking can lead to the QoS guarantee (1) being violated at some time.

update (node i)

```
1  i.TRT -= NOW - i.last
2  i.last = NOW
3  if ( i.TRT < 0 )
4     y = 0
5     i.TRT = i.TRT - floor(i.TRT / TTRT) * TTRT
6  else
7     y = i.TRT
8     i.TRT = TTRT
9  return y
```

serve (node i)

```
10 if ( i.type == CBR )
11    x = i.H
12 else
13    y = update (i)
14    if ( i.type == VBR )
15       x = min (i.H + y, TTRT)
16    else
17       sleep (y)
18       return
19 if ( i.type == CBR || i.type == VBR )
20    if ( i.direction == DOWNLINK )
21       transmit (i, x)
22    else
23       poll (i, x)
24 return
```

Figure 3: Pseudo-code of the WTTP algorithm

lowing admission control procedure. Assume a QSTA requests the admission of TS k. The QAP first computes the new value of $TTRT$ according to (4). If D_k is smaller than $\min_{i \neq k}\{D_i\}$, the new $TTRT$ value is smaller than the old one, and all H_i ($i \neq k$) should be recomputed according to (4). Then H_k is computed according to (5) as well. If the sum of H_i (including H_k) is greater than $TTRT$, then the QAP rejects the admission of TS k, and restores the former $TTRT$ and H_i values. Thus, the admission of a new TS may require up to $\mathcal{O}(n)$ operations in the worst case (i.e. $D_k < min_{i \neq k}\{D_i\}$), where n is the number of admitted TSs. In practice, such computational complexity should not be an issue, since admission of TSs happens on a time scale much larger than that of TXOP scheduling. Furthermore, in Section V, we show that WTTP reacts to violation of the admission control test by gracefully degrading the performance of all streams.

We now analyze the complexity of WTTP. The **serve** procedure in Fig. 3 requires a constant number of operations with respect to the number of TSs. This procedure is invoked at the end of each TXOP (or contention period). As described above, the round robin list is managed so as to contain only nodes associated to TSs that either are busy (in the downlink case) or are considered to be busy (in the uplink busy), in addition to the node associated to contention traffic. In the former case, the **serve** procedure is invoked only once, whereas in the latter case the **serve** procedure is invoked once or twice, depending on the timeliness of the visit. Thus, WTTP has an $\mathcal{O}(1)$ per-packet computational complexity.

IV. Related work

In this section, we review the existing literature related to scheduling of traffic in IEEE 802.11e HCCA. A number of different approaches have been proposed in the recent past [30], which we classify in three categories: algorithms based on packet deadlines, which exploit the properties of well-known algorithms from the literature of real-time scheduling in multiprogrammed operating systems; algorithms based on queue length estimation, which periodically adapt the scheduling of TXOPs to the estimated needs of uplink TSs; and, finally, hybrid HCCA/EDCA approaches, which exploit service differentiation via EDCA based on channel load estimation.

Since some of the works described below are based on the sample scheduler defined by the IEEE 802.11e standard for informational purposes ([20] App. K) we briefly describe it first. The sample scheduler produces TDM-like schedules: each TS is periodically allocated a fixed capacity. The period is called Service Interval (SI) and it is the same for all traffic streams. It is computed as the smallest value of the MSIs (or delay bound values, if the latter is not specified) of admitted TSs, i.e. $SI = \min_i\{D_i, MSI_i\}$. The TXOP duration is then set to the smallest value that satisfies (1). The TXOP is rounded up to contain an integer number of packets of nominal SDU size, which may produce a surplus bandwidth allocation to that TS. In order to avoid head-of-line blocking, the actual TXOP value is the maximum between the value obtained by the above procedure and the time to transmit a packet of Maximum SDU Size. That is, the TXOP of TS i is computed as follows:

$$TXOP_i = tx(P) \cdot I(i) + \max\left\{\left\lceil \frac{R_i \cdot SI}{N_i} \right\rceil \cdot tx(N_i, \Gamma_i), tx(M_i, \Gamma_i)\right\}. \quad (6)$$

Since the sample scheduler produces a periodic schedule, it does not serve VBR traffic efficiently. In fact, periodically scheduling fixed TXOPs may result in wasting resources, since overprovisioning (with respect to the mean data rate) is required in order to provide feasible delay bounds. This has been shown in [11], where the authors present a simulation study that highlights the ineffectiveness of the sample scheduler in handling VBR traffic.

IV.A. Deadline-based algorithms

The problem of scheduling real-time tasks in a multiprogrammed environment has been widely studied over the last decades. The results obtained in that field have been adapted to the context of HCCA scheduling [7] [14] [24].

The Real-Time HCCA Scheduler (RTH) [7] combines the Earliest Deadline First (EDF) algorithm with the Stack Resource Policy (SRP), which are efficient policies for scheduling real-time periodic tasks in a multiprogrammed environment. When a TS requests admission to the QAP, RTH computes a periodic timetable in which TSs are granted a fixed capacity. This allows TSs with different requirements to be scheduled efficiently, and therefore RTH admits more TSs than the sample scheduler. However, RTH cannot deal with VBR traffic.

In [14] the authors propose the Scheduling based on Estimated Transmission Time (SETT-EDD). SETT-EDD requires that TSs additionally specify the *minimum service interval* (mSI), which is the minimum

time that must elapse between two consecutive service periods. Each SDU is assigned an eligibility time (i.e. time before which the SDU cannot be served) equal to its arrival time plus mSI, and a deadline equal to the arrival time plus MSI. The arrival time of packets at uplink queues is estimated by the QAP, which assumes that SDUs of nominal size arrive at QSTAs at the specified mean data rate. SDUs are then served using the well-known algorithm Earliest Due Date (EDD). SETT-EDD has been shown to perform better than the sample scheduler, especially with VBR traffic. However, it exhibits a high per-SDU computational complexity.

Finally, in [24] the authors propose a scheduling algorithm that considers both CBR and VBR traffic. The algorithm is run periodically by the QAP, which only considers QSTAs which perceive a good state channel[3]. CBR TSs are served first, according to the same algorithm used by the sample scheduler. Then, the remainder of the service period is scheduled to VBR TSs according to a deadline-based approach: a TXOP is served as late as possible without violating its deadline, which is set equal to the specified delay bound. Since the QAP does not know the arrival time of uplink packets, the worst case is assumed: the next packet is enqueued as soon as the previous one has been dispatched. Although the number of operations required for scheduling a single TXOP is low, the computation of the periodic schedule adds a considerable overhead, due to the need of sorting TXOPs based on their relative deadlines. In order not to waste channel capacity in a real environment, the QAP may actually be forced to start the computation of the schedule well in advance of the beginning of a service period, thus using partial information. On the other hand, we remark that WTTP, being a rate-based algorithm (instead of deadline-based), is $\mathcal{O}(1)$ with respect to the number of TSs, which makes it feasible for implementation at high rates with low-cost hardware.

IV.B. Algorithms based on queue length estimation

Another approach to dealing with uplink VBR traffic is to estimate the length of uplink TS queues, and tune the length of TXOPs granted to them accordingly [2] [3]. This can be done by exploiting the optional queue size piggybacking mechanism. Specifically, in [2], the information provided by the QSTAs about the state of their own queues is used to feed a control system at the QAP, which estimates the requirements of the established TSs. The resulting scheduler is the Feedback Based Dynamic Scheduler (FBDS), which schedules TXOPs whose duration depends on the requirements estimated by the control system during each period. If the sum of the allocations exceeds the service period, then all the TXOPs are decreased proportionally. On the other hand, the queue size estimation algorithm in [3], namely Fair HCF (FHCF), is based on a linear combination of the queue length samples collected over a fixed-size time window. TXOPs are then computed proportionally to the estimated queue length distributions. Before scheduling TXOPs, the QAP tunes the allocation by fairly redistributing the unused service period, if any, or by fairly decreasing the TXOPs if their sum exceeds the service period.

Both FBDS and FHCF may require a high computational overhead at the beginning of each service period, due to the queue length estimation and the TXOPs tuning procedures. On the other hand, WTTP does not perform any queue length estimation.

IV.C. Hybrid HCCA/EDCA scheduling

Finally, we review the following approaches, based on using EDCA, either alone or in combination with HCCA, to provide TSs with parameterized QoS.

In [6] the authors propose to use EDCA, which has been designed for prioritized access, to guarantee parameterized QoS. To do so, they implement a centralized admission control procedure for EDCA, which uses the same set of TSPEC parameters intended by the standard for HCCA. Medium access is subject to controlled airtime usage, which is provided by means of either tuning the maximum duration of a frame exchange sequence, or limiting the accessing frequency of each traffic category. Both methods assume that QSTAs continuously monitor the channel load, which warrants good performance when the stations transmission rates change over time. However, the inherent random-access nature of EDCA makes it impossible to derive deterministic bounds on QoS guarantees a priori. Thus, the performance is in fact assessed via simulation.

In [29] the authors propose to allow VBR flows to access the medium using both HCCA and EDCA, according to a dynamic association of traffic flows to the two medium access modes based on a current channel load estimation. However, this implies that VBR traffic requiring parameterized QoS guarantees may actually collide with (e.g.) non QoS traffic using

[3] this requires a MAC signaling protocol that notifies the QAP of slow-changing channel state information, such as the one currently under specification in the Task Group k of the IEEE 802.11 committee [21].

EDCA. In WTTP, on the other hand, although VBR TSs and contention traffic contend for the available asynchronous capacity, their packets cannot collide with each other, which limits channel wastage.

V. Performance analysis

In this section we analyze WTTP through simulation. We first define the traffic models and metrics used in the performance analysis and the settings under which the latter is carried out, and then describe the simulation scenarios. Simulation results are presented last, from Section V.C to Section V.G. More specifically, we assess the effectiveness of WTTP to serve uplink VBR traffic streams, under different load conditions. Furthermore, we analyze the resiliency of WTTP with respect to violation of the admission control limit, and its sensitivity to the selection of $TTRT$ and to irregularities in the arrival pattern of VBR traffic. Then, we assess the performance of WTTP in a mixed scenario, which involves downlink and uplink TSs, and CBR and VBR TSs, respectively. The effectiveness of the polling strategy on the capacity left for contention-based access is evaluated in both the uplink VBR traffic only and the varied scenario. Finally, we compare WTTP to FHCF, which also exploits the piggybacking of the backlog on outgoing uplink data messages, and is specifically suited for VBR traffic.

V.A. Traffic models and performance metrics

We consider two types of QoS traffic transmitted through HCCA: VoIP and videoconference (VC). Data traffic, which possesses non specific QoS requirements, is instead transmitted using DCF. Stations with data traffic operate in asymptotic conditions, i.e. they always have a frame to transmit. The packet length of data traffic is constant and equal to 1500 bytes.

We simulate a VoIP traffic stream as an ON/OFF source: during the ON (talkspurt) periods the traffic is CBR with parameters that depend on the encoding scheme; during the OFF (silence) periods no packets are generated. The encoding scheme that we employ is the G.711 [10], which produces 50 packets of 160 bytes (including IP/UDP/RTP headers) per second. Talkspurt and silence periods are distributed according to Weibull distributions[4], as reported in [5] so as to model a one-to-one conversation: $\lambda_{ON} = 1.423$ s, $k_{ON} = 0.824$, $\lambda_{OFF} = 0.899$ s, $k_{OFF} = 1.089$ (which yields $E[ON] = 1.58$ s, $E[OFF] = 0.87$ s).

We simulate VC traffic according to a pre-encoded MPEG4 trace file (LectureHQ-Reisslein, from the Internet archive of traces [32]). MPEG4 encoders produce streams of frames of variable size at fixed intervals [12]. In our simulation analysis, the frame rate is 30 fps, which corresponds to a frame interarrival time of about 33.3 ms, the average rate is about 158 kb/s, and the peak rate is about 2.7 Mb/s. In both VoIP and VC traffic models, the downlink and uplink traffic flows of a bidirectional TS are not correlated.

As far as HCCA traffic is concerned, we measure the access delay (or *delay*, for short), defined as the time elapsed from the packet reaching the MAC layer to that packet being successfully acknowledged. Furthermore, we define the *null ratio* of an uplink TS as the ratio between the number of *null* messages issued by a QSTA in response to *polls* from the QAP, and the overall number of polls issued by the QAP. The null ratio is a measure of the overhead due to the specific polling scheme that is used by the corresponding scheduling algorithm. For uplink TSs, we also measure the *polling interval*, which is defined as the time elapsed between two consecutive *polls* directed to the same TS. Finally, with regard to contention-based access, we measure the throughput of DCF traffic. Note that, since data traffic stations operate in asymptotic conditions, this is a direct measure of the amount of channel that is left available by the QAP for contention-based access.

V.B. Simulation settings

The physical layer parameters are those specified by the High Rate Direct Sequence Spread Spectrum (HR-DSSS) [19], also known as 802.11b, and are reported in Table V.B. We assume that the wireless channel is error-free. Hence, MAC level fragmentation and multi-rate support are disabled. This assumption allows us to focus specifically on the systems performance in ideal conditions. Channel errors should obviously be taken into account to have a more realistic system model, but the analysis of their influence on the scheduling algorithms is left for future study. Furthermore, we assume that all nodes can directly communicate with each other. Therefore, the hidden node problem and packet capture are not taken into consideration, and the RTS/CTS protection mechanism and automatic rate switching are disabled.

We have implemented WTTP and FHCF in the ns2 network simulator [18], using the HCCA implementation framework described in [8]. The analysis has

[4]The cumulative density function of the Weibull distribution with scale λ and shape k is $F(x;\lambda,k) = 1 - \exp\left(-(x/\lambda)^k\right)$, with $x > 0$.

Parameter	Value	Parameter	Value
SIFS	10 μs	PHY header	192 μs
PIFS	30 μs	Data rate	11 Mb/s
DIFS	50 μs	Basic rate	1 Mb/s
SlotTime	20 μs		

Table 1: MAC and physical layer parameters.

been carried out using the method of independent replications [25]. Specifically, we ran twenty independent replications of 1000 s each, with a 100 s initial warm-up period. In all the simulation runs, we estimate the 95% confidence interval for each performance measure. Confidence intervals are not drawn whenever negligible.

V.C. Uplink VBR traffic

In this scenario, we analyze the system performance with uplink VBR traffic only. The delay bound value of TSs is set to the video frames generation interval (i.e. 33 ms). The maximum number of TSs that can be served with WTTP according to the admission control test in (3) is 11[5]. More specifically, we aim at analyzing the performance of VBR traffic, in terms of delay, with an increasing number of QSTAs, each having an uplink VC TS. According to the definition of synchronous capacity in (5), TSs are provided with a reserved rate equal to the average rate of the application, which is much lower than its peak rate. Hence we expect a fraction of packets to experience a delay higher than 33 ms.

The average polling interval, reported in Fig. 4, increases when the number of QSTAs increases. This is because the average WTTP round duration increases with the number of uplink TSs, which entails higher polling intervals. More specifically, the average polling interval is higher than 33 ms because, in most cases, the duration of the TXOP granted by the QAP is enough for the QSTA to clear its backlog. Hence, the QSTA indicates to the QAP an empty TS queue. The QAP, in turn, will poll the TS after 33 ms have elapsed. This fixed contribution to the polling interval sums up to the increasing average round duration, which accounts for the increasing WTTP curve.

In regard to delay, Fig.4 also reports the 99th percentile and the average value. In both cases, the curves increase with the traffic load, and the average delays are about half the 99th percentile respective ones. This

[5]As we show further on in this section, inequality (3) can be relaxed, so as to allow WTTP to serve up to 17 TSs without significant performance degradation, in terms of the 99th percentile of the delay.

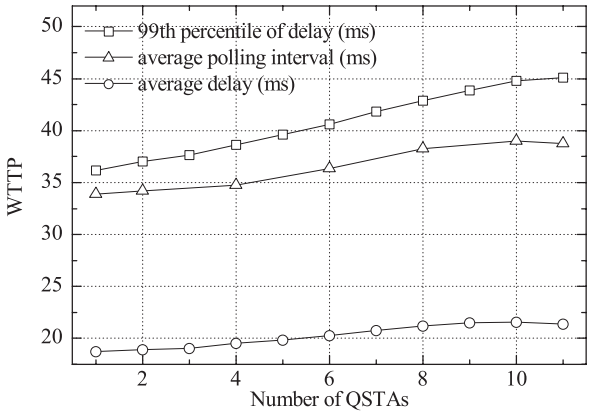

Figure 4: Delay (both 99th percentile and average) and average polling interval of an uplink VBR TS.

can be justified as follows. The 99th percentile of delay is the sum of two components: the first one, equal to the packet generation interval (i.e. 33 ms), takes into account that an uplink TS may indicate an empty queue to the QAP just before receiving a packet from the application, hence it will not be polled before a packet generation interval; the second one, on the other hand, is upper bounded by the WTTP round duration, which in turn depends on the number of admitted TSs. However, when a packet which experiences a delay higher than 33 ms is served, the next one is already available for transmission. Thus, the QAP has a chance to transmit both packets in the same TXOP, provided that enough asynchronous bandwidth was assigned by the QAP to the current TXOP. In other words, the extra bandwidth allocated to VBR TSs compensates for the high delays experienced by a fraction of packets, which is due to the polling strategy based on queue status feedback from QSTAs.

V.D. Sensitivity analysis

We now carry out a sensitivity analysis of WTTP with respect to the $TTRT$ system parameter. To do so, we repeated the simulations analyzed in the previous section (i.e. uplink VC traffic only), by setting the $TTRT$ value to $T \cdot (1 + x)$, with x equal to -10%, -5%, $+5\%$, $+10\%$, and T equal to the value computed according to (4), that is 16,67 ms. In doing this, we also evaluate the resilience to violations of the admission control limit derived from (3), in order to assess how gracefully the QoS of admitted TSs degrades in case of overload. To do so, in the experiments we allow the number of QSTAs to grow up to 17, which is a 50% increase with the respect to the limit computed for the same scenario in Section V.C (i.e. 11 QSTAs). Finally, we modified the VC traffic model described in

Section V.A by adding a random time interval to the packet interarrival times.

The average polling interval, reported in Fig.5, exhibits an irregular behavior, which is due to the concurrence of the following two mechanisms. On the one hand, part of the average polling interval is due to the average round duration, which increases when the number of QSTAs increases. On the other hand, as soon as an uplink TS clears its backlog, that TS has to wait for at least 33 ms, that is the packet generation interval, which does not depend on the number of QSTAs. When the system is underloaded, the duration of a round is small, thus it is likely that a TS has to wait for the expiration of the packet generation interval timer before being served again. This corresponds to the first increasing part of the curves in Fig. 5. As the number of QSTAs increases, it becomes more and more probable that the packet generation interval timer expires before a full round has elapsed. In other words, the probability that an uplink TS misses a service turn decreases when the number of QSTAs increases, thus reducing the average polling interval as well. This corresponds to the decreasing part of the curves in Fig. 5. Finally, when the systems gets overloaded, the effect of the packet generation interval timer becomes negligible with respect to the average round duration. Hence, the average polling duration increases again with the number of QSTAs, until it eventually shows a steep increase (not shown in Fig. 5), which entails unfeasible VC delays. As can be seen, in most cases the variations on the $TTRT$ value lead to proportional variations on the average polling interval curves. However, this is not true in the cases of -5% and -10% when the system is underloaded, where the curves lie significantly above the 0%, $+5\%$, and $+10\%$ curves. This is because with small $TTRT$ values, when the system is extremely underloaded, the packet generation interval is higher than twice the average round duration. Thus, in most rounds the uplink TSs miss *two* service turns, instead of one, as in the cases with larger $TTRT$, which leads to substantially higher average polling intervals.

In Fig. 6 we show the 99th percentile of the delay. As long as the system is underloaded, the delay curves increase with almost the same rate as the average polling interval ones. Also, when the system is extremely underloaded, the -5% and -10% curves experience the same anomaly discussed above in regard to the average polling interval. However, when the system load increases, the 99th percentile of delay increases with a smaller rate than the average polling interval. This is because the reduction

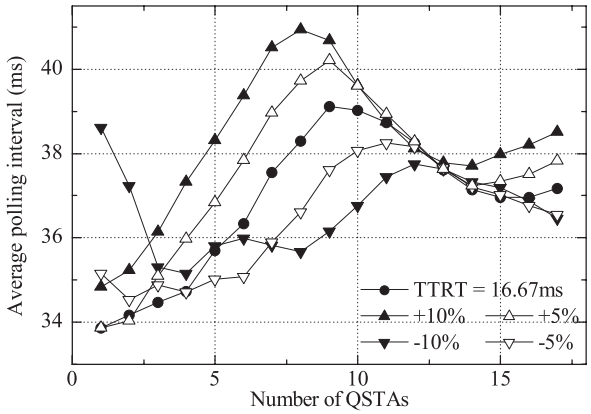

Figure 5: Average polling interval of an uplink VBR TS for various $TTRT$ values.

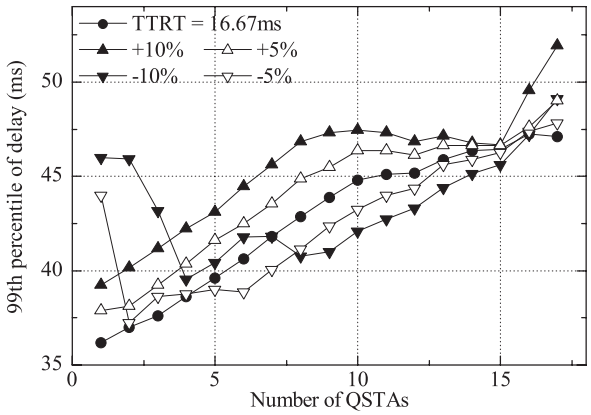

Figure 6: 99th percentile of the delay of an uplink VBR TS for various $TTRT$ values.

in the average polling interval cannot overcompensate the reduction of asynchronous bandwidth available to VBR TSs, due to the increasing load. The net effect is that there is a region where the 99th percentile of the delay does not depend on the number of QSTAs. Note that the longer the $TTRT$ duration is, the larger the size of this plateau is (e.g., in the case of -10% there is no such a region). Finally, as in the average polling interval case, the curves raise again when the system gets overloaded.

As shown in the analysis above, the performance of admitted TSs degrades gracefully (if at all) when the admission control limit is exceeded so as to serve up to 50% more TSs. In fact, the size of MPEG video frames is highly variable, hence the synchronous bandwidth computation in (5), which assumes a constant size of all packets equal to the nominal SDU size, is substantially overprovisioned. This is due to the ceiling operator in (5), and to the fact that MAC and physical layer overheads are proportional to the number of packets, not to the packet size or rate.

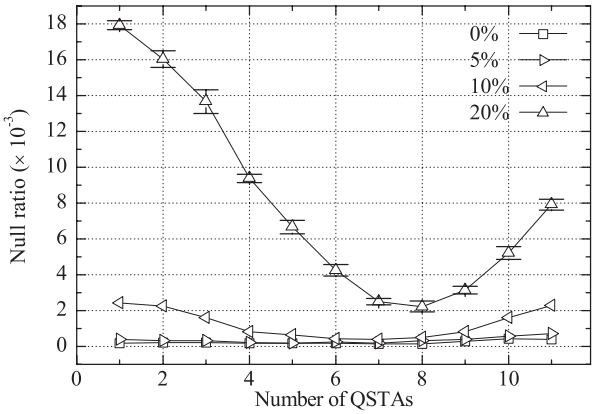

Figure 7: Null ratio when packet arrivals at uplink VBR TSs are not strictly periodic.

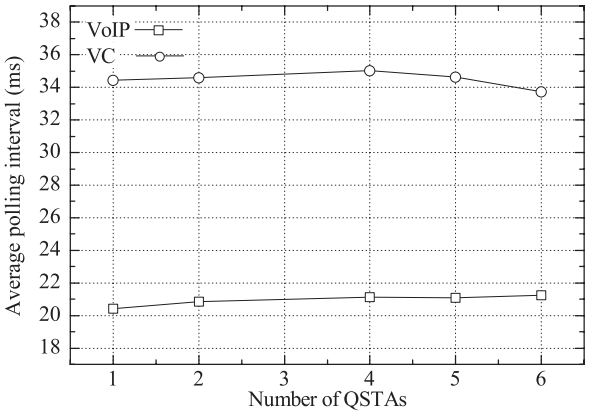

Figure 8: Average polling interval of uplink TSs when bi-directional CBR and VBR traffic is transmitted.

Furthermore, we repeated the set of simulations in Section V.C, this time modifying the frame generation pattern of VC applications. As discussed in Section III, the frame generation interval should be constant and equal to 33 ms. However, operating system mechanisms (e.g., task scheduling) often add a random term to the above interval, making the nominal information unreliable. Thus, we assess the impact on the performance of imperfectly periodic packet generation by adding a random value to the nominal packet generation interval. The random value is sampled from a uniform distribution between $-x$ and x, with x equal to 5%, 10%, and 20% the packet generation interval, respectively. The observed effect of imperfect packet generation is negligible. In fact, since the round duration is variable, WTTP inherently compensates arrival times irregularities, up to a large extent. As a matter of fact, the only metric slightly affected by the imperfect packet generation is the null ratio, reported in Fig. 7. The latter increases with the width of the uniform distribution interval. Nonetheless, the absolute value is small enough not to affect the performance significantly, in terms of both the delay of VC TSs and the throughput of contention traffic (the observed difference for those measures, not shown here, is less than 1%).

V.E. Bi-directional CBR and VBR traffic

In this scenario we compare the performance of WTTP with mixed CBR and VBR traffic. To do so, we set up an increasing number of QSTAs, from 1 to 6, each having a bi-directional VoIP TS and a bi-directional VC TS. The delay bound value of VoIP TSs is set to 20 ms, and that of VC TSs to 33 ms.

Fig.8 shows the average polling interval of VoIP and VC TSs. The curves are almost constant and approximately equal to the packet generation interval of VoIP and VC traffic, respectively. This is in contrast with the results obtained with uplink VBR traffic only, where the average polling interval increases with the number of QSTAs. This is due to the presence of CBR traffic, which keeps the average round duration much smaller (in fact, $TTRT = 10$ ms) than the packet generation interval of VC applications (i.e. 33 ms).

Fig. 9 shows the 99th percentile of the delay of VoIP and VC traffic, respectively. Let us consider VoIP traffic first. The downlink curve lies significantly below the uplink one. On the one hand, each node associated to a downlink TS is added to the round robin list as soon as the TSs queue becomes backlogged. Therefore, the service delay is only due to the time that it takes for the server to cycle through the backlogged nodes. On the other hand, the minimum polling interval of VoIP TSs is equal to the VoIP packet generation interval, i.e. 20 ms, which sums up to the round duration delay. The same line of reasoning holds for VC traffic as well. In fact, in downlink, the VC curves are almost overlapping with the VoIP ones. On the other hand, in uplink, the 99th delay percentile is always larger than the VC packet generation interval, i.e. 33 ms.

V.F. Effectiveness of the polling strategy

We now evaluate the effectiveness of exploiting cross-layer information about uplink TSs by measuring the capacity left for contention-based access. To this purpose, we compare WTTP with a modified version, in which the queues corresponding to uplink TSs are never removed from the round robin list, i.e. they are assumed to be always backlogged. Thus, the QAP does not exploit neither the piggybacking of the TS backlog on data messages, nor cross-layer infor-

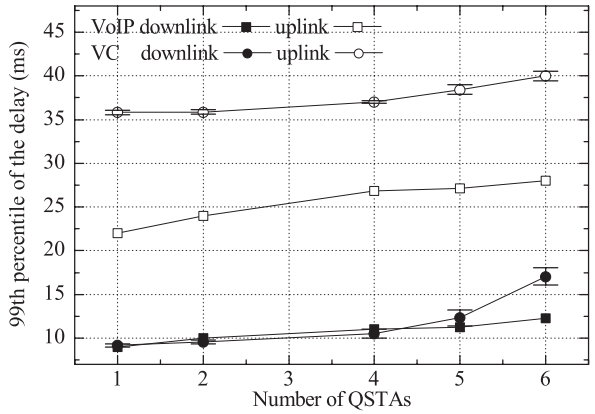

Figure 9: 99th percentile of the delay of uplink and downlink TSs when bi-directional CBR and VBR traffic is transmitted.

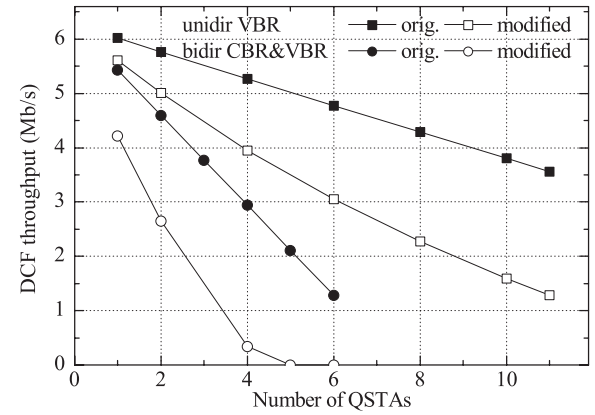

Figure 10: DCF traffic throughput when a) uplink VBR traffic, and b) bidirectional CBR and VBR traffic are transmitted.

mation about the nominal packet generation interval. We simulated the scenarios analyzed in the previous sections, i.e. both uplink VBR traffic only and bidirectional CBR and VBR traffic, using both WTTP and the above mentioned modified version.

Quite clearly, the MAC overhead involved in unnecessarily polling idle uplink TSs is large. In fact, in Fig. 10 we show the throughput of DCF traffic under both WTTP and the modified version. As expected, in both cases, the throughput decreases with the number of QSTAs, since the channel is shared among an increasing number of TSs served with HCCA. However, WTTP largely outperforms the modified version, in terms of the capacity left for contention access. Thus, the polling strategy adopted for uplink TSs in WTTP allows a large amount of capacity to be saved, thus improving the performance for DCF traffic.

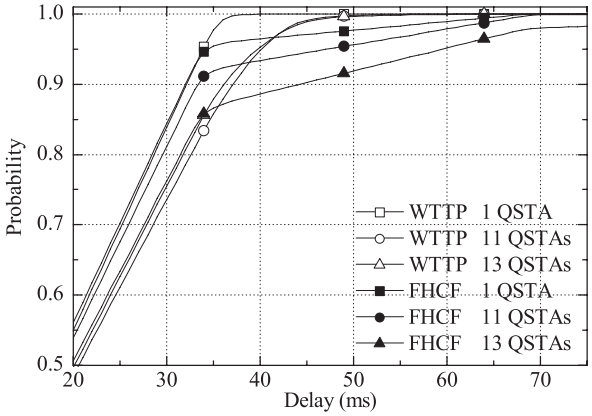

Figure 11: CDF of the delay of an uplink VBR TS under WTTP and FHCF.

V.G. Comparison with FHCF

To conclude our analysis, we compare the results obtained with WTTP and FHCF, in the case of uplink VBR traffic only.

We analyze the Cumulative Distribution Function (CDF) of delay, reported in Fig. 11, in the following cases: system highly underloaded (i.e. 1 QSTA only); 11 QSTAs admitted, which is the admission control limit of WTTP according to (3); 13 QSTAs admitted, which is the admission control limit of FHCF. In all load conditions, the distribution tails with FHCF are heavier than those with WTTP. This can be justified as follows. TXOPs are scheduled by FHCF based on the queue size estimation over the previous service periods. Assume that a burst of packets is enqueued at a TS, due to the generation of a large video frame, during round i. In this case, FHCF will not be able to schedule an adequate TXOP until the next round $i + 1$, which leads to large video frames being served with a delay greater than a service period (i.e. 33 ms), although the system is underloaded (e.g. 1 QSTA only). As the load increases, the delay of large video frames further increases due to the corresponding reduction of available capacity per traffic stream. WTTP, in particular, exhibits lighter delay tails also with 11 and 13 QSTAs, when the average polling interval of FHCF (i.e., about 33 ms) is lower than that with WTTP (shown in Fig. 4). This is because WTTP polls uplink TSs with larger TXOPs, although at a lower rate.

Finally, in Fig. 12 we show the throughput of contention traffic, using DCF. The WTTP curve lies slightly above the FHCF curve, because the former does not follow a strictly periodic polling scheme, and hence it sometimes serves two video frames within the same TXOP, thus saving one *poll*. The higher the

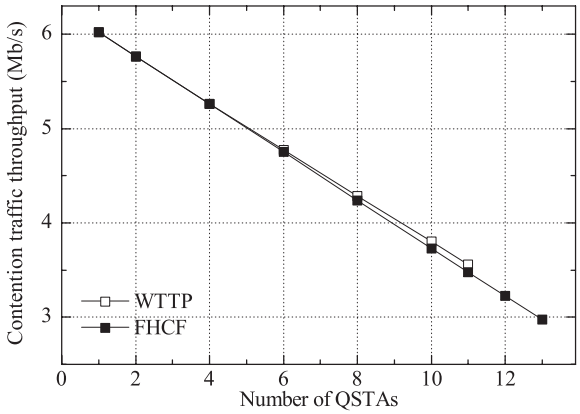

Figure 12: DCF traffic throughput under WTTP and FHCF when uplink VBR traffic is transmitted.

number of QSTAs (i.e. the average round duration), the higher the probability of such an event.

VI. Conclusions

In this paper we presented a centralized scheduling algorithm, namely WTTP, to serve multimedia traffic in IEEE 802.11e networks using HCCA. WTTP, which has a $\mathcal{O}(1)$ per-packet computational complexity with respect to the number of TSs, is able to provide all admitted traffic streams, either CBR or VBR, with rate-based guarantees. WTTP exploits cross-layer information, namely the application packet generation pattern (either CBR or VBR) and application packet generation interval, to devise an effective bandwidth sharing and polling strategy. In fact, VBR TSs are provided with an extra amount of bandwidth so as to keep packet delays low also when their current rates temporarily exceed the mean reserved rates. Furthermore, the QAP refrains from polling uplink TSs which notified an empty buffer before a full packet generation interval has elapsed.

We evaluated the performance of WTTP via an extensive simulation analysis. The results showed that WTTP is able to serve VBR TSs with a 99th percentile of the delay which is almost constant with respect to the number of admitted traffic streams. Moreover, with mixed bi-directional CBR and VBR traffic, the 99th percentile of delay with both traffic types was shown to be stable under different load conditions, and slightly higher than the respective packet generation intervals. Finally, we analyzed the resilience of WTTP under the following conditions: random jitter in the packet interval generation of uplink applications; variations of the TTRT; violation of the admission control limit. WTTP has been shown to be resilient to the above effects, in terms of the performance of both HCCA and contention-based access functions.

VII. References

References

[1] T. Anjali, C. Scoglio, G. Uhl. A new scheme for traffic estimation and resource allocation for bandwidth brokers, Computer Networks 41 (2003) 761–777.

[2] A. Annese, G. Boggia, P. Camarda, L. A. Grieco, and L. Mascolo. Providing delay guarantees in IEEE 802.11e networks, proc. IEEE VTC 2004, Milan, Italy, 17–19 May, 2004, vol. 4, pp. 2234–2238.

[3] P. Ansel, Q. Ni, and T. Turletti. FHCF: A Simple and Efficient Scheduling Scheme for IEEE 802.11e Wireless LAN, Mobile Networks and Applications (Springer), vol. 11, no. 3, June 2006, pp. 391–403.

[4] D. Bertsekas and R. Gallager. Data Networks, Prentice-Hall, 1987.

[5] C.-N. Chuah and R. H. Katz. Characterizing packet audio streams from Internet multimedia applications, proc. IEEE ICC 2002, New York, USA, April 28–May 2, 2002, vol. 2, pp. 1199–2203.

[6] Chun-Ting Chou, S. N. Shankar, K. G. Shin. Achieving per-stream QoS with distributed airtime allocation and admission control in IEEE 802.11e wireless LANs, proc. IEEE INFOCOM 2005, Miami, USA, March 1317, 2005, vol. 3, pp. 1584–1595.

[7] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea. Design and Performance Analysis of the Real-Time HCCA Scheduler for IEEE 802.11e WLANs, to appear in Computer Networks (Elsevier), February 2007.

[8] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea. A software architecture for simulating IEEE 802.11e HCCA, proc. IPS MoMe 2005, Warsaw, Poland, March 14–15, 2005.

[9] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea. WTTP: A Scheduling Algorithm for Supporting QoS in IEEE 802.11e, proc. European Wireless 2005 (invited paper), Nicosia, Cyprus, April 10–13, 2005, pp. 231–239.

[10] Cisco Press. Traffic Analysis for Voice over IP, November 2001.

[11] J. Cowling and S. Selvakennedy. A detailed investigation of the IEEE 802.11e HCF Sample scheduler for VBR traffic, proc. IEEE ICCCN 2004, Chicago, USA, October 11–13, 2004.

[12] Frank H.P. Fitzek and Martin Reisslein. MPEG4 and H.263 Video Traces for Network Performance Evaluation, IEEE Network, vol. 15, no. 6, November 2001, pp. 40–54.

[13] D. Gao, J. Cai, L. Zhang. Physical rate based admission control for HCCA in IEEE 802.11e WLANs, proc. AINA 2005, Tamkang University, Taiwan, March 28–30, 2005, pp. 479–483.

[14] A. Grilo, M. Macedo, and M. Nunes. A scheduling algorithm for QoS support in IEEE 802.11e networks, IEEE Wireless Communications, vol. 10, no. 3, June 2003, pp. 36–43.

[15] R. M. Grow. A timed-token protocol for local area networks, proc. Electro/82, Token Access Protocols, Electronic Conventions, Inc., May 1982.

[16] I. Haratcherev, J. Taal, K. Langendoen, R. Lagendijk, H. Sips. Automatic IEEE 802.11 rate control for streaming applications, Wireless Communications and Mobile Computing (Wiley), vol. 5, no. 4, 2005, pp. 421–437.

[17] G. Hiertz, L. Stibor, J. Habetha, E. Weiss, S. Mangold. Throughput and Delay Performance of IEEE 802.11e Wireless LAN with Block Acknowledgments, proc. European Wireless 2005, Nicosia, Cyprus, April 10–13, 2005, pp. 246–252.

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

[19] IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications, August1999.

[20] IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications. Medium Access Control (MAC) Quality of Service (QoS) Enhancements, December 2005.

[21] IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications. Medium Access Control (MAC) Radio Resource Measurement, D4.0, March 2006.

[22] M.J. Johnson. Proof that timing requirements of the FDDI token ring protocol are satisfied, IEEE Trans. on Communications, vol. 35, no. 6, June 1987, pp. 620–625.

[23] A. Köpsel, J.-P. Ebert, and A. Wolisz. A performance comparison of point and distributed coordination function of an IEEE 802.11 WLAN in the presence of real-time requirements, proc. MoMuC 2000, Tokio, Japan, October 2000.

[24] T. Korakis and L. Tassiulas. Providing quality of service guarantees in wireless LANs compliant with 802.11e, Computer Networks (Elsevier), vol. 47, no. 2, February 2005, pp. 239–255.

[25] A. M. Law, W. D. Kelton. Simulation modeling and analysis. Third edition, McGrawHill, 2000.

[26] L. Lenzini, E. Mingozzi, and G. Stea. Design and performance analysis of the generalized timed token service discipline, IEEE Trans. on Computers, vol. 53, July 2004, pp. 879–891.

[27] A. Lindgren, A. Almiquist, and O. Schelen. Quality of Service Schemes for IEEE 802.11 Wireless LANs: An Evaluation, Journal on Mobile Networks and Applications, vol. 8, no. 3, June 2003, pp. 223–235.

[28] Q. Ni. Performance Analysis and Enhancements for IEEE 802.11e Wireless Networks, IEEE Network Magazine, vol. 19, no. 4, July 2005, pp. 21–27.

[29] N. Ramos, D. Panigrahi, S. Dey. Dynamic adaptation policies to improve Quality of Service of multimedia applications in WLAN networks, proc. Workshop on Broadband Wireless Networks, San Jose, USA, October 25–29, 2004.

[30] N. Ramos, D. Panigrahi, Sujit Dey. Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions, IEEE Network Magazine, vol. 19, no. 4, July 2005, pp. 14–20.

[31] S. Shankar, S. Choi. QoS Signaling for parameterized traffic in IEEE 802.11e Wireless LANs, proc. AISA 2002, Seoul, Korea, August 12, 2002, pp. 67–84.

[32] http://trace.eas.asu.edu/, continuously updated.

[33] P. Wang, H. Jiang, W. Zhuang. IEEE 802.11e enhancement for voice service, IEEE Wireless Communications, vol. 13, no. 1, February 2006, pp. 30–35.