

Performance Evaluation of Real-Time Schedulers for HCCA Function in IEEE 802.11e Wireless Networks

G. Cecchetti
g.cecchetti@sssup.it

A. L. Ruscelli
a.ruscelli@sssup.it

Retis Lab.
Scuola Superiore S. Anna
Pisa, Italy

ABSTRACT

IEEE 802.11 standard for wireless networks recently has been enhanced with IEEE 802.11e amendment to this protocol which introduces Quality of Service support. It provides differentiation mechanisms at the Medium Access Control layer, using two additional access functions: the Enhanced Distributed Channel Access (EDCA) function and the HCF Controlled Channel Access (HCCA) function.

The EDCA function is based on a distributed control and enables prioritized channel access while the latter requires centralized scheduling and allows the applications to negotiate parameterized service guarantees. Only HCCA mechanism is suitable for serving traffic streams with real-time requirements such as multimedia applications and Voice Over IP. The IEEE 802.11e standard does not specify a mandatory HCCA scheduling algorithm, while it offers a reference scheduler.

In this paper we analyze four HCCA schedulers which are alternative to the reference one and which offer real-time guarantees. A performance evaluation through simulation is conducted to show the main differences between the considered schedulers, including the reference one.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless Communication*

General Terms

Algorithms, Performance

Keywords

Quality of Service, Real-Time guarantees, Scheduling Algorithms, Performance evaluation, Wireless LAN.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Q2SWinet'08, October 27–28, 2008, Vancouver, BC, Canada.
Copyright 2008 ACM 978-1-60558-237-5/08/10 ...\$5.00.

1. INTRODUCTION

The multimedia applications like VoIP, videoconference, and multimedia streaming, need Quality of Service (QoS) support which can meet their different requirements. In fact, the presence of traffics with Constant Bit Rate (CBR) and Variable Bit Rate (VBR) involves to provide differentiated QoS levels. Moreover the time-varying characteristics of the wireless channel [7] do not allow to assure hard constraints. However, for the multimedia applications the support of soft real-time constraints is sufficient since missing deadlines implies only a degradation of the provided QoS.

Even if IEEE 802.11 [19] is the recognized standard for Wireless Local Area Networks (WLAN), it was designed to provide only best effort service. Recently it has been enhanced with the amendment IEEE 802.11e in order to include the QoS guarantees. The recent standard has introduced a differentiation mechanism at the *Medium Access Control* (MAC) layer, using two additional access functions: the *Enhanced Distributed Channel Access* (EDCA) function and the *Hybrid Controlled Channel Access* (HCCA) function. The EDCA function is based on a distributed control and enables prioritized channel access while the latter requires centralized scheduling and allows the applications to negotiate parameterized service guarantees. In particular, only HCCA mechanism is suitable for respecting real-time constraints. However the IEEE 802.11e standard does not specify a mandatory HCCA scheduling algorithm, while it offers a reference scheduler that is compatible with the use of link adaptation and that respects a minimum set of performance requirements. In particular it assigns a fixed transmission opportunity to all the managed flows, every a constant common period. Several studies have evaluated the new standard through analytical techniques and simulation [12, 16] and they had demonstrated that it improves the QoS support more specifically for CBR traffic whereas it displays its limits for VBR traffic, since it assigns fixed transmission parameters and it is not suitable for following the variability of this typology of traffic. Based on these considerations, several works [4, 13, 15] have suggested alternative scheduling algorithms to the reference one, in order to improve its QoS provisioning for VBR traffic, but they do not consider specifically the real-time constraints.

In this paper we present an evaluation of main schedulers tailored for the real-time guarantees support over IEEE 802.11e HCCA networks. The analysis considers the performances in terms of the admission control, of the resource utilization efficiency, and finally of the access delay.

The rest of the paper is organized as follows: in section 2 we describe the IEEE 802.11e standard, in section 3 we illustrate the considered scheduling algorithms including the reference scheduler and in Section 4 we evaluate their performances through simulation. Finally, the conclusions are drawn in Section 5.

2. THE IEEE 802.11e HCCA

The IEEE 802.11e *Hybrid Coordination Function* multiplexes between two medium access modes: the EDCA, which provides prioritized QoS to different traffic levels through the introduction of *Access Categories* (ACs), and the centralized HCCA. To ensure compatibility with legacy devices, the standard allows the coexistence of *Distributed Coordinator Function* (DCF) and *Point Coordination Function* (PCF) with EDCA and HCCA. Because this paper is about schedulers for the latter mechanism we describe more in detail the HCCA function.

HCCA provides a QoS-aware *Hybrid Coordinator* (HC), usually located at the *QoS Access Point* (QAP) in infrastructure WLANs, which provides polled access to the wireless medium. In order to be included in the polling list of the HC, a *QoS Station* (QSTA) must send a QoS reservation request to the QAP for each *Traffic Stream* (TS), using the special QoS management frame, *Add traffic Stream*, which contains the *Traffic Specification* (TSPEC). TSPEC includes parameters as *mean data rate* (R_i), *nominal Service Data Unit size* (L_i), *minimum PHY rate* (Γ_i), *Delay bound* (D_i) and *Maximum Service Interval* (MSI).

HC aggregates QSTA TSPECs to determine the values of transmission parameters: *Service Interval* (SI) and *Transmission Opportunity* (TXOP). SI is the time duration between successive polls for the node and it is a submultiple of the 802.11e beacon interval duration. TXOP is the transmission duration of each node based on the mean application data rates of its requested flows. Before the calculation of the latter parameters, QAP has to verify if the admission of each TS does not compromise the service guarantees of the already admitted TSs and, if the specified TS is accepted, QAP sends a positive acknowledgement which contains also the service start time that indicates the time from when the QSTA is allowed to transmit frames relative to considered TS.

When there are admitted QSTAs, the QAP listens to the medium and, if it is idle for a *PCF InterFrame Space*, HC gains control of the channel and, within the *Controlled Access Phase* (CAP), it polls a single QSTA at turn, according to the polling list, generated by the scheduler. It is necessary to distinguish between downlink TXOP, during which the QAP sends burst QoS Data to QSTA and uplink TXOP, that starts when the polled QSTA takes the medium control. If the polled QSTA does not have packets for the considered TS (TS of the polled QSTA is not backlogged) or if the head-of-line packet does not fit into the remaining TXOP duration, the QSTA sends a QoS CF-Null frame to the QAP.

3. REAL-TIME SCHEDULERS FOR IEEE 802.11e

In this section we describe the schedulers we are going to analyze in this paper.

3.1 Reference scheduler

The standard IEEE 802.11e proposes a sample scheduler as a reference in the design of a scheduling policy. It computes SI as the sub-interval of the beacon interval value smaller than the minimum of the Maximum SI, MSI_i , and the TXOP as the time to transmit the max number N_i of max size MAC Service Data Unit ($MSDU$) at the mean data rate R_i . N_i and $TXOP_i$ are computed as:

$$N_i = \left\lceil \frac{SI \cdot R_i}{L_i} \right\rceil \quad TXOP_i = \max\left(\frac{N_i \cdot L_i}{R_i}, \frac{M_i}{R_i}\right) + O$$

where M_i is the maximum allowable size of MSDU, i.e., 2304 bytes, and O is the Overheads in time units.

Both SI and TXOP are fixed values, based on "worst case" conditions and they can be recomputed only if a new TS arrives. This means that all the TSs, which have different characterizations, are polled with the same period and are served with the same computation time.

The admission control test for this scheduler is:

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=0}^k \frac{TXOP_i}{SI} \leq \frac{T - T_{CP}}{T}$$

where k is the number of existing streams, $k+1$ is used as index for the newly arriving stream, T indicates the beacon interval and T_{CP} is the time used for EDCA traffic. The use of such values causes a more stringent condition and a not-optimal resource management.

3.2 Fair HCF

Fair HCF (FHCF) [4] aims to improve the fairness both of CBR and VBR flows and the delay performances assigning variable TXOPs by means of the estimation of the uplink TSs queues length. The mathematical model proposed for the queues shows the relationship between polling interval and queuing delays and it is used to estimate the global packet delay. More specifically, it distinguishes between the packet queuing delay Q , due to the delay in the queue, influenced by the variations in packet size and data rate, by the waiting time delay W , defined as the interval between the packet arrival time and the QSTA polling time. In the estimation of queuing delay experienced by a packet, the authors distinguish the case where the queues are empty at the end of the TXOP, especially for CBR traffic, and the case where the queues are not empty, that is more realistic for real wireless networks. In the first situation, there is a further distinction if the packet arrives before the polling time of its QSTA (the packet has to wait for the polling QSTA), during (the packet has to wait for the transmission of previous queued packets) or after the QSTA polling time (the packet has to wait for the next SI). The delay for packet P is computed as

$$d_i(t) = \left(\sum_{j=1}^{i-1} T_j - t + \frac{q_i M_i}{R_{eff}} + \frac{\rho_i}{R_{eff}} \right) \chi_{[0, \sum_{j=1}^i T_j]}(t) + \left(SI - t + \sum_{j=1}^{i-1} T_j + \frac{\rho_i}{R_{eff}} \left(t - \sum_{j=1}^{i-1} T_j \right) \right) \chi_{[\sum_{j=1}^i T_j, SI]}(t)$$

where T_j is the allocated TXOP for TS_j , q_i is the number of packets in the queue, M_i is MSDU size, R_{eff} is the effective data throughput, ρ is the application data rate, χ is a factor equal to one if $t \in [i, j]$, otherwise it is equal to zero. In

this ideal case the delay is bounded by the SI. Instead, if the queues are not empty at the TXOP end, the packets can be queued later than the next SI because the delays due to the other queues are cumulative. In the best case the packet will be transmitted in the next SI, but in general the packet delay can become really unpredictable if there is traffic highly variable. This delay is expressed by:

$$D_i = \max_t \{d_i(t)\} = SI - T_i + \frac{q_i^e M_i}{\rho_i}$$

where q_i^e is the queue length at the end of the TXOP T_i . This equation suggests two different ways to control the maximum delay. One possible solution is that the QAP increases T_i , which allows to reduce the queued packets at the end of the TXOP. That can be obtained also reducing SI but that implies an increasing number of polling and an increasing overhead. FHCF adopts the alternative method to reduce the delay through the control of the queue length before the polling time. FHCF is composed of two schedulers, the QAP scheduler and the node scheduler. The QAP scheduler estimates the varying queue length for each QSTA at the beginning of the next SI, q_i^{est} , and compares this value with the ideal queue length, q_i^{ideal} . In particular,

$$q_i^{est} = \frac{\rho_i(SI - t_i^e)}{M_i} + q_i^e$$

where t_i^e is the corresponding time at the TXOP end to q_i^e which is recorded by the QAP in the QoS control header field. Because sending rate and packet size can change, this estimation can not be accurate, so the QAP scheduler corrects the q_i^e computation using the expected value $E[|\Delta_i(n)|]$, calculated by means of a window of previous queue length estimations. The number of previous estimations is chosen as a tradeoff between complexity and accuracy. After the mentioned comparison, QAP computes the additional required time t_i^{est} (positive or negative) for each TS of each QSTA and reallocates the corresponding TXOP duration according to the number of additional packets in the queue, DN_i^{est} . Moreover, it takes into account the actual available time after the allocation of all the TXOPs in one SI in the ideal case and, if that is not sufficient, it decreases fairly all the assigned times by a percentage of t_i^e . That allows to manage the traffic variations.

The node scheduler, located in each QSTA, just after the CF-Poll reception, can redistribute the additional time to the TXOP, which is always allocated to a whole QSTA, among its TSs. It executes the same computation than the QAP scheduler but with more accuracy since each QSTA knows exactly the TSs queues size at the beginning of the polling and it is able to estimate its queue length at the end of TXOP and the requested additional time for TS. According to its allocated TXOP, it evaluates the remaining time T' that can be re-allocated considering the number of packets N_i to transmit in the TS_i and the time required to transmit a packet, computed according to its QoS requirements:

$$T' = T - \sum_{i=1}^p N_i \cdot \left(\frac{M_i}{R_{eff}} + 2SIFS + ACK \right).$$

3.3 Wireless Timed Token Protocol

Wireless Timed Token Protocol (WTTTP) [11] is based on the *Timed Token Protocol* (TTP) [17], which is a token pass-

ing scheme employed as the MAC protocol for ring-based networks.

The token is used to manage a circular list of nodes in a round-robin manner. Each node refers downlink and uplink TSs. One special node represents contention traffic using contention-based schemas as EDCA and DCF. The scheduler visits each node of the list for a time called *sojourn time* and either schedules TXOP or refrains from generating CAP according to the TSs for that node (if the node represents contention traffic). The nodes are inserted in the round robin list only if they actually have traffic to be served:

- *downlink TSs* are added and removed whenever they become backlogged or idle;
- *uplink TSs* are added and removed if each QSTA *piggybacks* the backlog of its TSs on outgoing data messages (standard 802.11e feature). However such nodes are put back in the round robin list at least each minimum SI.

The token circulation is ruled by the *Target Token Revolution Time* (TTRT), which is a protocol parameter selected as reference round duration: its value is computed by the QAP according to the TSPEC values negotiated by the QSTA during the admission control phase. The authors set the value of TTRT to half the smallest delay bound, since the round robin duration is bounded by $2 \cdot TTRT$: $TTRT = \frac{1}{2} \min_i \{D_i\}$.

The sojourn time of the QAP scheduler at a node is given by either one or both the following two components:

- *synchronous bandwidth* which is a fixed time H_i computed as a percentage of TTRT, and it is used to transmit frames with HCCA rate based guarantees, and
- *asynchronous bandwidth* which is a variable and not reserved portion of TTRT, and it is used to transmit the remaining not guaranteed traffic.

In other words, each node i has a $H_i \geq 0$ and a *Token Rotation Timer* (TRT_i), initially set to TTRT, which 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:

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

In this way the asynchronous flows are not reserved any capacity: they may transmit frames only if the token arrives earlier than expected, i.e., before a TTRT time has elapsed from the last token visit. The early arrival of the token usually occurs when the synchronous flows consumed less than the reserved capacity during the previous token revolution. Because $TTRT$ is the average inter-service time for a node, and $2 \cdot TTRT$ is an upper bound of such time, the following equation is the admission control test for this algorithm:

$$\sum_{node_i} H_i + \tau \leq TTRT.$$

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 is an overhead due to the time required to regain the control of the medium and to start a new CAP after a DCF/EDCA phase.

3.4 Real-Time HCCA

The Real-Time HCCA (RTH) algorithm [10] is designed to provide real-time support in HCCA assuring to traffic flows a fixed amount of capacity during a fixed period. Its activity is split into *offline* activity at flow lifetime timescale which performs the more complex activity and *online* activity at the frame transmission timescale. In particular, the online activity executes the enforcement policy with a periodic scheduler based on *Earliest Deadline First* (EDF) algorithm [20] plus *Stack Resource Policy* (SRP) [6] in order to take into account the non-preemptability of frame transmissions that are considered critical sections. The scheduling parameters are computed by the offline activity. In this manner it consists only of reading the next entry composed by $[i, t_i, \text{TXOP}_i]$, i.e., the index of the next QSTA which can access the medium, the polling time t_i and the duration of its transmission. The computational complexity of this part of scheduler is $O(1)$ in term of the number of QSTA. Conversely, the offline activity attends to the admission control and to the timetable computation. The latter is based on the parameters obtained through the admission control phase itself: it takes into account the requirements of the QSTAs and the guarantees already assured by means of previous admissions.

In particular, each time a QSTA asks a new TS to be transmitted, the *MAC sublayer Management Entity*, performs the admission control and, in the case of positive result, it translates a set of the TSPEC mandatory parameters $[R_i, N_i, D_i, \Gamma_i]$ into that used in the enforcement procedure by the MAC and notifies the QSTA of successful operation. Each TS is characterized through two parameters, the capacity C_i and the period T_i , which are computed from TSPEC parameters. C_i is computed as the time needed to send at rate R_i how many Nominal SDU can be held during a period T_i , and it can be expressed as follows: $C_i = \lceil (R_i \cdot T_i) / N_i \rceil \cdot t_{N_i}$, where t_{N_i} is the Nominal transmission time. The period T_i is set equal to the delay bound D_i , if $D_i < \lceil (N_i / R_i) \rceil$, otherwise $T_i = \lfloor (R_i / N_i) \cdot D_i \rfloor \cdot N_i / R_i$.

In order to admit a TS the QAP verifies the schedulability test introduced in the multi-programmed environment by SRP which includes in the test the blocking time due to the non-preemptability of TS transmission. In the case of HCCA, the minimum critical section b_i for a TS_i is equal to the nominal size SDU transmission time t_{N_i} including the poll time for uplink TSs which is t_{P_i} , i.e., $b_i = t_{N_i} + t_{P_i}$. When TS_i is in a critical section, and is thus scheduled instead of the highest priority TS_j , TS_i is said to block TS_j . So, the blocking time for a TS_i is the maximum critical section durations of TSs with a period longer than TS_i , i.e., $B_i = \max_{j>i} \{b_j\}$. The schedulability analysis produces the following sufficient condition to determine the set of n schedulable TSs and which has an $O(n)$ computational complexity:

$$\frac{B_i}{T_i} + \sum_{j \leq 1} \frac{C_j + \pi_j + t_{P_j}}{T_j} \leq 1 \quad \forall i : 1 \leq i \leq n.$$

Moreover, larger versions of B_i and b_i , \bar{B}_i and \bar{b}_i , which are the maximum quantities that the equality holds in the equation, are used in order to reduce the MAC overhead. Admitted TSs are scheduled following the timetable computation based on EDF-order of periods T_i , for $i = 1..n$. Its computational complexity is due to the selection of the TS

to be transmitted between that with unfulfilled capacity, so it is equal to $O(n)$ in function of number n of admitted TSs.

3.5 Capacity Based Scheduler

In this section we describe a Capacity Based Scheduler (CBS) for HCCA which is suitable for serving Soft Real-Time applications. This scheduler is derived from the basic ideas of Constant Bandwidth Server, a soft real-time scheduling algorithm [3]. The original algorithm is based on EDF real-time scheduler and it provides a mechanism to serve multimedia applications with soft real-time constraints along with hard real-time applications served by hard real-time algorithms.

The Constant Bandwidth Server schedules soft real-time tasks assigning each of them, for each period, a budget, i.e., a fraction of the total execution time available for such kind of tasks, and a deadline. The tasks have to be executed before the deadlines expire. If the capacity available at the task activation is not enough for completing the considered task, the deadline is postponed and the capacity is recharged to the maximum budget value, allowing the other tasks to be executed. The admitted tasks set is served following the EDF deadlines scheduling.

WiFi networks handle traffic streams instead of tasks: some of them require temporal guarantees, while other are just best effort traffic streams. The nature of the wireless medium does not allow to transmit information which have hard real-time requirements, so here we do not have a separate real-time algorithm for them. The best effort traffic is served during the contention phase provided by medium access protocol, while the traffic streams requiring QoS guarantees will be served during contention-free phase according to HCCA protocol.

The scheduler assigns to each TS_i an ordered pair of static parameters:

Q_i : the budget, i.e., the maximum transmission time which can be assigned during a period;

P_i : the service interval of the TS_i .

In particular, Q_i is the maximum capacity, expressed in time units, that a stream i can consume in its period P_i . These parameters are computed during the admission control phase and their values are based on the TSPEC_i . They do not change during normal conditions. The ratio $U_i = Q_i / P_i$ is denoted as the factor utilization of the stream, i.e., the TS_i bandwidth.

During the scheduling, each TS_i is characterized by the following dynamic parameters, which represent the actual stream status:

c_i : the current capacity, i.e., the remaining time that can be assigned to TS_i during the next TXOP;

d_i : the absolute deadline before the budget transmission time has to finish;

p_i : the next time an uplink TS_i will be polled when it has no more data to transfer or it has exhausted its TXOP;

state : the current state of the stream and it can be one between *transmitting*, *active*, *polling*, *idle*.

The status of a TS is *transmitting* if it is transmitting; it is *active* if the TS is in the transmitting queue because it has packet to send and $c_i > 0$, moreover if it is an uplink stream it is the next flow which will be polled; a downlink stream is *idle* because it has no packet to transmit or has exhausted its capacity. Finally, the uplink stream status is *polling* if it is in the polling queue because it has packets to send and $c_i > 0$ but is still too early to be polled. Only one stream at time can be in the transmitting state.

The scheduler manages the capacity allocation to the TSs, taking into account the already assigned budgets to the admitted TSs. When a new TS asks the QAP for the right to be transmitted, the scheduler checks if the new traffic stream can be admitted to the medium without jeopardizing the guarantees of already admitted streams and if it would require more capacity than the system can provide.

The admission control test represents a sufficient feasibility condition for the scheduling algorithm and it allows to determine the set of streams which can transmit within their timing constraints, i.e., the schedulable streams set. For this algorithm the admission control test is:

$$\sum_{i=0}^N \frac{Q_i}{P_i} \leq \frac{T - T_{CP}}{T}$$

where P_i is computed as the maximum SI and Q_i is determined by means a weighted function f of Q_{min} and Q_{max} , which evaluate the minimum and maximum budget needed to transmit during a period T_i , respectively, Nominal SDUs at the mean data rate and Maximum SDUs at the peak data rate. If a stream cannot be admitted the QSTA is notified of insufficient available capacity, otherwise the scheduler updates the total used bandwidth and allocates data structures to perform TS scheduling.

After the admission control phase, the temporal evolution of the scheduler is as follows.

1. For each new admitted TS_i , at the beginning: $c_i = Q_i$, $d_i = now + P_i$, $state = ACTIVE$, where *now* is the current time.
2. Whenever a TS_i is *active* its transmission request is enqueued in a EDF queue.
3. The streams are *served* in EDF order: the scheduler extracts the next TS_i to serve from the top of EDF queue, and it sets: TS_i state to *transmitting*, $TXOP = c_i$, then it decreases the capacity c_i by the effective transmission time.
4. When a TS_i finishes to transmit, the next pending transmission, if any, is served using its current capacity and deadline.
5. When a TS_i with $c_i < min_capacity^1$ is served, if TS_i is an uplink stream, its state becomes *polling*, it is inserted in the polling queue, and the following quantities are set: $c_i = Q_i$, $d_i = d_i + P_i$, $p_i = d_i$. If its deadline is still expired, i.e., $d_i < now$, then it is postponed by another period by now, i.e., $d_i = now + P_i$. In this way, using this *recharging mechanism*, a TS_i has not to wait for a deadline expiration to recharge

¹*min_capacity* is the minimum capacity needed to transmit an SDU, and eventually CF-Poll for an uplink TS_i .

its capacity and then it is ready earlier to transmit again.

6. A TS_i remains in the *polling* state until $p_i \leq now$, then it is extracted from the polling queue, it becomes *active* and it is inserted in EDF queue.
7. When an *idle* downlink TS_i is served because it has new data to transmit, if $c_i \geq (d_i - now)U_i$ the scheduler recharges the stream capacity to the maximum value, $c_i = Q_i$, and it generates a new deadline by a period from now: $c_i = Q_i, d_i = now + P_i$. Then it becomes *active* and it is inserted in EDF queue.
8. If there are no active streams a *Contention Period* is started.

4. PERFORMANCE ANALYSIS

In this section we present some results of the performance evaluation of the described algorithms obtained through simulation. We first describe the simulation settings under which the performance analysis is carried out and the traffic model used. Then we discuss the results about admission control analysis, the null rate experienced over the medium, the unreserved capacity available for contention traffic, and the mean access delay.

4.1 Simulation settings

We used the physical layer parameters specified by the High Rate Direct Sequence Spread Spectrum (HR-DSSS) (Table 1). MAC level fragmentation, multirate support,

Table 1: MAC/PHY simulation parameters

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	Bit error rate	0 b/s

RTS/CTS protection mechanism are disabled and we assume that all nodes can directly communicate with each other, without the hidden node problem. We use the HCCA implementation described in [9] for *ns-2* network simulator [1], as a framework to implement the proposed algorithm. The analysis has been carried out using the method of independent replications. Specifically we ran independent replications of 600 seconds each with 100 seconds warm-up period until the 95% confidence interval is reached for each performance measure. Confidence intervals are not drawn whenever negligible.

4.2 Traffic model

We use two types of uplink (UL) traffic streams requiring QoS guarantees: VoIP and video. The VoIP traffic is simulated using a VoIP generator module for *ns-2* described in [5]. The VoIP streams of packets are modeled 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. Talkspurt and silence periods are distributed according to Weibull distributions [8] so to model a one-to-one conversation: $\lambda_{ON}=1.423s, k_{ON}=0.824s, \lambda_{OFF}=0.899s, k_{OFF}=1.089s$ (which yields $E[ON]=1.58s, E[OFF]=0.87s$). The employed encoding schemes are G.711, G.723.1 and

G.729A [2] with the parameters as showed by table 2. For both encoding schemes we set the TSPEC delay bound to the packet interarrival time (*period*) and the mean data rate to the peak rate during talkspurts.

Table 2: VoIP encoding schemes

Codec	G711	G723.1	G729
Frame size (<i>s</i>)	80	30	10
Period (<i>s</i>)	0.02	0.0455	0.02
Sample per packet	2	1	2
PayLoad size (<i>B</i>)	160	30	20
IP/UDP/RTP hdr size (<i>B</i>)	40	40	40
SDU size (<i>B</i>)	200	70	60
Data rate (<i>b/s</i>)	80000	12320	24000

The video stream traffic is generated using pre-encoded MPEG4 trace files from the Internet archive of traces [18]. MPEG4 encoder produces streams of variable size frames at fixed intervals [14]. They are chosen to represent a videoconference session (*LectureHQ-Reisslein* trace file) and a video streamed over the network (*Jurassic Park High Quality* trace file). The TSPEC parameters of these traffics are shown in table 3.

Table 3: Traffic parameters for video streams

video stream	VideoConf.	VideoStr.
Mean frame size (<i>B</i>)	660	3800
Maximum frame size (<i>B</i>)	11386	16745
Period (<i>s</i>)	0.033333	0.040
Mean data rate (<i>b/s</i>)	157712	770000
Peak data rate (<i>b/s</i>)	2732640	3300000

Best effort data traffic transmits using legacy 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.

4.3 Admission Control Analysis

We evaluated the number of stations admitted using the considered schedulers under different scenarios: CBR traffic only, VBR traffic only and mixed traffic. This approach shows the schedulers behavior when we introduce progressive traffic variability.

Fig. 1 shows the number of admitted G729 TSs as a function of admitted G711 streams. Because these TSs are CBR and their codecs have the same SI, the number of admitted flows is substantially similar for each scheduler except for WTTP which underutilizes the medium.

In Fig. 2 we analyze the number of admitted G723 TSs as a function of admitted G711 streams. Taking into account that G723 can have two different rates, here we show that the reference scheduler and FHCF perform worse: this is due to that these codecs have different periods (20 ms and 45.5 ms for G711 and G723.1 respectively) while these schedulers poll the stations at the smallest interarrival period (thus 20ms). That is more often than needed. In such way the computed TXOP for TSs is overestimated. RTH and the CBS have the better performances since they are derived by EDF and so they allow a more precise estimation of the needed computation time for variable traffic. They assign different SI and TXOP for each TSs. The WTTP admission control performs even worse than the reference scheduler because it takes into account only an *a priori* evaluation of the lowest Delay Bound of considered TSs. Such

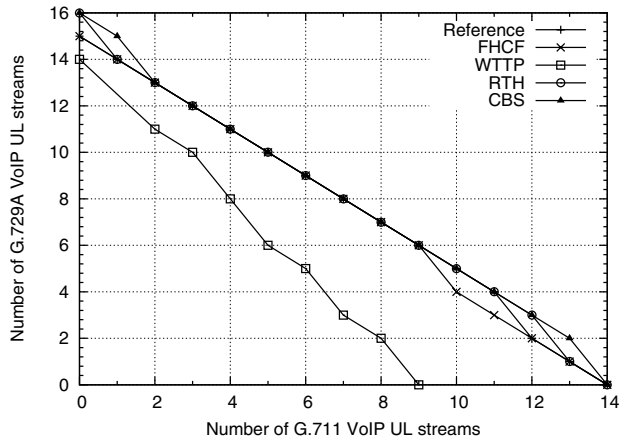


Figure 1: Admission Control: number of admitted VoIP G.729 vs. G.711 UL TSs.

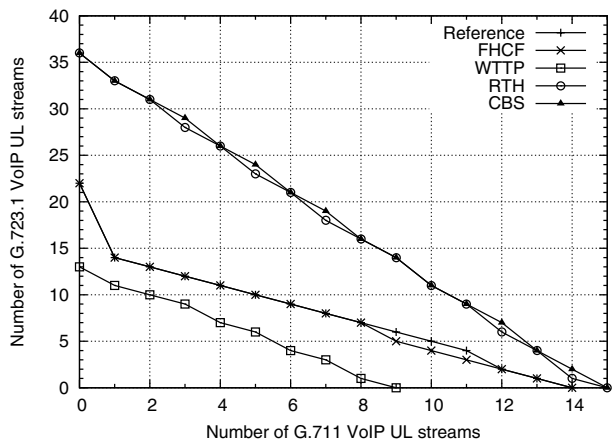


Figure 2: Admission Control: number of admitted VoIP G.723 vs. G.711 UL TSs.

parameter is the most strict requirement with respect to the other admission control tests.

Finally, Fig. 3 shows the number of admitted flows under different scenarios involving two different kinds of VBR streams, the videoconference and the videostream, as a function of the number of admitted VoIP G711 and G729 TSs. These TSs have different four periods: 33ms, 40ms, 20ms and 45.5ms. In this scenario the schedulers which are able to deal with so different TSPECs are winning. In fact CBS and RTH perform better than the others, and in particular CBS admits more flows because it computes a smaller TXOP. Instead, the reference scheduler cannot efficiently manage TSs with different TSPECs, because it polls TSs with different periods more often than needed, by setting the scheduling duration to the smallest TS period, and assigning an overestimate TXOP to the TSs. Similarly, WTTP performs even worse because of his restrictive admission control test.

4.4 Efficiency Analysis

The efficiency analysis is performed in terms of null rate, i.e., in term of the number of *Null* packets received by the QAP after it has sent a *CF-Poll* frame. This approach dis-

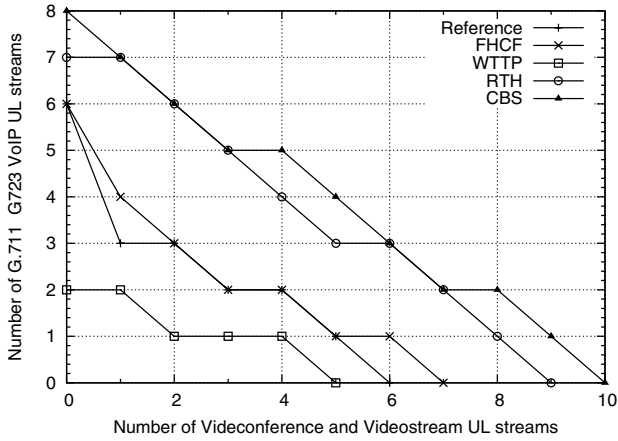


Figure 3: Admission Control: number of admitted VoIP G.711 and G723.1 vs. VC and VS UL TSs.

plays if the polling time computation is suitable for the considered traffic or if the QAP polls more frequent than necessary the QSTAs, which increases the system overhead.

We consider a scenario with four VoIP G.711 uplink TSs and four VoIP G.723.1 uplink TSs and in Fig. 4 we show the null rate value. With G711 TSs the null rate is almost

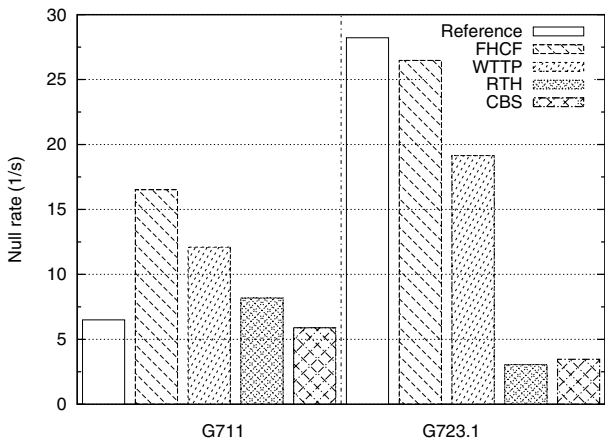


Figure 4: Null rate of UL VoIP TSs.

the same for the reference, RTH and CBS because they have the same SI and the same computed TXOP. The null rate is higher for FCHF and WTTP, because they try to empty the queue node with the FCHF strategy of polling more aggressively (like WTTP does). With G723.1 TSs the reference scheduler, WTTP and FCHF perform a higher null rate because they poll also these TSs at the 20 ms instead of 45.5ms period. The CBS and RTH poll the TSs only at their packet arrival, so their polling is more efficient. In particular, FCHF performs worse respect to the other schedulers since it adapts the TXOP to the traffic variations but maintains the same SI. This means that even if the scheduler exhausts the queued packet until the next SI, which is a good choice, it polls the QSTA when it does not have any packet to send. Note that *Null* messages are only due to the silence periods, and they are unavoidable.

In Fig. 5 we show the polling intervals for a scenario with a mix of VoIP and videoconference TSs. The reference, the FCHF and the WTTP schedulers compute the same polling interval, even if there are TSs with different interarrival time, whereas the CBS and RTH calculate two different values which allow to poll the QSTAs when they have really packets to send.

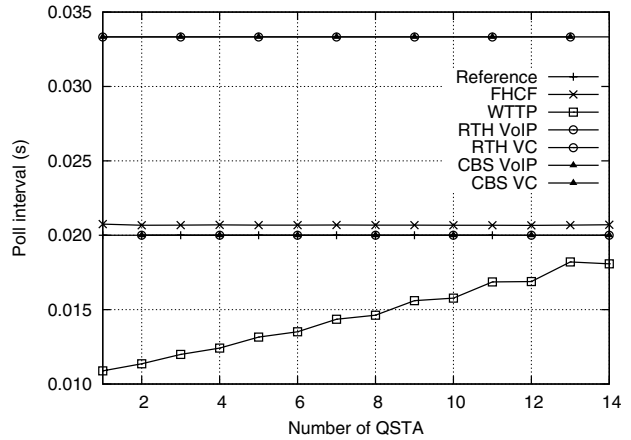


Figure 5: Poll interval of mix of VoIP and Video-Conference UL TSs.

Finally in fig. 6 we show the unreserved capacity that the schedulers leave for contention based traffic. This analy-

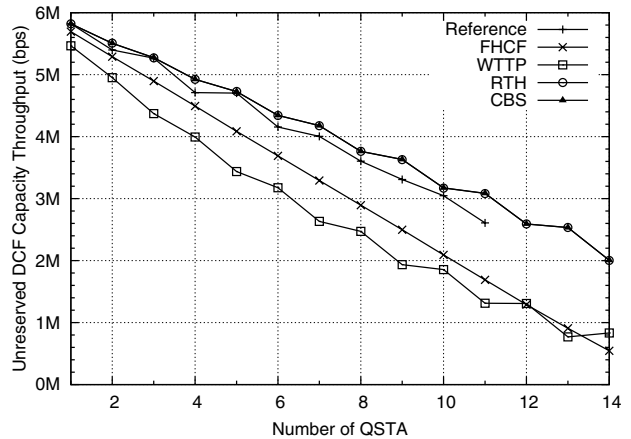


Figure 6: Unreserved DCF capacity throughput with VoIP and VBR traffic.

sis aims to evaluate the efficiency of the adopted scheduling policies: CBS and RTH schedulers have more efficient admission control schemes and less polling overhead.

4.5 Delay Analysis

In this section we want to investigate about the access delay defined as the time elapsed from the packet reaching the MAC layer to that packet being successfully acknowledged. We take into account a scenario with increasing number of upload videoconference TSs. Fig. 7 shows the results obtained. We note that the schedulers based on EDF algorithm produce a bigger mean delay respect to the others.

That is due to the fact that EDF algorithm executes a new sorting each CAP phase, where the other schedulers maintain a fixed order of TSs. In this context the CBS performs even worse because it uses EDF sorting and when there is not enough capacity to transmit it postpones the current deadline.

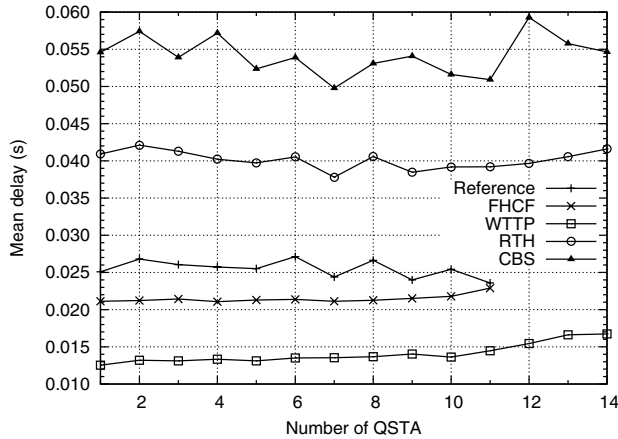


Figure 7: Mean delay of VideoConference UL TSs.

5. CONCLUSIONS AND FUTURE WORKS

In this paper we have compared five different schedulers for HCCA IEEE 802.11e networks: they are named reference scheduler, FHCF, WTTP, RTH and CBS. We have described their parameters, admission control tests, and their temporal evolution. In order to evaluate how they are suitable to support traffic streams requiring soft real-time guarantees we analyzed their characteristics through simulation. In particular the admission control phase has been tested under different scenarios. Then the efficient use of the medium has been studied considering the null rate and the polling interval produced with mixed traffic streams. Such result has been confirmed by evaluating the unreserved capacity available for contention-based traffic. While the admission control and the efficiency tests have shown that EDF-based schedulers perform better than the others, the delay analysis with VBR scenarios has illustrated that RTH and CBS offer a greater access delay. The drawback of the reference, FHCF and WTTP schedulers is essentially due to restrictive admission control test, fixed SI, and overestimated TXOP. The future works will concern the analytical comparison of the properties of the considered scheduling algorithms.

Acknowledgements

This work has been supported by FRESOR European project (Contract n. 034026).

6. REFERENCES

- [1] *Network Simulator 2*. <http://www.isi.edu/nsnam/ns/>.
- [2] *Traffic Analysis for Voice over IP*. Cisco Press, November 2001.
- [3] L. Abeni and G. Buttazzo. Integrating multimedia applications in hard real-time systems. In *IEEE Real Time-Systems Symposium*, pages 4–13, Dec. 1998.
- [4] P. Ansel, Q. Ni, and T. Turetli. FHCF: A simple and efficient scheduling scheme for IEEE 802.11e wireless lan. *Mobile Networks and Applications*, 11(3):391–403, June 2006.
- [5] A. Bacioccola, C. Cicconetti, and G. Stea. User-level performance evaluation of voip using ns-2. In *Workshop on Network Simulation Tools (NSTools) 2007*, Nantes, France, October 2007.
- [6] T. P. Baker. Stack-based scheduling for realtime processes. *Real-Time Syst.*, 3(1):67–99, 1991.
- [7] Y. L. Boudec and T. Thiran. A short tutorial on network calculus I: Fundamental bounds in communication networks. In *IEEE ISCAS*, pages IV–93–IV–96, Geneva, Switzerland, May 2000.
- [8] P. T. Brady. A model for generating on-off speech pattern in two way conversation. *Bell System Technical Journal*, 48:2445–2472, 1969.
- [9] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea. A software architecture for simulating IEEE 802.11e HCCA. In *3rd IPS MoMe*, Warsaw (Poland), Mar 2005.
- [10] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea. Design and performance analysis of the real-time hcca scheduler for ieee 802.11e wlans. *Comput. Netw.*, 51(9):2311–2325, 2007.
- [11] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea. An efficient cross layer scheduler for multimedia traffic in wireless local area networks with IEEE 802.11e HCCA. *ACM Mob. Comput. and Commun. Reviews*, 11(3):31–46, 2007.
- [12] J. Cowling and S. Selvakennedy. A detailed investigation of the IEEE 802.11e HCF reference scheduler for VBR traffic. In *13th ICCCN 2004*, Chicago, US, Oct 2004.
- [13] H. Fattah and C. Leung. An overview of scheduling algorithms in wireless multimedia networks. *IEEE Wireless*, 9(5):76–83, Oct 2002.
- [14] F. H. P. Fitzek and M. Reisslein. Mpeg4 and h.263 video traces for network performance. *IEEE Network*, 15(6):40–54, November 2001.
- [15] A. Grilo, M. Macedo, and M. Nunes. A scheduling algorithm for QoS support in IEEE802.11 networks. *Wireless Communications, IEEE*, 10(3):36–43, Jun 2003.
- [16] A. Grilo and M. Nunes. Performance evaluation of IEEE 802.11e. In *Proc. PIMRC 2002*, volume 1, pages 511 – 517, Lisboa, Portugal, Sep 2002.
- [17] R. Grow. A timed-token protocol for local area networks. In I. Electronic Conventions, editor, *Proc. Electro/82*, number Paper 17/3 in Token Access Protocols, El Segundo, Calif., May 1982.
- [18] <http://traces.eas.asu.edu/>, 2005.
- [19] IEEE802.11. Wireless LAN medium access control (MAC) and physical layer (PHY) specification, 2007.
- [20] C. Liu and J. Layland. Scheduling algorithm for multiprogramming in a hard-real-time environment. *Journal of ACM*, 20(1), Jan 1973.