32nd IEEE Conference on Local Computer Networks

Analysis of Different Scheduling Strategies in 802.11e Networks with Multi-Class Traffic

Renato Lo Cigno, Luigi Palopoli, Alessio Colombo
Dipartimento di Informatica e Telecomunicazioni, University of Trento, Italy
Email: {locigno,palopoli}@dit.unitn.it, alessio.colombo@studenti.unitn.it

***Abstract*—This paper tackles the problem of traffic and packet scheduling in HCCA, the contention-free portion of the 802.11e MAC protocol. Scheduling traffic belonging to different categories or priorities on a shared channel, as in Wireless LANs, is a multi-class complex optimization problem.**

We consider two different scheduling techniques, one based on an extremely simple positional controller and one based on the optimal solution of the linearized control problem based on a fluid approximation. Simulation results are presented discussing pros and cons of the two solutions both in terms of performances (resource exploitation and traffic differentiation properties) and in terms of implementation complexity and fairness.

I. INTRODUCTION

The diffusion of 802.11 WLANs (Wireless LAN), and the recent standardization as 802.11e [1] of the MAC (Medium Access Control) extensions for traffic differentiation and Quality of Service (QoS) support are spawning growing interest in WLAN use as access support in many different situations, including public HotSpots support and integration with 3-4G cellular networks.

The commercial use of WLANs, as well as the support of multi-media traffic, calls for an efficient —and predictable— use of resources, which suggests supporting premium and real-time traffic with centralized scheduling rather than with a contention-based access. The MAC protocol or 802.11e allows for the coexistence, with a dynamic time-sharing approach, of a polling-based centralized scheduling, named HCCA (Hybrid Coordinated Channel Access), with a backward compatible, evolute version of the contention based MAC of 802.11 named EDCA (Enhanced Distributed Channel Access).

EDCA remains an entirely decentralized protocol based on random access for stochastic coordination. Access differentiation is obtained with the per-flow customization of the protocol parameters, such as the Inter Frame Spaces (IFSs) and the contention window size.

HCCA overcomes the limitations and bugs of the optional 802.11 PCF (Point Coordination Function) centralized management scheme, and offers all the means required to implement efficient and guaranteed traffic management and differentiation. Given that propagation delays in WLAN are marginal, the use of polling does not jeopardize channel utilization. Besides, the use of feedback and prediction techniques can maintain access delay comparable to a random access scheme also under low loads.

In this paper we analyze different possibilities for the scheduling algorithms to be used within HCCA. The paper is organized as follows. Sect. II discusses the problem of scheduling in 802.11e networks and defines the problem model and constraints. Sect. III extends the control-optimization problem defined to the case of multiple traffic categories, both with and without priorities. Sect. IV presents the numerical results obtained via simulation with the ns [2] simulator. Sect. V ends the paper with remarks and comments.

II. PROBLEM OVERVIEW AND RELATED WORK

The MAC protocol of 802.11e [1] alternates dynamically instances of two entirely different protocols: HCCA during periods called *Contention Free* (CFP) and EDCA during periods with contention (CP). The schemes vaguely resembles a time division multiplexing approach, with a centralized entity, called *Hybrid Coordinator* (HC), which is normally the Access Point (AP) of an Infrastructured Basic Service Set (IBSS)[1], that manages both the CFP–CP subdivision, generates beacons and is the master of the HCCA polling protocol. HC can run whatever scheduling algorithms is deemed fit to obtain the proper allocation of resources to the Stations (STA) within the IBSS.

The detailed description of all features (most of them optional) of 802.11e is out of scope of this paper. We instead give an overall view of the protocol which is functional to the understanding of our work.

Fig. 1 illustrates the alternation of contention based and polling based periods together with the transmission of beacons within the 802.11e superframes repetitions. Beacon transmissions should be as regularly spaced as possible, and the superframe is an integer number of beacon transmissions (set to 2 in the figure). CFPs and CPs need not to be equally spaced, nor to have the same duration: the only requirement is that a new superframe starts with a CFP. Resources allotted to stations are time-based[2] and they are called TXOP (*Transmission Opportunity*). Within a TXOP a STA can transmit one or more frames; if the AP must transmit to some STA the HC can allocate TXOPs to itself. The explicit allocation of resources via polling solves the well know unfairness problem between uplink and downlink [3], making HCCA use quite appealing.

---

[1] An IBSS in 802.11 context can be sees as a fundamental service unit with dedicated resources —the spectrum— where a portion of the infrastructure — the AP— provides service to users/clients: laptops, PDAs or whatever device can access the WLAN.

[2] This is a major modification w.r.t. legacy 802.11 where channel access is based on data: the contention is won for a single MPDU (MAC Protocol Data Unit) transmission, regardless of its size or the bit-rate of its transmission.

0742-1303/07 $25.00 © 2007 IEEE
DOI 10.1109/LCN.2007.80

IEEE computer society

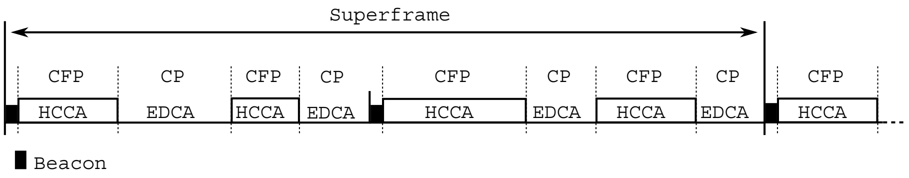

Fig. 1. Dynamic alternation of CFP and CP periods within beacons and superframes

TABLE I
TSPEC PARAMETERS RELEVANT IN THIS WORK

TSPEC parameter	Description
Maximum SI	Maximum admitted time between consecutive polls
Min. Data Rate	The scheduler is expected to allocate at least the time needed to serve this data rate
Mean Data Rate	The average expected traffic
Peak Data Rate	The maximum data rate of the stream
Delay Bound	Maximum time a MSDU can wait to be transmitted
MSDU size	Frame or MAC Service Data Unit dimension, both nominal and max.

STAs must signal with the HC to negotiate polling parameters and inclusion in the polling list. Signaling is done exchanging Traffic Specifications (TSPEC). It is assumed that the HC will run some Connection Admission Control algorithm (see [4], [5] for examples of CAC specific for broadcast channels and 802.11 networks) in order to guarantee the service negotiated with stations. The CAC mechanism must be tailored to the scheduling algorithm, however CAC analysis and its mapping to scheduling is out of the scope of this paper. There are many TSPEC parameters ranging from MPDU size to the maximum time between successive polling —SI (Service Interval). Tab. I summarizes the TSPEC parameters that will be used later on in this paper.

Many works were dedicated to the analysis of differentiation and performance when only EDCA is used as a first enhancement of the legacy DCF function. Non exhaustive examples are [6]–[9], where the analysis can be based on theoretical models or simulations. However, many authors recognize that it is difficult to obtain high resource utilization with good support of real-time, high priority traffic only using stochastic differentiation.

Mixing HCCA and EDCA should solve the problem of real-time traffic support. What we are concerned with in this work is the performance of the HCCA scheduling algorithm.

Works dedicated to HCCA proposals and analysis are far less in number and more recent than those dedicated to EDCA. When a centralized, polling protocol is used the access problem can be efficiently reduced to a scheduling problem with uncertainties. In fact, differently from a multiplexing problem, in multiple access the information available at the scheduler (the HC) are limited and made stale by the feedback delay. In 802.11e information available at HC includes the TSPEC and the backlog (queue length) information sent by stations within each frame, expressed either in bytes or number of packets.

Given the framework above, scheduling techniques can be broadly classified in open- and closed-loop. Open-loop techniques use only TSPEC information and are static. The advantage is a small computational complexity, since the schedule is (re)computed only on acceptance of new flows. Closed-loop techniques use also the feedback information and can be reactive or predictive, trying to optimize performance extrapolating the source behavior in the next polling cycle from the feedback information and previous behavior.

The papers [10] and [6] discuss the use of variable service intervals trying to meet the deadlines of frames, the first one with an open-loop approach, and the second one accounting also for QSTAs feedbacks. Similar is also the approach described in [11], where an open-loop predictive scheduler is proposed. The prediction algorithm is based on measures of the actual traffic sent by stations.

In [12] the authors use a continuous time modeling and use predictive techniques, but the approach does not seem to be prone to optimization applying control-theoretical results.

In [13] we have proposed a very simple Max-Min fair closed-loop proportional controller, and showed that in simple, homogeneous scenarios a closed loop approach with low complexity outperforms any open-loop scheduler. In [14] we have explored the theoretical feasibility of a more sophisticated approach based on prediction and the optimal solution of a linear programming formulation of the problem.

In the reminder of the paper we build on top of these theoretical results and explore the use of closed-loops schedulers in realistic scenarios with different classes, different priorities and Variable Bit Rate (VBR) real-time traffic.

As discussed above the problem of assigning resources in the HCCA polling protocol can be reduced to a scheduling or control problem, where the resource assignment can be based both on static descriptive parameters (the TSPECs) and on dynamic backlog information sent by stations to the HC within each frame. The resource to be assigned is transmission time and the system is discrete both in time and resources. The discrete time identifies the *Polling Cycle* (PC) number. Resources are quantized because of frame size constraints, and because the protocol forbids fragmenting frames just to fit them in the TXOP.

Let k be the discrete time index, N^f be the total number of flows admitted to the polling cycle, $T^s(j,k)$ the arrival rate in bytes for flow j during PC k, and $r_a(k)$ the total amount of resources to be assigned in PC k. We define:

- $B(j,k)$: backlog in bytes of flow j at the end of PC k;
- $r(j,k)$: resources (transmission time) allocated to flow j in PC k;

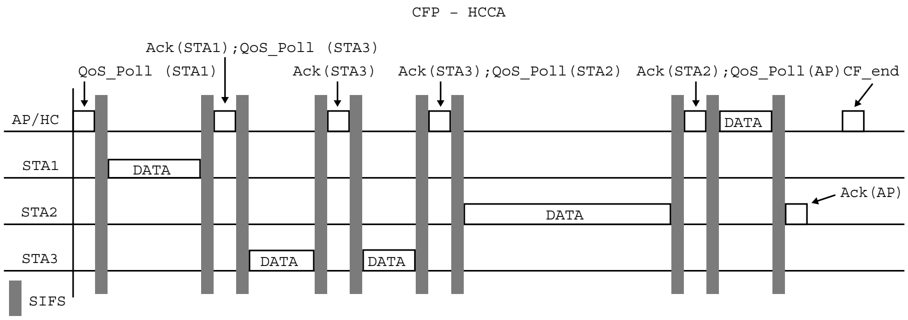

Fig. 2. Transmission Opportunities Assignment within a polling cycle

Following our previous works in [13], [14], but also [12], the dynamic behavior of the system is given by:

$$\begin{aligned} B(j,k) &= \max\{0, B(j,k-1) - r(j,k) + T^s(j,k)\}, \\ &\text{for } j = 1, \ldots, N^f \end{aligned} \tag{1}$$

subject to the constraint

$$\sum_{j=1}^{N^f} r(j,k) \le r_a(k) \tag{2}$$

The goal of the scheduler is finding the best possible dynamic allocation of all $r(j,k)$. We consider both up- and down-link flows sharing the same resources and following the same dynamic behavior, and the HC treating them in exactly the same way, even if the feedback delay for down-link flows could be smaller: Accounting for this difference is not complex, but would introduce a bias favoring down-link flows that would have faster dynamics.

III. MULTI-CLASS SCHEDULING

As already discussed, this work leverages on the results obtained in [13], [14]. We assume the same global results for stability and feasibility of the schedulers. We are interested in the extension of those schedulers to multi-priority (or class) traffic, and in presenting a comprehensive set of results in realistic scenarios.

We limit the analysis to strict priorities, with and without minimum guarantees (i.e., higher priorities are entitle to grab all resources if they can, unless a minimum is guaranteed to lower priority traffic). Let N_p be the number of priority classes, with priority decreasing from 1 to N_p; i.e., class i has higher priority than class h if $i < h$. If no minimum resources are guaranteed to low priority classes, strict priority means that the amount of resources available to class i flows at time k is

$$0 \le r_a(i,k) = r_a(k) - \sum_{h=1}^{i-1} r_a(h,k) \tag{3}$$

if instead a minimum $r'_a(i,k)$ is guaranteed to every class, then we have

$$\begin{aligned} 0 < r_a(i,k) &= r'_a(i,k) + \\ \max\{0, &\left[r_a(k) - \sum_{h=1}^{N_p} r'_a(h,k) - \sum_{h=1}^{i-1} r_a(i,k) \right]\} \end{aligned} \tag{4}$$

It is the task of a CAC function to guarantee that (4) or (3) are correctly satisfied.

A. An Equivalent Bandwidth Approach

Equivalent Bandwidth (EB) approaches for resource allocation have been studied in different contexts (see for instance [15] for a traditional approach, but also [16] for a more modern perspective). We only use this approach as a benchmark applying EB to the reference scheduler described in the 802.11e standard, which is suitable only for CBR (Constant Bit Rate) traffic, thus to be applied to VBR (Variable Bit Rate) traffic requires the allocation of resources larger than the average.

In our case the equivalent bandwidth can be defined as the amount of resources EB(j) allotted to station j during one PC so that the probability that the source generates more traffic is smaller than a probability p^o_j:

$$\text{EB}(j) : \mathbf{P}[T^s(j,k) > \text{EB}(j,k)] < p^o(j) \tag{5}$$

With this definition using (3) or (4) for priority separation is not natural, while it is very easy to separate different priority traffic using different values of $p^o(j)$.

The biggest problem of EB approaches in real scenarios is the difficulty to characterize sources stochastically so that (5) can be computed with precision. For this reason, in spite of their elegance and simplicity, they are rarely used. In this paper we use Markovian synthetic traffic for the evaluation, so that the problem of identification does not arise.

B. Proportional Control and Assignment

The Proportional Control Scheduling based on a Max-Min criterion was first proposed in [13], showing the advantages closed-loop scheduling vs. open-loop approaches in simple homogeneous traffic scenarios.

Applying either (3) or (4) to obtain priority differentiation the Max-Min fair proportional scheduling for traffic flow j in class i is

$$r(i,j,k) = r'(i,j,k) + \\ \beta(i,j)\left[r_a(i,k) - \sum_{h=1}^{N_i^f} r'_h(h,k)\right] \tag{6}$$

where N_i^f is the number of flows in class i,

$$\beta(i,j) = \frac{B(i,j,k-1)}{\sum_{j=1}^{N_i^f} B(i,j,k-1)}$$

and $r'(j,k)$ is 0 if (3) is used.

The rational of this scheduler is trying to minimize the probability that packets are lost due to buffer build-up. The different typology of flows is taken into account with a different minimum guaranteed assignment $r'(j,k)$; also the minimum assignment can be function of k (the polling cycle counter), because requirements can vary over time, but also because polling cycles can be of different duration. Extra-minimum resources are assigned trying to equalize all buffer levels. The complexity of this closed-loop scheduler is extremely small, since it requires only one multiplication per flow.

C. An Optimal Predictive Scheduler

In [14] we have proposed a closed-loop, prediction-based optimal scheduler based on the application of Model Predictive Control (MPC) [17].

The idea of the MPC is to consider the dynamic evolution of the system over an horizon of some steps and choose the control value by solving an optimization problem. In particular, for our system, we consider, for each class, the vector of state variables associated to the buffer levels $\mathbf{B}$ and the optimization problem minimizes the maximum level of a buffer (i.e., the ∞-norm of the buffer level $\mathbf{B}$). In order to introduce a moderate complexity in the problem, we consider the system evolution for only one step. The use of the ∞-norm as a cost function enables the following set up for the optimization problem [18]:

$$\begin{aligned} \min_{r(i,1),\ldots,r(i,N_i^f),h_i} \quad & h_i \\ & h_i \geq B(i,j,k+1) \geq 0,\ \forall j \\ & r(i,j,k) \geq r'(i,j,k+1)\ \forall j \\ & \sum_{j=1}^{N_i^f} r(i,j,k+1) \leq r_a(i,k+1) \end{aligned} \tag{7}$$

The decision variables of this problem are the transmission times $r(i,j,k)$. Buffer levels $B(i,\,j,\,k+1)$ are not simply induced by the choice but also by the arrival rates $\mathbf{T}^s(i,j,k+1)$ that are not known at the beginning of each round. We solve this problem by using a predictor. The predictor is for us a *given* component and, in principle, it can be adapted to the different types of application. For instance, in most cases traffic dynamics are much slower than the SI and the arrival rate of the previous round $\mathbf{T}^s(i,j,k)$ are a good predictor of $\mathbf{T}^s(i,j,k+1)$.

The constraint $\mathbf{B}(i,j,k+1) \geq 0$ is useful since: 1) it avoids over-allocations of bandwidth to a channel, 2) it allows eliminating the $\max(\cdot)$ nonlinearity in (1). Therefore the resulting problem is a linear optimization and it is parametrized by the current level $\mathbf{B}(i,j,k)$ of the buffers and to the predicted vector of the arrival rates $\mathbf{T}^s(i,j,k+1)$.

There are at least two points that deserve our consideration:

- is the algorithm well-posed (i.e., is the optimization problem feasible)?
- is the algorithm efficient?

As far as the first point is concerned, [14] proves that the problem is feasible if and only if the following holds:

$$\forall i,\ r_a(i,k+1) \geq \sum_{j=1}^{N_i^f} r'_a(i,j,k+1). \tag{8}$$

As far as complexity is concerned, the solution of a linear program is known to be a polynomial problem (w.r.t. the number of decision variables and of constraints), but in the general case it entails a non-negligible computation effort. In our particular case, however, it is possible to take advantage of the particular structure of the problem and it is possible to solve it with the algorithm in Fig. 3.

Skipping technicalities, the basic idea is very simple. Line **(a)** verifies the feasibility of the problem. The check in Line **(b)** verifies whether or not the minimum assignment of bandwidth $r'(i,\,j,\,k+1)$ is sufficient to completely deplete each buffer. In the affirmative case each station is simply assigned $r'(i,\,j,\,k+1)$. If the test fails, from Line **(c)** on we search for the optimal assignment. The rational of this search is to give the maximum bandwidth to the group of stations for which the sum of the current level of the buffers and of the predicted arrival rate is maximum. This search can be proved to converge to the optimum of Problem (7) [14]. The primary source of complexity in this algorithm is is the ordering of a vector (Line **(d)**) and it is $n \log n$.

D. Reclaiming Unused Resources

The details of the HCCA protocol prevent the possibility of assigning TXOPs perfectly tailored to the sources needs if these do not behaves deterministically. In particular rate variations and frame size changes can lead to waste of resources, due to the limited prediction capabilities of the HC as well as the limited amount of information that can be fed back to the HC by the STAs[3]. The time assigned to a source and unused, either because it does not have any additional frame to send, or because the frame to be transmitted next does not fit into the remaining time, can be freed by sending a Null Frame. If no action is taken by the HC this spare time will go to the following EDCA-based period. However, under medium to

---

[3] The field in the header used for feedback can contain the number of frames in the buffer or the buffer size in bytes, but not the detailed description of the size of each frame in the buffer.

```
In:     $B(i, j k), T^s(i, j, k), r'(i, j, k+1), r_a(i, j, k+1)$
Out:    $r(i, j, k+1)$
    $T^s(i, j, k+1) =$ predict($T_s(k)$)
    $\phi(i, j, k+1) = T^s(i, j, k+1) - r'(i, j, k+1)$
    $\psi(i, k+1) = r_a(i, k+1) - r'_a(i, k+1)$
    finished = false
(a) if $\psi(i, k+1) < 0$
        return infeasible;
(b) if $\sum_{j=1}^{N_i^f} B(i, j, k) + \phi(i, j, k+1)) - \psi(i, k+1) \leq 0$
        $B(i, j, k+1) = 0 \ \ \forall j$
    else {
(c)     $\forall j, B(i, j, k+1) = B(i, j, k) + \phi(i, j, k+1)$
        //$\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$ is an index set
(d)     $\mathcal{H} =$ sort_increasing_order($[B(i, 1, k+1), \ldots, B(i, N_i^f, k+1)]$)
        $i = 1$
        while $((finished == false)$ and $\mathcal{H} \neq \emptyset)$ {
            $B^* = \frac{\sum_{j \in \mathcal{H}} (B(i, j, k) + \phi(i, j, k+1)) - \psi(i, k+1)}{|\mathcal{H}|}$
            if $B(i, h_j, k) + \phi(i, h_j, k+1) > B^*$ {
                $B(i, h_j, k+1) = B^* \ \forall \ h_j \in \mathcal{H}$
                finished = true
            } else
                $\mathcal{H} = \mathcal{H} \backslash \{h_i\}$;
                i = i + 1;
        }
    }
    $\forall j \, r(i, j, k+1) = B(i, j, k+1) + T^s(i, j, k+1) - B(i, j, k+1)$
```

Fig. 3. Optimization algorithm

heavy load conditions performances of high priority, HCCA-based traffic can be affected non-marginally by this waste of resources, and reusing the left-over resources within the HCCA protocol at the end of each polling-cycle can be very effective.

Since most of the left-over resources are due to quantization problems, in re-scheduling them care must be put to avoid excessive resource fragmentation. We selected two possible strategies. The first one is trivial, simply assigning all left-over resources to the one station that has the largest backlog (if more than one exists we select one at random); the rational is that left-over resources are normally just enough to support the transmission of one-two frames, and finding an "optimal" assignment is too costly to be worth. We call this rescheduler *greedy*. The second one is used only with the Optimal Predictive Scheduler and re-re-applies the same optimization procedure as if a new CFP was started. This approach may not be feasible in real networks due to the lack of time[4].

IV. NUMERICAL RESULTS

We use the ns-2 [2] simulator to evaluate the different scheduling algorithms described in Sect. III. The scheduling algorithms as well as the traffic sources described in Sect. IV-A have been added to ns-2 and are available at [19]. For the sake of brevity we restrict the analysis to the case of two different priority classes with identical sources (Sect. IV-B) and a single priority class, where non-homogeneous sources share the medium (Sect. IV-C). All simulation points are averages obtained with the batch-means technique; simulations are stopped when the confidence intervals are are within 10% of the point estimate with a 95% confidence level.

A. Evaluation Setup and Parameters

In order to have a clear and understandable picture we use synthetic traffic on top of UDP to evaluate the performances. We have defined 2- and 3-states discrete time Markov generators where packets are generated upon leaving a state, with characteristics determined by the state itself. In each state, parameters like the generated rate and the packet size are changed. The 2-state sources are a good representation of hard real-time applications like conversations, while the more

---

[4] In a normal CFP, the HC can use the whole duration of the preceding CP to run the scheduling algorithm, while in re-scheduling, there is only a scant PIFS (Point Inter-Frame Space) of a few microseconds to run it.

TABLE II
SOURCE PARAMETERS RELEVANT TO SCHEDULING

Param.	2-state	3-state
Min. T^s	64 kbit/s	64 kbit/s
Mean T^s	128 kbit/s	409 kbit/s
Max. T^s	640 kbit/s	640 kbit/s
Nom. MSDU size	400 bytes	280 bytes
Max. MSDU size	1200 bytes	1200 bytes

TABLE III
PRINCIPAL SIMULATION PARAMETERS

MAC Protocol	802.11e / HCCA
PHY Protocol	802.11b
Prop. delay	2 μs
PHY Tx speed	11 Mbit/s
SI	50 ms
STA buffer size	50 pcks
MSDU Delay Bound	150 μs or ∞
p^{oh}; p^{ol}	0.01; 0.2

complex 3-state ones can be representative of streaming video with milder soft real-time requirements.

We only present results for piecewise constant packet arrival rates (i.e., the transition time out of a state is constant) typical of real-time applications; however, results obtained with exponential inter-arrival times yielded similar results. We consider one flow per station, with all flows in up-link, which is the worst-case, albeit unrealistic, scenario for HCCA.

Table II summarizes the main parameters of the two traffic source used, while Table III presents the most relevant simulation parameters and default values if not otherwise stated.

Since we are concerned with real-time applications, MSDUs of the high priority class will have stringent requirements as far as the access delay is concerned. If one considers 2–400 ms the overall end-to-end delay budget available for a conversation, the delay bound within the access network cannot exceed 150 ms which is three times the nominal SI[5], which means that either MSDUs are transmitted within the third PC after their generation or they are discarded by the MAC protocol.

B. Different Priorities

We start analyzing a scenario with two priority classes, high (H) and low (L), homogeneous 2-state sources and no minimum per-source guarantee for either classes. The delay bound for H sources is 150 ms, while for L ones is set to ∞, so that losses are due to buffer overflow and not to frame discards. Fig.4 reports the loss rate of L sources when either 3 (upper plot) or 7 (lower plot) H sources are present as a function of the number of L sources present. Results with different numbers of H sources are similar as far as the system is not overloaded by H sources alone. High priority sources do not experience any loss with the MPC and MMF schedulers, while at very high loads some packets are discarded due to the expiration of the delay bound with the EB scheduler, the loss is however smaller than p^{oh}. The arrow on the abscissas of the upper plot indicates an hypothetical CAC threshold computed based on the EBs. It is clear that if this limit is met also the EB approach guarantees the "promised" performance. In the lower plot H sources alone saturate the resources applying EB based CAC, so that no L sources should be admitted, with a clear

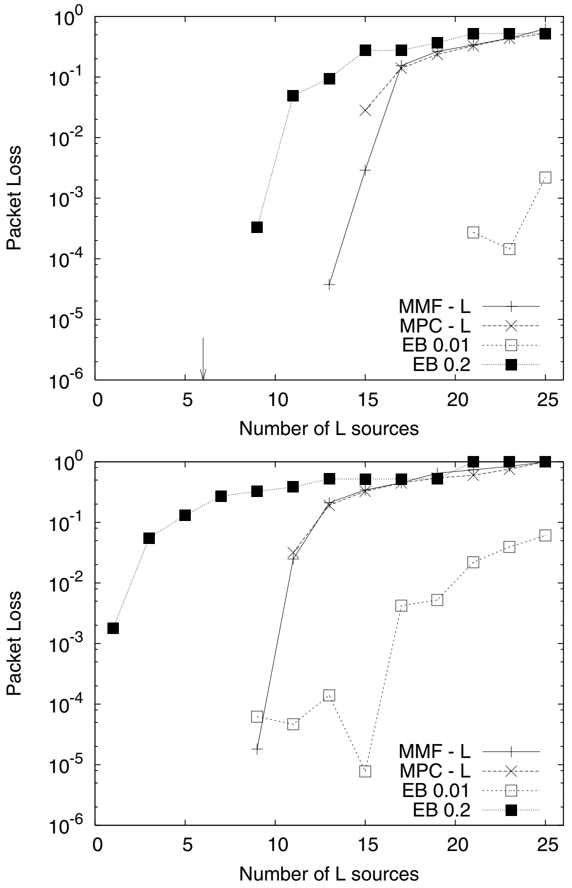

Fig. 4. Loss rate for the low priority class when no rescheduling is used; no guaranteed minimum; 3 high priority sources in the upper plot, 7 in the lower

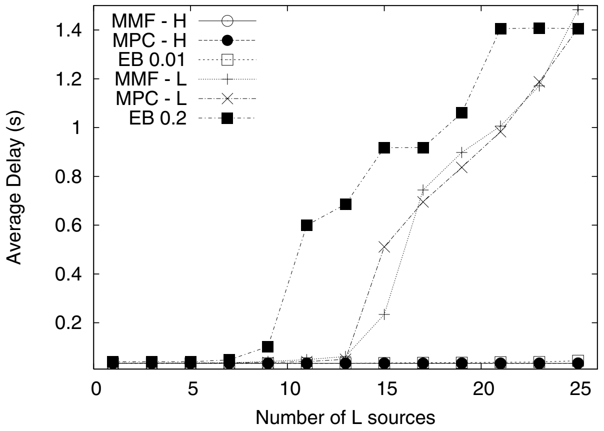

Fig. 5. Access delay for the high and low priority classes when no rescheduling is used; 3 high priority sources; no guaranteed minimum

---

[5] Further reducing the SI is not practical for many reasons, including quantization and resource waste

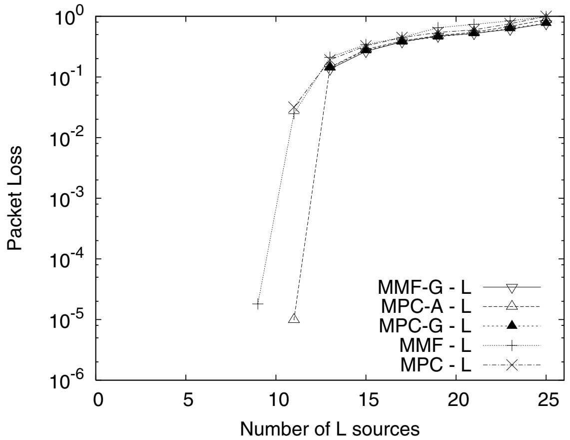

Fig. 6. Loss rate for the low priority class with and without rescheduling; 7 high priority sources; no guaranteed minimum; MPC and MMF only

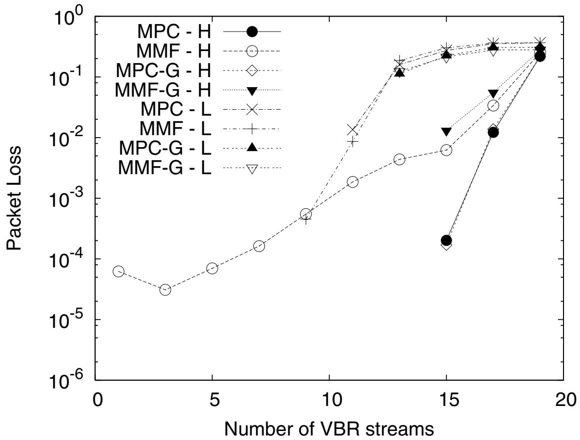

Fig. 7. Loss rate for the high and low priority classes in case of guaranteed minimum both with and without rescheduling for the MPC and MMF schedulers only; 7 high priority sources;

waste of resources. It must be noted that in any case both the closed loop schedulers provides a much better performance, interestingly without any meaningful difference between the very simple MMF and the more complex MCP schedulers.

Fig. 5 completes the picture reporting the average access delay of the frames in case of 3 H sources (results for 7 confirm the same behavior). H sources experience the minimum possible average delay of 25 ms (one half of the SI, which cannot be appreciated on the plot scale), while for L sources the average delay increases linearly with the load until the buffer is completely saturated and the loss rate very large. Also w.r.t. the delay the BE scheduler performs poorly.

To complete the picture of this first simple scenario, Fig. 6 reports a comparison of the L sources loss rate when wasted resource reclaiming via re-scheduling is used or not; we stick to the case of 7 H sources. The -G apposed to the scheduler name identifies the greedy reclaiming, while -A apposed to the MPC scheduler means that the optimization algorithm is re-applied. Being open-loop the BE schedulers do not admit rescheduling. The benefits of rescheduling are evident though marginal, allowing the admission of 1-2 more low priority sources with acceptable performances. Given that the -A rescheduling does not give any benefit compared to the -G one, and since its feasibility is questionable, we do not consider it anymore in the sequel.

Fig. 7 analyzes the case when the minimum assignment as computed using the standard MAC algorithms and the minimum generation rate of sources is guaranteed also to low priority traffic. For the sake of simplicity we only consider the MMF and MPC schedulers. Also in this case traffic differentiation is achieved and performances guaranteed both with and without rescheduling. Notice that rescheduling in this case does not provide substantial advantages and, also, the MMF scheduler seems to have some problems in handling the more complex case of guaranteeing minimums and correctly sharing extra resources: its interaction with the quantization and the details of the MAC protocol requires additional investigation.

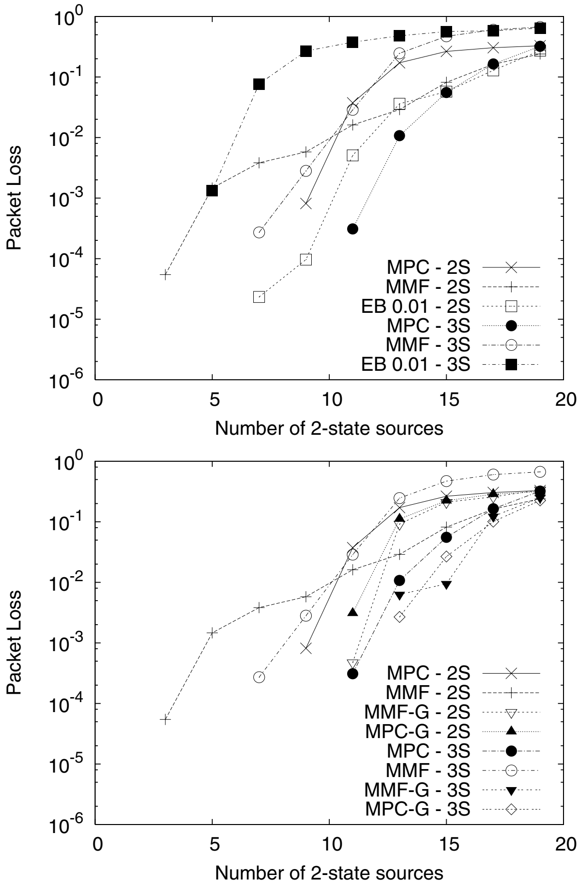

Fig. 8. Loss rate in case of 2- and 3-state sources sharing the same priority class as a function of the number of 2-state sources; delay bound 150 ms; three 3-state sources; upper plot no rescheduling and EB schedulers; lower plot comparison between re-scheduling and not

C. Non Homogeneous Sources

In this scenario we analyze the capability of schedulers to differentiate not between different priorities, but between flows with different characteristics, reflected in different TSPECs, that share the same access priority. Fig. 8 plots the loss rate of both 2- and 3- state sources as a function of the number of 2-state sources when three 3-state sources are present. The upper plot compares the results among closed-loop and EB scheduling without applying rescheduling, while the lower plot show the advantage of rescheduling in this scenario.

In this case the limits of EB scheduling becomes evident, since the same loss rate target results in very different performances. The reason lies in the fact that for such low target loss rates, the EB converges to the peak rate for both types of sources, which however, spend different times in the peak state rate. Given the re-normalization of assignments done by the MAC during overload periods, this results in unfair behavior toward sources with higher average traffic.

The MPC scheduler has good and fair performances, while the behavior of MMF is more complex, also due to the fact that, lacking prediction, it bases it behavior only on the buffer level. The impact of rescheduling in this case is much higher than in case of different priorities. The reason lies in a sort of automatic resource reclaiming implemented by the priority queuing. In fact, when scheduling low priority traffic, the HC can already take into account time unused by high priority traffic and assign it to L sources, so that at the end of the polling cycle the resources to be reclaimed are relative only to low priority traffic.

V. DISCUSSION AND CONCLUSIONS

In this work we have analyzed different scheduling techniques suitable for implementation within the HCCA framework of the 802.11e standard. Starting from theoretical works presented in [13], [14], we have extended the definition of two different closed-loop schedulers to the case of multi-priority traffic and to the case of heterogeneous traffic.

The results have been compared with a EB scheduler derived from the standard reference one and working under 'ideal' conditions since the traffic loading the network is derived from synthetic Markovian VBR sources and allows for the exact computation of EB, a situation that with real sources is impossible.

The closed-loop schedulers show better performances under any scenario, with the one based on Model Predictive Control and optimization techniques yielding better results, although also the simpler Max-Min Fair Proportional controller performs satisfactorily.

Details of the 802.11 MAC protocols leads to waste of resources due to quantization and packetizing problems. These resources can be reclaimed and reassigned by the HC by rescheduling them at the end of the polling cycle. This approach proved to increase performances, specially in case of heterogeneous sources sharing the same priority class.

An interesting problem open for future work is the possibility of running an optimization, MPC-based procedure that directly takes into account quantization and packetizing, for instance by applying Branch and Bound techniques starting from the optimal solution found with a fluid model, and comparing this solution with simple, approximate techniques like the MMF scheduler with greedy rescheduling.

ACKNOWLEDGMENTS

The authors thanks Paolo Larcheri for providing the initial version of the ns-2 code for the traffic sources and the Max-Min proportional scheduler.

REFERENCES

[1] "IEEE 802.11e-2005, IEEE Standard for Information Technology–Telecommunications and information exchange between systems," Part 11: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.

[2] "ns-2, network simulator (ver.2)," URL: http://www-mash.cs.berkeley.edu/ns.

[3] A. Grilo and M. Nunes, "Performance evaluation of ieee 802.11e," in *Proc. of IEEE PIMRC'02*, Lisbon, Portugal, September 2002.

[4] J. Ni, D. H. K. Tsang, S. Tatikonda, and B. Bensaou, "Optimal and structured call admission control policies for resource-sharing systems," *IEEE Transactions on Communications*, vol. 55, pp. 158 – 170, Jan. 2007.

[5] W.-F. Wang, D. H. K. Tsang, and B. Bensaou, "Admission control for variable bit rate traffic using variable service interval in ieee 802.11e wlans," in *Proc. of ICCCN 2004*, 2004, pp. 447 – 453.

[6] D. Skyrianoglou and N. Passas, "Traffic scheduling in ieee 802.11e networks based on actual requirements," in *Radio Network Management '04 (RNM'04)*, Athens, Greece, 2004.

[7] J. He, L. Zheng, Z. Yang, and C. T. Chou, "Performance analysis and service differentiation in ieee 802.11 wlan," in *28th Annual IEEE International Conference on Local Computer Networks (LCN'03)*, Bonn, DE, October 20–24 2003.

[8] Z.-N. Kong, D. H. T. Tsang, B. Bensaou, and D. Gao, "Performance analysis of ieee 802.11e contention-based channel access," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 2095–2106, December 2004.

[9] Y. Xiao, "Performance analysis of priority schemes for ieee 802.11 and ieee 802.11e wireless lans," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 1506–1515, July 2005.

[10] A. Grilo, M. Macedo, and M. Nunes, "A scheduling algorithm for qos support in ieee 802.11e networks," *IEEE Communication Magazine*, pp. 36–43, June 2003.

[11] L. Yang, "P-hcca: A new scheme for real-time traffic with qos in ieee 802.11e based networks," aPAN Network Research Workshop 2004.

[12] P. Ansel, Q. Ni, and T. Turletti, "Fhcf: An efficient scheduling scheme for ieee 802.11e," *ACM/Kluwer MONET journal*, 2005, special issue devoted to WiOpt'04.

[13] P. Larcheri and R. Lo Cigno, "Scheduling in 802.11e: Open-loop or closed-loop?" in *Proc. of IFIP WONS 2006*, Les Ménuires, France, January 18–20 2006.

[14] L. Palopoli, R. Lo Cigno, and A. Colombo, "Control and optimisation of hcca 802.11e access scheduling," submitted for publication. Pre-print available on-line URL: http://dit.unitn.it/locigno/preprints/PaLoCo07.pdf.

[15] A. I. Elwalid and D. Mitra, "Effective bandwidth of general markovian traffic sources and admission control of high speed networks," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 329–343, 1993.

[16] C. A. Viana Melo and N. Fonseca, "Envelope process and computation of the equivalent bandwidth of multifractal flows," *Comput. Networks*, vol. 48, no. 3, pp. 351–375, 2005.

[17] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice: a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[18] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming," *IEEE Trans. Automatic Control*, vol. 47, 2002.

[19] "Markovian sources and scheduling extensions to ns-2," URL: http://networking.dit.unitn.it/tools/802.11e-scheduling.