Scheduling in 802.11e: Open-Loop or Closed-Loop?

Paolo Larcheri, Renato Lo Cigno
Dipartimento di Informatica e Telecomunicazioni – Università di Trento
Via Sommarive, 14 – 38050 Povo, Trento, Italy
e-mail:paolo.larcheri@gmail.com,locigno@dit.unitn.it

***Abstract*— Scheduling in 802.11e networks is managed by the Hybrid Coordination Function (HCF), located within the access point. HCF has access both to traffic descriptions (TSPEC) and to feedback information sent in every frame by stations. In this paper we discuss the use of open-loop or closed-loop scheduling; the first one is based only on the TSPECs, while the second one relays also on the feedback information from stations and builds upon classical control theory design.**

We discuss how closed-loop scheduling can be used to manage both the QoS guaranteed traffic and the best-effort traffic with a non-marginal performance improvement compared to open-loop scheduling algorithms. We propose a simple max-min fair scheduling algorithm based on a positional controller which measures buffer levels. The controller is up- and down-clipped to meet strict QoS guarantees, while optimally distributing stochastically guaranteed and spared resources.

Simulation results based on ns-2 are presented to support the theory and the design, showing that the proposed scheduler is robust and performs always better than open loop scheduler in presence of traffic uncertainties.

I. Introduction

Wireless access through 802.11 W-LANs is spreading fast, becoming quickly as ubiquitous as cellular networks in places where access to the Internet is demanded.

While there are not hints that the use of the ISM 2.4 GHz spectrum is creating serious problems of interference and/or saturation, it is doubtlessly true that congestion within infrastructured Service Sets – i.e., “cells” served by an Access Point (AP), often spoils performances, specially of applications that require low latency. Even VoIP services work well on W-LANs, but their quality drops drastically as soon as traffic on-air becomes heavy or even just moderate [1]. Service support problems arise in all WLAN environments, but most of all in public HotSpots where customers pay for the service and specifically in networks conceived to support highly variable service demands, where dimensioning is more problematic, if at all possible.

IEEE 802.11 Task Groups have been at work to propose new solutions for the improvement of the Quality of Service (QoS) and service differentiation (IEEE 802.11 TGe), and more recently with 802.11 TGn to improve the throughput of the radio interface beyond the simple increase of the transmission speed, having recognized that the actual MAC protocol claims too high a toll on the scarce resources.

802.11 TGe completed its work on July 2005. The final document is now under revision for publication as IEEE standard, and we can expect the first standard-compliant devices to be on the market as early as mid 2006. 802.11 TGn work is still in an earlier stage, but it is already clear that the MAC and management part of the outcome of this TG will build upon 802.11e, making this latter an even more interesting solution.

As we discuss shortly in Sect I-A, 802.11e defines a framework for the management of resources and traffic, but the actual algorithms and techniques used within the framework are open to competing implementations, and they can heavily affect the final performance obtained by a QAP (a QoS enabled Access Point) and the associated QSTAs (QoS enabled Stations). How much the research community is still interested — and working— on service provisioning in WLANs is testified by very recent surveys and position papers as [2], as well as scientific works discussed in Sect. I-B.

This paper addresses the problem of traffic scheduling by the QAP, discussing different possible implementations. The reference point, albeit naïve, is the simple scheduler (SS) drafted as example by the 802.11 TGe [3], which is conceived purely for CBR traffic, and obviously fails under any other condition. We propose and discuss two radically different possibilities: i) improving performances by a better characterization of the traffic, leading to open loop schedulers based on the notion of “*equivalent bandwidth*” (see Sect. II); and ii) improving performances by taking advantage of the feedback QSTAs send to the QAP describing the status of the local queue and applying closed-loop control techniques (see Sect. III).

In light of the results presented in Sect. IV, we argue that only the second choice represent a safe and robust implementation, and the overall complexity of the system, taking into account not only QAP computational requirement, but also QSTA “cooperation” and the interaction with existing applications, is indeed not larger that an open-loop solution with fixed allocation.

A. Overview and Modeling of 802.11e

The MAC protocol defined by 802.11e is the compromise between different needs: maintaining a backward compatibility, keeping the system complexity at bay, and enhancing its performance in terms of QoS support and differentiation. We refer to the Draft 11.0 [3], which is almost definitive and should not bear substantial differences with respect to the version submitted for Standard approval in July 2005.

The MAC protocol blends together an enhanced, QoS enabled version of the CSMA/CA used in the Distributed

This work was supported by the Italian Ministry for University and Research (MIUR) under the PRIN project TWELVE (http://twelve.unitn.it)

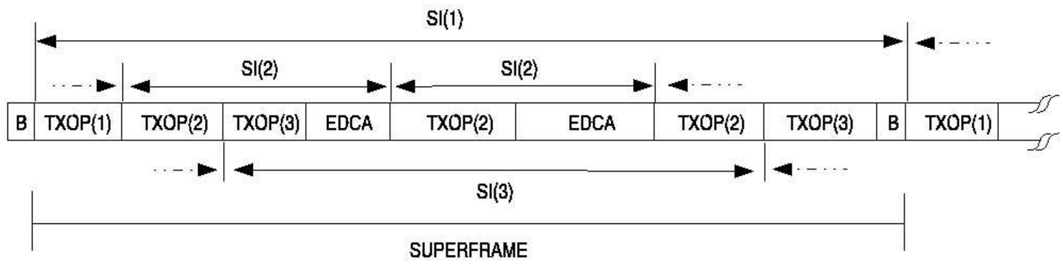

Fig. 1. General organization of the MAC protocol in 802.11e

Coordination Function (DCF) of 802.11, and named EDCA (Enhanced Distributed Channel Access), and a polling-based access named HCCA (HCF Controlled Channel Access[1]).

A plethora of solutions has been presented in recent years to differentiate services both in EDCA and HCCA. In this work we focus on scheduling within HCCA only, assuming that the EDCA period is used only for legacy stations, and is not used for QoS support or service differentiation. In Sect. I-B we discuss the literature that is most related with our work.

Fig. 1 describes the general evolution in time of the access procedure. The time is organized in superframes, which corresponds to the Beacon Interval (BI), i.e., the time between the transmission of two subsequent beacon frames. A superframe is divided into a contention free period (CFP), managed via HCCA, followed by a contention period (CP), where EDCA is used. The HCF, however, can interrupt the CP with periods, named Controlled Access Periods (CAP) when HCCA is used and the access is again contention free.

The HCF assigns resources to stations by allotting time in a Service Period (SP). An SP includes the time needed for the QAP to transmit frames to the QSTA and the Transmission Opportunity or TXOP for the QSTA. As shown in Fig. 2 TXOP assignment is done via polling. We identify the transmission opportunity of the i-th QSTA ad TXOP(i) and the service period for the same QSTA as SP(i). TXOP(i) represents the maximum time QSTA i can use the channel, including management and control frames. A QSTA that has nothing to transmit while its TXOP is still not expired should end it by transmitting a null frame.

QSTAs communicate their needs to the QAP on a per-connection basis (Traffic Stream – TS) and each QSTA can have up to eight parallel TSs.[2] The signaling and negotiation is done through the exchange of Traffic SPECifications (TSPEC), that contain (among others) the parameters relevant for traffic scheduling described in Table I. Besides the TSPEC, a QSTA sends the status of its internal queue to the QAP in the header of each data packet it transmits on-air. It is immediate to notice inspecting Fig. 1 and Table I that there are a number of open issues which must be addressed in implementing the HCF. For instance TXOPs are normally assigned on a per-station basis to spare resources wasted by multiple pollings to the same QSTA, while TSPEC are negotiated on a per-flow basis.

TSPEC parameter	Description
Nominal MSDU Size	Length (in octets) of MSDUs. One bit of this field indicates if MSDU size is constant or must be considered as "nominal"
Maximum MSDU Size	Maximum MSDU length in octets. The QAP should grant time to transmit at least one Maximum size MSDU per TXOP
Maximum Service Interval	Maximum admitted time between two consecutive polls (SI)
Minimum Data Rate	The scheduler is expected to allocate at least the time needed to serve this data rate
Mean Data Rate	The average traffic the TS generates
Peak Data Rate	The maximum data rate of the stream
Burst Size	Maximum amount of data that can arrive to the MAC-SAP with Peak Data Rate
Delay Bound	Maximum time a MSDU is allowed to queue before being successfully transmitted

TABLE I

TSPEC PARAMETERS RELEVANT FOR TRAFFIC SCHEDULING

QSTAs need not be polled in round robin, provided that consecutive pollings to the same station are not spaced more than an upper limit known as maximum Service Interval (SI) and negotiated between the QSTA and HCF. The service interval SI(i) of station i is the result of the different requirements of its traffic streams, but the HCF can also set it to a smaller value than needed, for instance to optimize the polling cycle. A very simple way to meet the requirements (though not necessarily the best in terms of performances) is setting all the SI identical one another and choose as common polling time the largest submultiple of the Beacon Interval which is smaller than the smallest required SI. This is the choice done in the simple scheduler SS.

Let k be a discrete time parameter indexing the successive polling cycles (PC). We define: $r_a(k)$ the vector representing the resources available during the k-th PC; $\overline{r}_i$ the mean (average) resources assigned to the i-th QSTA; $r_i(k)$ the resources actually assigned to the i-th QSTA during the k-th PC.

Notice that representing the system as a discrete time system there is no need to have all SI(i) equal one another: if station i is not polled at (discrete) time k, then $r_i(k) = 0$.

We assume that there is an admission control function that ensures stability of the system, so that for any meaningful

[1]HCF or Hybrid Coordination Function is the QAP entity that controls and manages the resources within the "cell."

[2]In the following we use the terms TS, connection, and flow interchangeably.

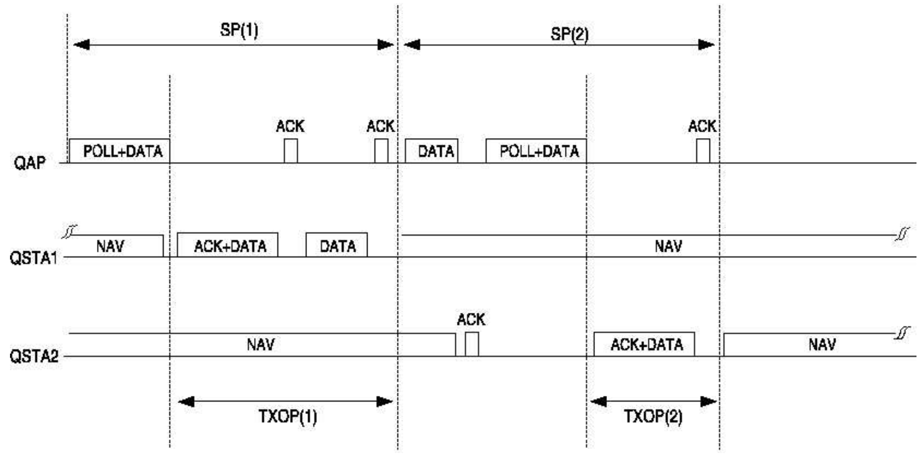

Fig. 2. Polling interaction between HCF and QSTA

observation interval K, given the number of associated QSTAs N_{QS}, the relationship

$$\frac{1}{K}\sum_{k=1}^{K} r_a(k) > \sum_{i=1}^{N_{\text{QS}}} \overline{r}_i \tag{1}$$

holds and $\overline{r}_i$ can also be seen as the *guaranteed* resources assigned to station i. $r_a(k)$ is not necessarily constant over k, since PCs can have different length, due to non constant SIs, or to varying resources reserved for EDCA access (e.g., a non-QoS enabled STA associates or leave the QAP), etc.

From a scheduling point of view, we can assume that the 802.11e framework can be represented as in Fig. 3, where the the system a) represent the case of open-loop scheduling, and the system b) the case of closed-loop scheduling. The only difference between an open-loop and a closed-loop scheduler is whether the resource assignment function takes into account the remaining backlog of QSTAs $\text{bl}_i(k)$ at the end of the k-th SI or not. The block D_1 is a one-step delay representing the fact that the schedule defined by HCF during the k-th PC will be implemented by QSTAs during the next PC.

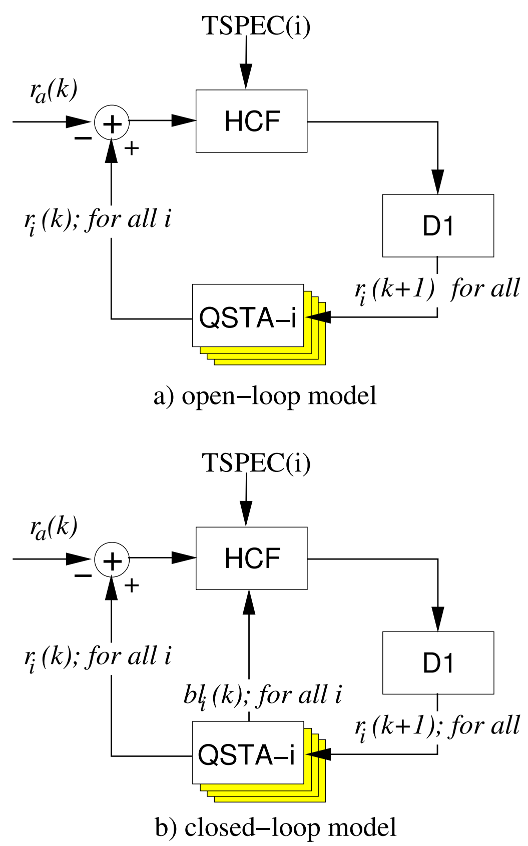

Fig. 3. Model of 802.11e scheduling: a) without using the feedback from QSTAs; b) using the feedback from QSTAs

B. Related Work

There are many different proposals for managing the scheduling of resources in 802.11; we just mention here those that are closer to our approach, leaving aside all proposals centered on EDCA mechanisms.

The work in [4] is probably the closer to our work. The authors use a continuous time modeling, which is very accurate in analyzing performances, but is less prone of future optimization applying control-theoretical results.

The papers [5] 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 QSTAa feedbacks. Similar is also the approach described in [7], where an open-loop predictive scheduler is proposed. The prediction algorithm is based on measures of the actual traffic sent by TSs.

Compared to all these works, the approach proposed in Sect. III is characterized by a faster dynamics of the closed loop scheduling adaptation, that leads to smaller delays in the channel access.

II. Equivalent Bandwidth Scheduling

The concept of "*Equivalent Bandwidth*" (EB) is not new. Indeed it dates back to the first CAC studies on VBR traffic.

The basic idea is quite simple. Given the stochastic characterization of an VBR source (e.g. a voice codec with Voice Activity Detection –VAD– and silence suppression) the EB of the source is the amount of resources r_{EB} that must be reserved to the source in order to guarantee that the amount of traffic T^s generated by the source will not exceed r_{EB} with a given probability p_o. Formally, given a source i

$$r_{\text{EB}}(i) : \mathbb{P}[T^s(i) > r_{\text{EB}}(i)] < p_o(i) \tag{2}$$

On the one hand, implementing an EB scheduler within the context of 802.11e scheduling is extremely easy. In fact, it is sufficient that the QSTA negotiate r_{EB} instead of the average bit rate in the reference SS of the standard draft and the QAP will regularly allocate the required resources. If the traffic generated by the flow is smaller than the negotiated r_{EB}, then the QSTA terminates TXOP(i) with the null frame and the spared resources are automatically freed for other stations' use. On the other hand, the main drawback of the equivalent bandwidth is not solved at all: if the QSTA traffic characterization is not precise, then r_{EB} will not meet the requirements of (2). Indeed, even if the source characterization is good, but it does not have Markovian properties, finding r_{EB} may not be an easy task. Finally, resource assignment in 802.11e is heavily quantized, since MPDU fragmentation is deprecated and inefficient. Quantization implies that the resources assigned will not match exactly the equivalent bandwidth of the flow, resulting in additional impairments.

We implemented (in ns-2) VBR sources as two- and three-states stochastic chains in order to to control the effect of approximated traffic characterization on the scheduler performance: if the transitions probabilities between chain states are geometric, then the chains are Markovian and (2) can be computed exactly. In most other cases (e.g., heavy tailed dwelling times, or even simple constant dwelling times) the equivalent bandwidth of the source is an approximation, and we expect the scheduler to have a worse performance in one way or the other, i.e., not meeting the requirements of the flow or performing poorly in accepting TSs. Fig 4 reports the discrete time chains implemented in ns-2. For each state s the user can choose the bit rate $R_b(s)$, the average frame dimension $D_f(s)$ and their distribution (e.g., constant, or truncated negative exponential), and the frame interarrival distribution.

The actual implementation is in form of a Discrete Time Chain with transitions upon packet transmission. The transition probabilities P_{ij} from state i to state j describe the behavior of the source every time a packet is generated. The transitions probabilities are computed so as to respect the average transmission rate of the source. For the three-state source some additional constraints may be required in order to have a unique solution.

III. Closed Loop Scheduling

As discussed in Sect. I-A QSTAs transmit their buffer level to the QAP. At the same time, the QAP is perfectly aware of

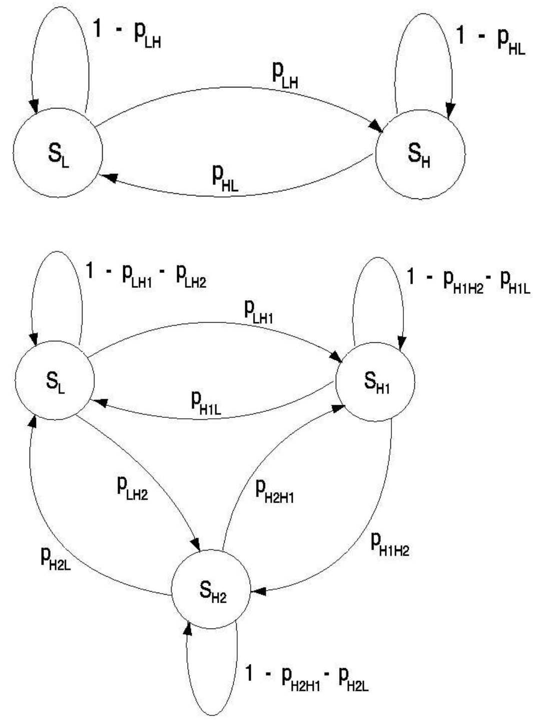

Fig. 4. Two- and three-state VBR models implemented in ns-2

the amount of traffic that has to be transmitted at the beginning of each SP to every QSTA.

Let's assume that the QAP knows the PHY transmission rates of QSTA, which is not unreasonable, since it can be assumed that transmission rates do not change from one SI to the next, so that the occasional transition from one rate to another can be accounted for as a disturbance in the control loop and it will be compensated automatically. In this situation the resource assignment can be done in bytes, and transformed in time to assign TXOPs taking into account the timing of the protocol, only at the end. Without loss of generality we can assume that the schedule is computed on a per-TS basis, and then the assignment is done on a per-QSTA basis in order to reduce the polling overhead.

The evolution of the transmission buffer B_j for each flow j is

$$B_j(k+1) = \max\left(0, [B_j(k) - r_j(k+1) + T^s_j(k+1)]\right)$$

If $B_j(k+1) > B^C_j$, where B^C_j is the buffer capacity for flow j, then $B_j(k+1) - B^C_j$ information is lost and $B_j(k+1) = B^C_j$. As stated with (1) we assume that the system is stable, so that the goal of any control technique that regulates r_j reduces to the minimization of some given metric $\mathcal{H}$ of the evolution in time of all the flows' buffers; first of all the loss probability.

There are many optimal control techniques that, given the metric $\mathcal{H}$ (e.g., $\mathcal{H}_2$ or $\mathcal{H}_\infty$) and the stochastic properties of the T^ss define the scheduling algorithm (see for instance [8]). At this stage of the research, however, we're more concerned with the fundamental properties of closed-loop scheduling, rather than finding a theoretical optimal scheduler, which may turn out to be computationally complex, or loose its optimality properties due to implementation impairments. Therefore we resort to a basic positional controller (P-control) that tries to assign resources proportionally to the backlog. Since there is a guaranteed assignment to all flows, the assigned resources to flow j are

$$r_j(k) = r_j^{\min}(k) + r_j^{+}(k)$$

where $r_j^{+}(k)$ is a non-negative amount of additional resources assigned to flow j based on any weighted proportional function of the backlogs $B_j(k-1)$. Notice that $r_j^{\min}$ is a function of the discrete time k since SIs may be non constant.

Since we are dealing with a system that is intrinsically stable and inf-clipped (the buffer cannot be negative), we do not have to worry about stability of the controller, so that the repartition of resources among flows can also lead to a controller gain larger than one without affecting stability of the system.

A max-min fair proportional scheduler assigns additional resources based on the following proportionality

$$r_j^{+}(k+1) = \beta_j \frac{B_j(k)}{\sum_{j=1}^{N_{\text{TS}}} B_j(k)} \left[r_a(k) - \sum_{j=1}^{N_{\text{TS}}} r_j^{\min}(k) \right] \quad (3)$$

where β_j is a coefficient that can take into account flow priorities or any other differentiation policy within a given traffic class. The apportioning coefficient β_j can also be used to properly "weight," flows with different mean rates, since draining the same amount of information from different buffers takes a time which depends on the draining rate, so that the same backlog can result in different queuing delays. N_{TS} is the number of TS admitted by the CAC function.

Traffic classes can be easily taken into account either with a fixed priority scheme (i.e., the highest priority class is assigned additional resources first, then leftover resources are assigned to the second class, and so on) or with any other priority scheme that leads to the computation of r_j^{+} for the given class. Deriving the overall assignment scheme with a given priority enforcement is just cumbersome and is not reported here for the sake of clarity.

It must be noted that uplink and downlink flows are not identical, since the backlog information of uplink flows is one polling cycle old, because it reflects the situation at the QSTA when the last frame of the flow was transmitted on air, while the backlog of downlink flows can be known at the QAP without delays. We do not consider this problem anymore in this work, assuming that it does not introduce any meaningful bias.

The max-min based resource assignment (possibly distorted by the weights β_j that include the different mean rates), is based on the following considerations:

- The system is stable and backlogs are due to statistic fluctuations of the traffic sources (voice with VAD, video, etc.);
- The larger the backlog, the larger is the delay imposed to the the waiting information, and the larger is the probability that the flow buffer will overflow;
- If nothing is known about the stochastic processes driving the buffers, then the information loss probability is minimized when the buffers are all equalized.

A. *Closed-Loop scheduling with fixed* SI

The 802.11e draft [3] recommends using a fixed polling time for each station. Notice that, as depicted in Fig. 2, this does not necessarily mean that every SI is identical, but only that the same station is polled at fixed time intervals.

Implementing the proportional max-min fair P-controller defined by (3) with fixed SIs is straightforward and does not require any additional explanation[3].

We call this scheduler '*MaxMin Fair–Adaptive*' or MMF-A.

B. *Closed-Loop scheduling with dynamic oversampling*

The implementation described in Sect. III-A is perfectly complying with the draft. However, as already noted in [4], the reaction time of a closed loop scheduler with fixed SI can be as large as 2SI even for very low load conditions, which might penalize VBR sources, specially if sources have high variability. As we discuss presenting the results, the trivial solution of reducing SI is not practical, because reducing SI increases the overheads, thus penalizing the network under heavy load.

One possible solution is using dynamic SI values. This is not explicitly admitted by [3] (albeit neither explicitly forbidden), but we'll see that may significantly increase performances under rather normal operating conditions.

The discrete time theoretical framework depicted earlier in this paper remains identical also if SIs are changed dynamically. The problem is finding a way of changing SI dynamically. Fortunately, releasing the requirement of polling intervals to be constant, the solution is easy: if the k-th CFP ends at time $t(k)$ before a deadline $\tau(k)$ that defines a minimum guaranteed EDCA period before the next CFP, then the resources relative to the the time interval $\tau(k) - t(k)$ can be re-assigned with a new CAP, which defines a dynamic 'oversampling' of the controlled system. When the traffic fluctuations temporarily bring the system in overload, then $t(k) = \tau(k)$ and the normal SI intervals are used, so that global efficiency is preserved; when the system is not overloaded, but a few QSTA offer more traffic than their guaranteed share, the possibility of immediately re-assigning resources reduces the queue length, but most of all increases the probability that a frame will not violate the flow delay bound.

Since $\tau(k) - t(k)$ is normally rather small, assigning resources proportionally to the queue length is not possible due

[3]Indeed, the actual implementation requires a great deal of care to fulfill all standard draft requirements, and also due to the fact that resource assignments are quantized. These are however cumbersome details that do not add much to the fundamental idea explained so far. We refer the interested reader directly to the ns-2 implementation available on the TWELVE project website under the 'tools' menu (twelve.unitn.it/tools.html) — '802.11e closed-loop scheduling' — both for this scheduler and for all the others mentioned in this paper.

to quantization, so we decided to assign all of them to the source i for which

$$\max_{1<j<N_{\mathrm{TS}}}\left(\beta_j \frac{B_j(k)}{\sum_{j=1}^{N_{\mathrm{TS}}} B_j(k)}\right)$$

up to the equalization of its buffer with the one immediately smaller (in case of multiple stations with equal backlog values the one which is first in the polling schedule is selected).

We call this scheduler '*MaxMin Fair–Adaptive with Re-scheduling*' or MMF-AR.

Concluding this discussion of closed-loop scheduling, let's consider best effort traffic. It is common idea that the EDCA access scheme in 802.11 is the best one in supporting TCP-based best effort traffic, since polling schemes tend to be too rigid to adapt to the fast variability of best effort traffic. Indeed a closed-loop scheduler that reacts quickly to the presence of additional best effort traffic would spare the resources spent in collisions and backoffs of the EDCA protocol. There is however a major difference with respect to guaranteed traffic. In presence of greedy best effort sources, the stability of the system is not guaranteed with the traditional control-theory meaning, and the assignment of resources based on a P-controller of the source buffer obviously results in heavy unfairness. An example helps visualizing the situation. If two sources compete, but one starts before the other, say s_1 starts before s_2, then the transmission window of s_1 is much larger than the transmission window of s_2 when this latter starts transmitting. The buffer level at the QSTA will reflect the dimension of TCP transmission window. Any assignment scheme proportional to the buffer size, will favor s_1 and maintain the unfairness in time, unless there are losses and TCP reduces the congestion window size.

Indeed, to correct this bias, it is sufficient to apply a counting function

$$U_{B_i} = \begin{cases} 1 & \text{if } B_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

and evenly distribute the resources among stations that have some backlog.

We are currently evaluating the performance of the closed loop schedulers applied to best effort traffic, but the topic of best-effort traffic is not discussed further in this paper.

IV. Initial Results

We have implemented the schedulers and the VBR sources discussed in previous Sections in ns-2 [9] to evaluate the performance of our proposal. We consider five different possible schedulers: i) SS – the draft simple scheduler conceived for CBR traffic only; EB(0.2) and EB(0.01) – the same scheduler, but applied to TSPECs that reflect an ideal computation of the equivalent bandwidth of the VBR sources; MMF-A – the closed-loop scheduler with constant SI; MMF-AR – the closed loop scheduler with re-scheduling. The simple, open-loop scheduler allocates CFP and CAPs based on fixed SI

Mean bit rate = 128 kbit/s						
State	Param.	Value	Param.	Value	Param.	Value
L	R_b	64 kbit/s	D_f	120 bytes	T_{dw}	2.38 s
H	R_b	640 kbit/s	D_f	1200 bytes	T_{dw}	0.03 s

TABLE II

Parameters characterizing the two-state VBR source in the simulations; T_{dw} is the average dwell time in the state

Mean bit rate = 128 kbit/s						
State	Param.	Value	Param.	Value	Param.	Value
L	R_b	64 kbit/s	D_f	120 bytes	T_{dw}	0.36 s
$H1$	R_b	640 kbit/s	D_f	120 bytes	T_{dw}	0.009 s
$H2$	R_b	640 kbit/s	D_f	1200 bytes	T_{dw}	0.021 s

TABLE III

Parameters characterizing the three-state VBR source used in the simulations; T_{dw} is the average dwell time in the state

intervals equal for all stations. The allocation is based only on the mean rate parameter of the TSPECs.

We have used both two- and three-state VBR sources, whose characterizing parameters are summarized in Tables II and III respectively. T_{dw} is the average time spent in the relative state.

For the sake of easy we only report results for homogeneous VBR sources competing for the uplink, leaving more complex scenarios including best effort traffic for further research. In this situation SI can be identical for all sources and we set it to 50 ms. Simulations have been run for 200 s of network operation or more.

Fig. 5 reports the loss probability performance of the five different scheduling schemes with non-delay sensitive sources. For these sources the delay bound of frames is set to ∞ and frame losses are only due to buffer overflows. The buffer size is measured in packets and is $B^C = 50$. Notice that this implies that the buffer size in bytes is variable depending on the size of frames it stores. Simulation points where no losses were recorded are not plotted.

The advantage of closed-loop scheduling with re-scheduling is evident in both plots, The other curves behavior is less straightforward to understand. With the simple two state sources, none of the other schedulers offer an acceptable behavior, even for highly underloaded networks where only a few flows are present. The reason lies in their impossibility to exploit unused resources, which are left for use to the EDCA access. Clearly letting all sources compete for resources during the EDCA phase would change the situation, but this is left for future research. MMF-AR, instead, re-scheduling unused resources at the end of polling cycle, is able to provide much better performance.

The results relative to the average transmission delay of packets, reported in Fig. 6, confirms the results, with the MMF-AR scheduler consistently obtaining lower transmission delays with respect to the others. We point out once more that fragmentation is inhibited in our results, and allowing fragmentation may change some results, allowing stations to exploit parts of the TXOPs where the whole packet to be

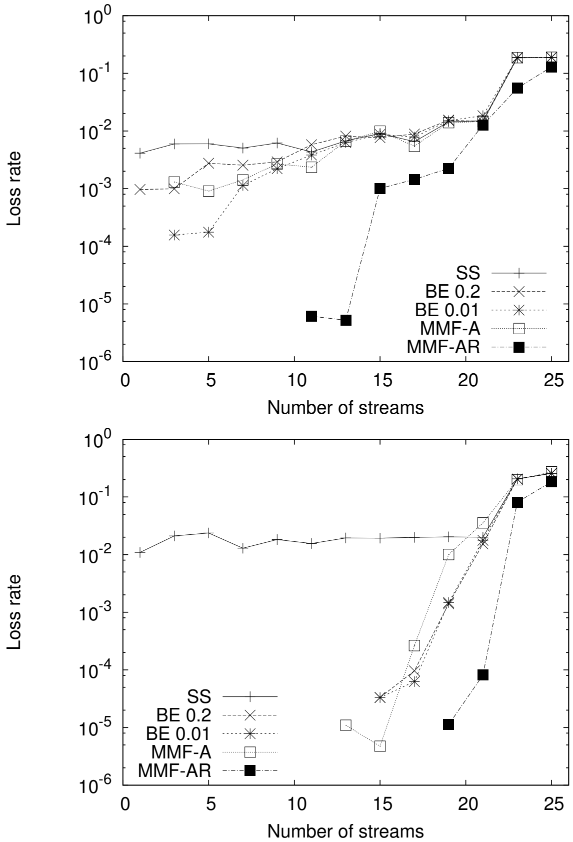

Fig. 5. Packet loss probability for the two-state (upper plot) and three-state (lower plot) VBR sources as a function of the number of concurrent flows for the five different considered schedulers for non-delay sensitive sources

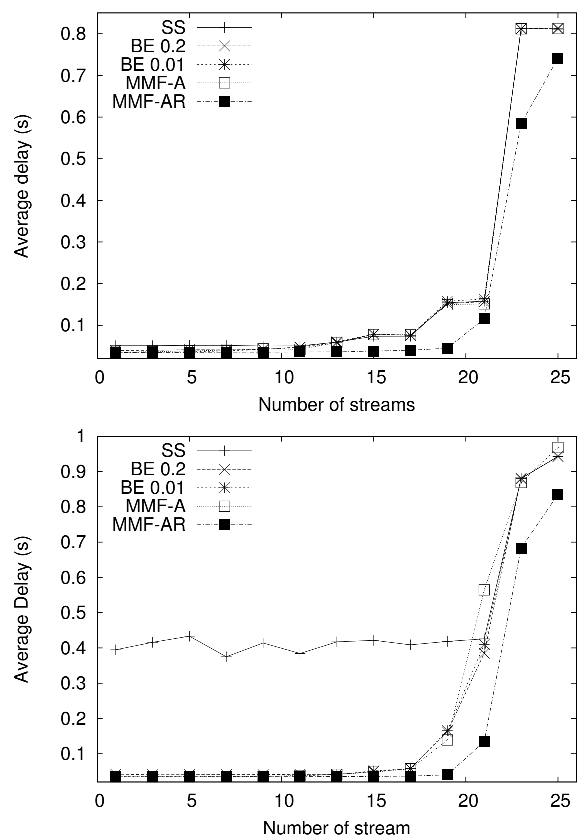

Fig. 6. Average frame delay for the two-state (upper plot) and three-state (lower plot) VBR sources as a function of the number of concurrent flows for the five different considered schedulers for non-delay sensitive sources

transmitted cannot be fit. However, we deem that finding a scheduler that performs well with variable size frames without requiring fragmentation is a major achievement, since fragmentation introduces overhead (in transmission and processing), and requires that both the sender and the receiver supports it.

We now restrict to study the more complex three-states sources for the sake of brevity. Fig. 7 refers to delay-sensitive sources, whose packets must be transmitted within 100 ms or they are discarded. This delay bound can be typical for voice of video-conferencing applications.

The behavior in this case is more easily interpreted. The SS and EB(0.2) schedulers make sources loose frames even in non-congested situations, simply because the resources allocated are fixed, and the VBR sources exceeds them. The loss rate in these cases can be theoretically computed starting from the VBR sources characteristics and the delay bound, and this computation confirms the simulation results. BE(0.01) and MMF-A behaves similarly, due to the delay in reaction of MMF-A, while the MMF-AR scheduler, reassigning unused resources to those flows that are currently above average obtains a performance that can be orders of magnitude better than the other schedulers. We point out here that the quantization of resources in the MMF-A and MMF-AR schedulers are slightly different and the one implemented

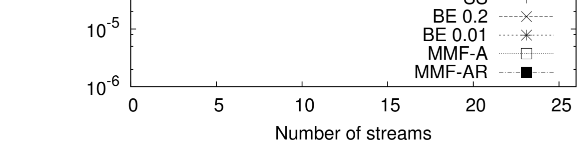

Fig. 7. Packet loss probability for the three-state VBR sources as a function of the number of concurrent flows for the five different considered schedulers for delay sensitive sources

in MMF-A is less efficient with large packets and high loads, which leads to the very bad behavior between 16 and 21 stations. We investigate this behavior in more detail at the end of the paper using real video sources.

As we already mentioned, the trivial solution to improve the performance of delay-sensitive traffic may seem the reduction of the SI. Fig. 8 reports the results with the same traffic configuration as Fig. 7, but with SI $=$ 25 ms. As correctly

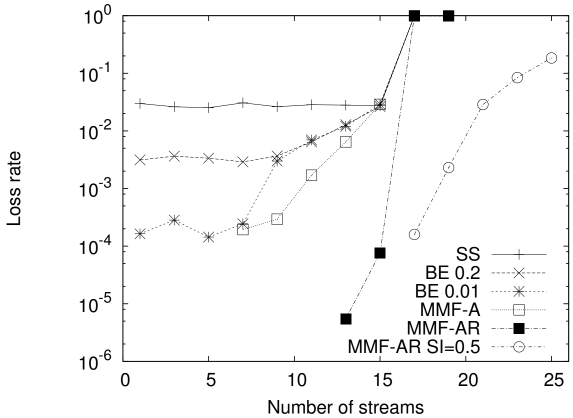

Fig. 8. Packet loss probability for the three-state VBR sources as a function of the number of concurrent flows for the five different considered schedulers for delay sensitive sources with SI= 25 ms

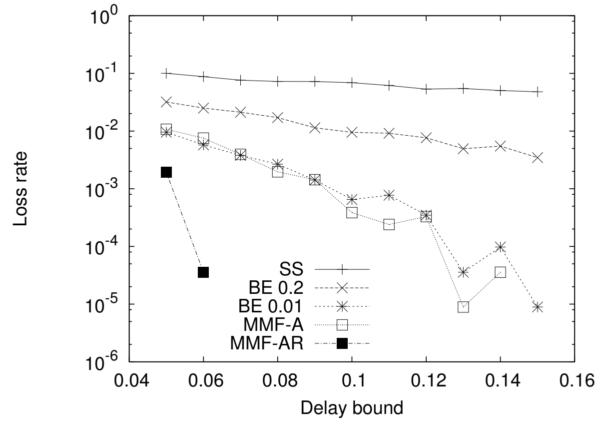

Fig. 10. Packet loss probability for the three-state VBR sources as a function the delay bound of frames with 8 concurrent flows for the five different considered schedulers

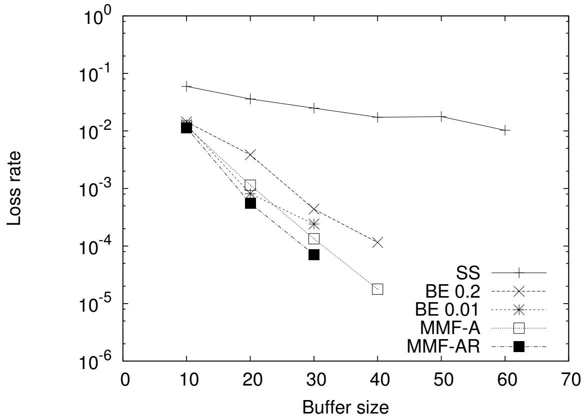

Fig. 9. Packet loss probability for the three-state VBR sources as a function the QSTA buffer size with 8 concurrent flows for the five different considered schedulers

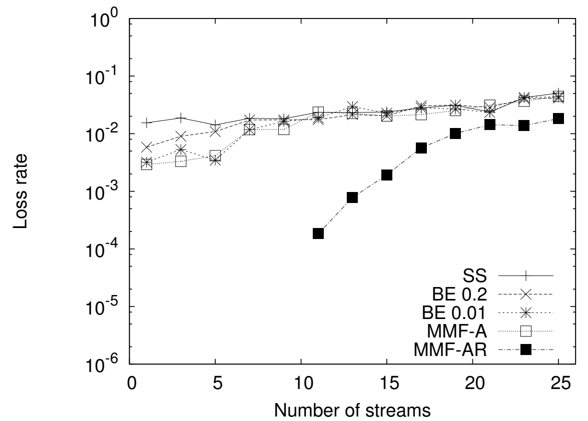

Fig. 11. Packet loss probability for the three-state VBR sources as a function the number of flows when the sources behaves differently from the declared TSPEC; non delay-sensitive sources

predicted in the analysis, the overheads of a shorter SI are dominant and it is very difficult to see any improvement, apart for very low loads. Indeed, comparing the two figures, it can be seen that all schedulers provide a smaller loss rate for very low loads, but as soon as the load increases the overheads, and the impossibility to assign the TXOPs to support a maximum size frame, makes the system performance unacceptable. The curve relative to MMF-AR with SI $= 50$ ms is reported in Fig. 8 for reference, showing that adapting the polling interval to the network conditions puts together the benefits of high load efficiency and low load performance.

To gain a better insight on the behavior, we analyze the sensitivity of the schedulers as a function of the buffer dimension (Fig. 9) and of the delay bound (Fig. 10). The behavior is consistent with theory, with the loss rate decreasing exponentially, but with different slopes depending on the scheduler efficiency. In both cases the MMF-AR scheduler performs consistently better than the other considered, while the MMF-A is always comparable to the EB(0.01), without requiring the complex (and unreliable) source characterization required to define the equivalent bandwidth.

We further analyze the performance in two non-standard, but realistic cases. Fig. 11 analyzes the behavior of the schedulers when the sources do not behave as declared in the TSPECs. Namely, the mean data rate is left unchanged, but the average time spent in the states is larger, so that high traffic bursts result longer. This result should be compared with Fig. 8. The performance loss of open-loop schedulers is evident, but also the MMF-A scheduler suffers, while the MMF-AR performance is far less influenced and remains acceptable. Fig. 12 explores what happens when sources are time sensitive, but the SIs supported by the QAP cannot be reduced. We set the delay bound to 1.5 the SI. All schedulers, apart from MMF-AR, have similar and unacceptable performances, with high losses even with only a few active flows. In particular MMF-A suffers from the fact that the delay bound is smaller than $2\times$SI, which is its response time.

Finally, Fig. 13 refers to simulations obtained with real video traffic[4] traces [10]. Since the stochastic descriptions

[4]The video traffic traces and the scripts to import them in ns-2 can be found at:
`http://www-tkn.ee.tu-berlin.de/research/trace/pub.html`

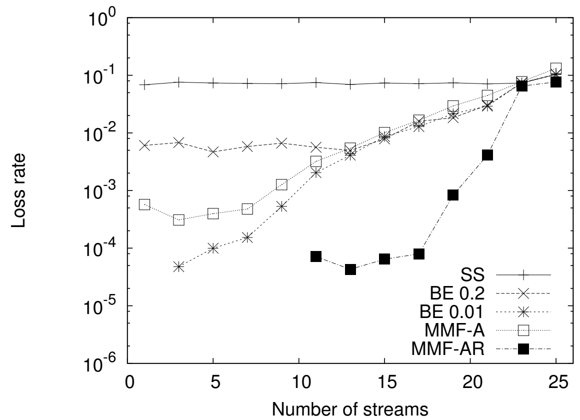

Fig. 12. Packet loss probability for the three-state VBR sources as a function of the number of flows with SI $= 100$ ms and delay bound 150 ms

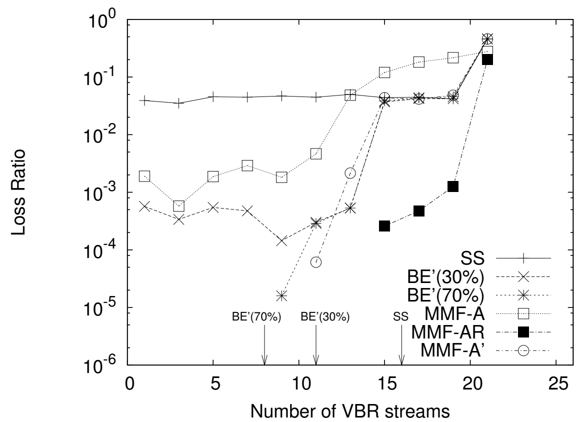

Fig. 13. Packet loss probability for real video traffic sources as a function the number of flows with SI $= 100$ ms and $DB = 150$ ms

of the sources is not available, we calculate EB' (we call it this way in order to distinguish it from EB previously defined in a more rigorous way) by considering as parameter the percentage of the difference between the peak and mean rates, and by adding such calculated value to the average rate. So, EB(0%) and EB(100%) corresponds to the mean and peak data rates respectively. The results show that SS is totally inefficient and MMF-A suffers quantization problems in adapting resources, due to the large average size of the MSDUs, while MMF-AR experiences better performances than all the others schedulers.

The curve labeled MMF-A' refers to the MMF-A scheduler with the quantization adopted by the MMF-AR scheduler. This enables to appreciate the behavior due to quantization phenomena from the one due to the SI flexibility.

The arrows labeled BE'(70%), BE'(30%), and SS reported on the x axis, indicate the CAC thresholds (resources nominally saturated) for the three open-loop schedulers. It is clear that BE'(30%) can guarantee the QoS, but only at the expenses of very low resources utilization. Finding a CAC surface for the closed-loop schedulers might be a difficult task; however, it is clear that simply using the declared mean data rate as in the SS scheduler guarantee a very good compromise between performance and resource utilization for the MMF-AR scheduler.

V. Discussion and Conclusions

This paper discussed different scheduler implementations to support QoS in 802.11e networks. In particular we argued about the use of open-loop, i.e., static and using only TSPEC information, or closed-loop, i.e., dynamic, using also information send back form the QSTAs schedulers.

Additionally we considered the possibility of using non-constant service intervals, showing that a close-loop scheduler with dynamic polling intervals trying to assign spare resources based on a P-controller on the buffer size can outperform other schedulers, including schedulers based on an Equivalent Bandwidth approach, and a closed-loop P-controller scheduler without dynamic polling.

The results we presented are not definitive, and additional research, for instance applying optimal control techniques or non-linear control techniques, is needed before the ideal 802.11e scheduler is found.

References

[1] Yang Xiao, Haizhon Li, "Voice and Video Transmissions with Global Data Parameter Control for the IEEE 802.11e Enhance Distributed Channel Access," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 1041–1053, Nov. 2004.

[2] N. Ramos, D. Panigrahi, S. Dey, "Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions," *IEEE Network*, pp. 14–20, July/August 2005

[3] "Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," IEEE Draft Std 802.11e, D11.0, Oct. 2004.

[4] P. Ansel, Q. Ni, T. Turletti, "FHCF: An Efficient Scheduling Scheme for IEEE 802.11e," to appear in ACM/Kluwer MONET journal, Special issue devoted to WiOpt'04, 2005; Ext. version available as "FHCF: A Fair Scheduling Scheme for 802.11e WLAN," INRIA Research Report No. 4883, Sophia Antipolis, France, July 2003.

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

[6] D. Skyrianoglou, N. Passas, "Traffic Scheduling in IEEE 802.11e Networks Based on actual Requirements," In *Radio Network Management '04 (RNM'04)*, Athens, Greece, 2004.

[7] Lu Yang, "P-HCCA: A New Scheme for Real-time Traffic with QoS in IEEE 802.11e Based Networks," APAN Network Research Workshop 2004

[8] K. Zhou, J. C. Doyle, K. Glover, *Robust and Optimal Control*, Prentice Hall, Englewood Cliff, NJ, 1996.

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

[10] Frank H.P. Fitzek, M. Reisslein, *MPEG-4 and H.263 Video Traces for Network Performance Evaluation (Extended Version)*, TU Berlin Dept. of Electrical Engineering, Telecommunication Networks Group, Technical Report: TKN-00-06, October 2000.