Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Control and Optimisation of HCCA 802.11e Access Scheduling

L. Palopoli, R. Lo Cigno, A. Colombo
DIT – University or Trento, Italy

Abstract— **This paper presents an optimal controller for the scheduling of real time traffic in 802.11e Wireless LANs. The problem is defined as a standard control problem, starting from a discrete event model of the systems composed by the Access Point and and the connected stations and is by Model Predictive Control scheme, for which we provide formal proofs of stability. The results are compared with an existing alternative proposal with lower complexity, which in certain conditions may exhibit limit cycles.**

The paper is closed with some realistic simulation results obtained with ns-2, which show that the proposed controller performs better than other proposals also in presence of real systems packet quantisation, which is not directly considered in the model.

I. Introduction

Access control in shared/broadcast media is an active research area both from the networking point of view and from the control theory one. Specifically 802.11 W-LANs are becoming as ubiquitous as cellular networks in places where access to the Internet is demanded, and the efficient use of their resources is of the utmost importance.

A single, broadcast access channel with resources dynamically varying between 1–2 and 15–20 Mbit/s depending on the selected physical standard (802.11b/g/a/h) and the local transmission conditions (distance between transmitter and receiver, propagation environment, noise and interference, etc.) is used by and Access Point (AP) to support several users and different services (e.g., voice, video, web-browsing, e-mail). Under these conditions, congestion often arises spoiling performance, specially for applications that require low latency such as VoIP applications. An interesting observation is that, using the standard 802.11 Medium Access Control (MAC) protocol, performance drastically degrades even with moderate traffic [16].

In late 2005, IEEE published the 802.11e standard amendment [9] defining a framework for QoS support. As we shortly discuss in Sect II, 802.11e defines the protocols for traffic and resource management, but the actual algorithms implemented in APs to support QoS 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). The importance of this topic can be understood reading such surveys as [11].

In this paper we consider a scenario in which a QAP shares the utilisation of a channel among a set of QSTA that produce real-time traffic. While the *distributed* management of 802.11 received enormous attention by the scientific community, this type of *centralised* approach, though included in the standard, was somewhat disregarded. A first simple possibility is to *statically* allocate a given fraction of the channel to each station. In this case when a new station requires the channel, it specifies its required quality level (and the subsequent resource demand) and a manager decides whether or not the request can be granted based on an *admission test*. Namely, if r_j is the channel time alloted to the $j-th$ station, then a station can be admitted only if $\sum_i^N r_i \leq r_a$, where r_a is the total transmission capacity of the channel. The problem here is how much capacity should be reserved to each station. Indeed, if we choose r_j based on the average resource requirements, there could be situations when the station accumulates packets in its buffer. This effect increases the latency and can lead to packet losses. On the other hand, if we reserve the channel bandwidth based on the worst case load scenarios, the admission test would radically reduce the utilisation. Since many interesting application with QoS demands (including VoIP) are *soft real-time* and can tolerate occasional delays if packet buffering is limited, such a strict admission control would unnecessarily reduce the possible revenues. To overcome these problems, we advocate a dynamic scheme, in which a feedback controller inside the QAP adjusts the allocation of resources to the different flows based on their time-varying needs. In the recent years, dynamic scheduling mechanisms with many different flavours have been proposed (see for instance [3], [5]).

In our case, we propose to use a feedback controller that bases its decision on "sensor" measurements of the state of the system (which in our model is associated to the buffer levels). This approach was already taken in [14], where a very simple closed loop control system was proposed. This work motivated our study and it is particularly noteworthy for it is based on a formal model of the system intended as a plant to be controlled (see Eq. (1)). Another important source of inspiration for our work is [13], wherethe authors propose a continuous time model to predict the level of the buffers. In this work, we advocate a discrete event model, where the system evolution is observed at the beginning of each polling cycle. This model combines realism and simplicity, and it allows the design of a Model Predictive Controller, which overcomes important limitations of the approach proposed in [14]. Moreover, we show an efficient implementation of the algorithm and prove that the resulting closed loop system is practically stable.

Other approaches such as [8], [17] are more simplistic and the resulting feedback schemes have more of heuristic solutions than of theoretically sound control laws. On the

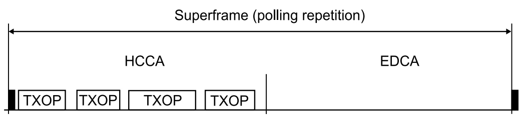

Fig. 1. General organisation of the MAC protocol in 802.11e

contrary, in the control community we can list many interesting results in related fields. For instance, a wide body of research regards congestion control of the TCP/IP protocol (see [1], [4] and references therein). An approach bearing some resemblance with the one presented in this paper is in [10], where the authors set up an optimal control problem for the rate allocation problem considering both station with QoS requirements and best effort traffic. However, in our case we deal with scheduling problems in 802.11e MAC layers, while in [10] the authors models some aspects (e.g., the Rayleigh fading) that interest cellular networks, but only marginally affect WLAN transmission channels.

The paper is organised as follows. In Sect. II we formally state the problem and show a dynamic model for the evolution of the system. In Sect. III, we formulate our Model Predictive Control problem as a particular class of linear program and show an efficient solution approach. In Sect. IV we prove the stability properties of our control scheme. In Sect. V we show some simulation results obtained for a realistic setup built using the NS2 network simulator. Finally, in Sect. VI we state our conclusion and outline the future work directions.

II. PROBLEM PRESENTATION

802.11e defines a MAC protocol whose goal is the efficient and fair sharing of resources among stations. Fig. 1 reports the general timing organisation of 802.11e: periods with controlled access —HCF Controlled Channel Access (HCCA)— and periods with contention access —Enhanced Distributed Control Access (EDCA)— are continuously alternated within superframe repetitions. During HCCA periods the QAP controls and manages the resources using a polling protocol. Our work addresses HCCA periods, which are used for the real-time traffic.

EDCA periods are simply not observable in the controlled system, and do not influence it in any way, being assigned a lower priority by the protocol. Within this framework, traffic flows are assigned resources as Transmission Opportunity or TXOP. We identify the transmission opportunity of the i-th flow as TXOP(i), which represents the maximum time flow i can use the channel, including management and control frames. A flow that has nothing to transmit while its TXOP is still not expired should end it by transmitting a null frame, thus enabling other flows to exploit the unused resources. The TXOP in our framework plays the role of an *actuator* to steer the evolution of the system.

Negotiation of the resources is done through the exchange of Traffic SPECifications (TSPEC), that contain (among others) the parameters relevant for traffic scheduling, such as the maximum and the nominal packet size, the maximum interval between two subsequent polls, etc.

Besides the TSPEC, a flow backlog (i.e., the number of bytes awaiting transmission) can be included in the header of each data packet transmitted on-air. Therefore, it can be used as a "sensor" by the QAP.

Our goal is to define a control algorithm that, at the beginning of each polling cycle, samples the sensors and decides a TXOP (i.e., the fraction of communication resource allocated to each QSTA during the polling period). Informally speaking, the objective of the controller is to keep in check the number of queued packets awaiting transmission. The importance of this goal is twofold. On the one hand, by reducing the number of queued messages we also reduce the need for buffer memory (and memory can be an expensive resource on portable, embedded devices). On the other hand, we reduce the end-to-end latency of the messages.

A. Dynamic model

The system is quantised due to packet transmission, however, we shall use a fluid approximation. In Sect. V we will show that this approximation does not compromise the performance of the algorithm.

Without loss of generality, we can build a discrete event system, where each event correspond to a resource assignment/utilisation cycle (i.e., a polling cycle). Let k be a parameter indexing the successive polling cycles (PC), N be the total number of flows admitted in the systems, and define:

- $B_j(k)$: level, or size, in bytes of the buffer of flow j at the end of PC k;
- $r_j(k)$: number of bytes allocated for transmission to flow j in PC k;
- $T_j^s(k)$: number of bytes that flow j requires to transmit during during PC k (we assume that such a request is formulated at the beginning of the PC).

The dynamics of the system is given by:

$$\begin{aligned} B_j(k+1) \quad &= \max\{0, B_j(k) - r_j(k+1) + T_j^s(k+1)\}, \\ &\text{for } j = 1, \ldots, N \end{aligned} \tag{1}$$

subject to the constraint

$$\sum_{j=1}^{N} r_j(k) \leq r_a(k) \tag{2}$$

where $r_a(k)$ are the resources available in the system during the $k-th$ polling cycle (i.e., the total availability of transmission time). From a control theoretical perspective, the vector of the buffer sizes B_j is the state variable of the system, r_j is an N dimensional command variable and T_j^s is an N dimensional exogenous disturbance term. We will henceforth refer to vector $[B_1, B_2, \ldots, B_N]^T$ as $\mathbf{B}$. Likewise, we will denote vector $[r_1, r_2, \ldots, r_N]^T$ by $\mathbf{R}$ and vector $[T_1^s, T_2^s, \ldots, T_N^s]^T$ by $\mathbf{T}^s$. Moreover, for each flow we assume that a minimum amount of transmission time $r_j^{min}(k+1)$ is negotiated. Therefore, we require, as a further constraint,

$$r_j(k+1) \geq r_j^{min}(k+1) \tag{3}$$

Both $r_a(k)$ and $r_j^{min}(k)$ are time-varying. There are at least two unavoidable reasons for this variability. The first one is the non constant sampling of the system. In fact the PCs are not necessarily equal in time provided they respect the negotiated maximum service interval, so that the amount of resources (available and to be assigned) is a function of k. The second one is more subtle. There may be different priority classes in the system, so that if the traffic load of a higher class varies, then the available resources $r_a(k)$ for the considered class change. Finally, r_j^{min} can change due to re-negotiation at the admission level. In order for the system to work properly, we assume the existence of an interval K, such that

$$\sum_{k=k_0+1}^{K} r_a(k) > \sum_{k=k_0}^{K} \sum_{i=1}^{N} T_i^s(k), \forall k_0 \in \mathbb{N}. \tag{4}$$

The above is simply a balance equation stating that the system is given enough bandwidth to send the incoming packets of data over a time horizon of length K. We do not require to have enough bandwidth to be pointwise able to exhaust the incoming packets in all polling rounds. Nor do we require to have enough bandwidth only in the average (over an infinite horizon). Instead, we require to have a sufficient bandwidth on a given time horizon of a fixed and finite length. As a motivational example, consider an MPEG streaming application. In this case, we have frames of type `I`, of type `P` and of type `B`. Frames of type `I` contain an entire (compressed) picture, whereas frames of type `P` and `B` only contain differences w.r.t. other frames. MPEG encoders use a fixed Group of Pictures (GOP) meaning a periodic sequence (e.g., `IPBPBPBPB IPBPBPB ...`), where `I` frames are typically the largest in size. Therefore, we could very likely be in a situation where some `I` frames could take more resources than available for the stream in each polling cycle, *but* the transmission time could nonetheless be sufficient over a GOP or over several GOPs.

In the sequel, we will frequently use following notations: vector $\phi(k)$ whose i-th element is defined as $\phi_i(k+1) = T_i^s(k+1) - r_i^{min}(k+1)$, and symbol $\psi(k+1) = r_a(k+1) - \sum_{j=1,\ldots,N} r_j^{min}(k+1)$.

III. Control strategies

The simple control strategy recently presented in [14] is called Max-Min-Fair adaptive (MMF-A). The control algorithm is simply the following:

$$\begin{aligned} r_j(k+1) &= r_j^{min}(k+1) + \\ &+ \tfrac{\beta_j B_j(k)}{\sum_{i=1}^{N} \beta_i B_i(k)} \left(r_a(k+1) - \textstyle\sum_{i=1}^{N} r^{min}(k+1)\right) \end{aligned} \tag{5}$$

We can see that this control law respects both Constraint (2) and (3). Moreover, it has a considerable advantage: simplicity ($r_j(k+1)$ is computed by a weighted sum). However, it also has evident shortcomings, since the nonlinearities of the system can hinder the convergence of the system to a single equilibrium point. Consider a very simple example with constant $r_a(k) = r_a \geq \sum_{j=1}^{N} T_j^s$; assume that we use the MMF-A algorithm and that $r_j^{min}(k) < T_j^s(k)$. In order for the system to have a fixed equilibrium on the origin $B(k) = 0, \forall k \geq \overline{k}$, we must have $r_j(k) = T_j^s(k), \forall k \geq \overline{k}$. However, if for channel j we have $B_j(k) = 0$, the application of control law (5) implies $r_j(k+1) = r_j^{min}(k)$, which, in turn, implies $B_j(k+1) > 0$ (in view of (1)). The possible outcome is, in this case, a limit-cycle like the one shown in Figure 3. This effect is due to the following problems:

1) MMF-A only considers the level of the buffer as an input variable, while it assumes no knowledge on the incoming flow $\mathbf{T}^s$;
2) the weight β_j of the different stations is statically chosen; at some points in time this obliges the algorithm to over-allocate/under-allocate bandwidth to a station regardless of its real needs.

As far as the first problem is concerned, even if the QAP does not know the packets each station has to transmit, it can use some kind of prediction. As an example, in the case of Voice over IP (VoIP) applications, a simplified model for the packet source is a discrete time Markov Chain with two states (voice and silence), and the probability of transition is pretty low. Therefore, the packet arrival rate can be regarded as piecewise constant signal and a predictor of the next arrival rate is given by the arrival rate during the previous polling cycle, provided that the distance between two subsequent polling cycles (Service Interval) is much larger than the dwell-time in one of the states. Generally speaking, this prediction is largely application dependent and we will henceforth assume that it is externally provided.

As for the second problem, we can attack it in different ways. The simplest one is to re-distribute the unused bandwidth amongst the stations that have back-logged packets at the end of the polling cycle. In this paper, we advocate a different approach, based on Model Predictive Control [7].

A. Model Predictive Control

The idea of the Model Predictive control is to consider the dynamic evolution of the system over an horizon of some steps and choose the control value by solving an optimisation problem. In our particular case, we optimise the ∞-norm of the buffer level $\mathbf{B}$. In order to avoid an excessive complexity, 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 optimisation problem [2]:

$$\begin{aligned} &\min_{r,h} h \\ &\quad h \geq B_i(k+1) \geq 0, \forall i \\ &\quad r_i \geq r^{min}(k+1) \forall i \\ &\quad \textstyle\sum_{j=1,\ldots,N} r_j(k+1) \leq r_a(k+1) \end{aligned} \tag{6}$$

The constraint $B_i(k+1) \geq 0$ is useful since: 1) it avoids over-allocations of bandwidth to a channel, 2) it allows us to get rid of the nonlinearity in (1) and to deal with a linear program. The problem is parametric with respect to

In: $B(k)$, $T^s(k)$, $r^{min}(k+1)$, $r_a(k+1)$
Out: $r(k+1)$

//Initialise variables
$T^s(k+1) =$ **predict**$(T_s(k))$
$\phi(k+1) = T^s(k+1) - r^{min}(k+1)$
$\psi(k+1) = r^a(k+1) - \sum_i r_i^{min}(k+1)$
finished = **false**
(a) **if** $\psi(k+1) < 0$
return **infeasible**;
(b) **if** $\sum_{j=1}^{N} B_j(k) + \phi_j(k+1)) - \psi(k+1) \leq 0$
B(k+1) = 0
else {
(c) $B(k+1) = B(k) + \phi(k+1)$
//$\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$ is an index set
$\mathcal{H} =$ **sort_increasing_order**$(B(k+1))$
$i = 1$
//Loop
while $((finished == false)$ **and** $\mathcal{H} \neq \emptyset)$ {
$B^* = \frac{\sum_{i\in\mathcal{H}}(B_i(k)+\phi_i(k+1))-\psi(k+1)}{|\mathcal{H}|}$
if $B_{h_i}(k) + \phi_{h_i}(k+1) > B^*$ {
$B_{h_j}(k+1) = B^*$ for all $h_j \in \mathcal{H}$
finished = $\mathbf{true}$
} **else**
$\mathcal{H} = \mathcal{H}\backslash\{h_i\}$;
i = i + 1;
}
}
$r(k+1) = B(k) + T^s(k+1) - B(k+1)$

Fig. 2. Optimisation algorithm

the current level $\mathbf{B}(k)$ of the buffers and to the predicted vector of the arrival rates $\mathbf{T}^s(k+1)$.

1) Solution of the optimisation problem: The problem can be equivalently written using the vector $\mathbf{B}(k+1)$ as a decision variable:

$$\begin{aligned}&\min_{B(k+1),h} h\\ &h \geq B_j(k+1)\,\forall j\\ &B_j(k+1) \geq 0\,\forall j\\ &\textstyle\sum_{i=1}^{N} B_i(k+1) \geq \sum_j (B_j(k) + \phi_j(k+1)) - \psi(k+1)\\ &B_i(k+1) \leq B_i(k) + \phi_i(k+1)\,\forall i\end{aligned} \quad (7)$$

For the sake of brevity and without loss of generality, we restrict the analysis to the case $\phi_j(k+1) > 0\,\forall j$ and $\forall k$ (the generalisation to the case $\phi_j(k+1) \leq 0$ is obvious).

An efficient method for solving this problem is the aglorithm shown in Figure 2, as stated in the following.

Theorem 1: Consider Problem (7) and assume that

$$\phi_j(k+1) > 0,\ \forall j \text{ and } \forall k.$$

The optimal solution is computed (whenever possible) by the algorithm in Figure 2.

Proof: The proof can be found in [15]. ■

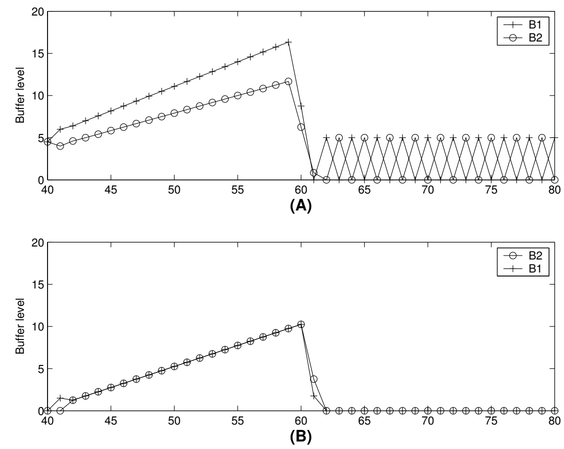

Fig. 3. Numeric example of the application of the different algorithms. (A) shows the evolution if the MMF-A scheduler is in place, (B) shows the evolution when the MPC scheduler is in place.

Remark 1: The complexity of the algorithm is equivalent to sorting an array ($O(n \log n)$). The loop in the algorithm performs a linear search (for the sake of clarity). We could as well perform a binary search (since the array is ordered) with considerable time savings in the case of a high number of connections.

B. A simple numerical example

In order to have a first idea of how the control algorithms work, we considered a simple case with two stations. The incoming packet flows $\mathbf{T}^s$, as well as the available transmission time r_a, are assumed to be piecewise constant signals. In particular, in the polling cycles between $k = 40$ and $k = 60$ the total availability of resources $r_a(k)$ is not sufficient to sustain the incoming flows. As a consequence the buffer level increases (the controller is not clearly able to compensate). On the contrary, in the interval of polling cycles between $k = 60$ and $k = 80$, the $r_a(k)$ outweighs the incoming flow $\mathbf{T}^s$. In Fig. 3, we show the evolution of the buffers if the MMF-A algorithm is used (3(A)) and if the MPC algorithm is used (3(B)). In the first case, we can see the emergence of limit cycles even when the availability of resources is sufficient to dominate the incoming flows, while in the second case the level of the buffer is reduced to 0 (at least for as long as the availability of resources is sufficient). We can also see that the MPC algorithm reduces the peak of the buffer level, in this case by a very significant 50%.

IV. PROPERTIES

In this section, we study the closed loop properties of system (1) when it is controlled by the MPC control law (6).

Let us we introduce a notion of stability that befits our application. Any notion of stability typically subsumes two different facts:

- the existence of an equilibrium,
- the attractivity of the equilibrium.

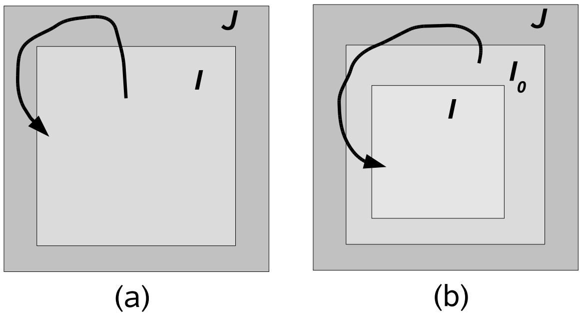

Fig. 4. Pictorial representation of our notion of practical stability. (a): $(\mathcal{I},\ \mathcal{J})$-stability, (b): $(\mathcal{I}_0,\ \mathcal{J},\ \mathcal{I})$-stability

As far as the equilibrium is concerned, the most natural notion could be requiring a fixed value (possibly zero) for the fill level of the buffers (which is the state vector of the system). However, this notion is unsuitable to our system. The reason descends from condition (4): by no means is it guaranteed that in a single step the total bandwidth $r_a(k)$ is sufficient to balance the total amount of incoming data. Therefore, at any step $\overline{k}$ such that $r_a(\overline{k}) \leq \sum_{i=1}^N T_j^s(\overline{k})$, we will necessarily have, for some j, $B_j(k+1) > B_j(k)$. This is well illustrated by Fig. 3, where even the MPC controller cannot obviously prevent the growth of the buffer between $k = 40$ and $k = 60$.

For the reason above, we introduce the following definition of stability:

Definition 1: Consider system (1). Let $\mathcal{I} \subset \mathbb{R}^N$, $\mathcal{I}_0 \subset \mathbb{R}^N$, $\mathcal{J} \subset \mathbb{R}^N$, be three sets with $\mathcal{I} \subseteq \mathcal{I}_0 \subseteq \mathcal{J}$ and assume that a control law $r(k) = r(B(k), k)$ be used. The closed loop system is said $(\mathcal{I}_0,\ \mathcal{J},\ \mathcal{I})-$ stable if $r(k) = r(B(k), k)$ complies with constraints in (2) and (3), and $\forall k_0 \in \mathbb{N}$ and $\forall B(k_0) \in \mathcal{I}_0$, there exist $H \in \mathbb{N}$ such that:

- $B(k) \in \mathcal{J}$ for $k = k_0, \ldots, k_0 + H - 1$,
- $B(k_0 + H - 1) \in \mathcal{I}$.

The definition above can be used in several directions. As a first instance, if $\mathcal{I} = \mathcal{I}_0$, the definition above implies that if at some point in time the state of the system is in $\mathcal{I}$, then the state will never leave $\mathcal{J}$ (see Fig. 4(a)). In this case we will simply speak of $(\mathcal{I}, \mathcal{J})$-stability. This concept can also be interpreted as set invariance of $\mathcal{J}$ (see [6]). If $\mathcal{I} \subset \mathcal{I}_0$, our definition implies that, starting from a state in $\mathcal{I}_0$, we have got both set invariance of $\mathcal{J}$ and attractivity of the set $\mathcal{I}$ (see Figure 4.(b)). In the rest of the section we will show the stability properties of the MPC controller using the notation $\mathcal{Q}(l) \subseteq \mathbb{R}_+^N$ to denote the set $\left\{x \in \mathbb{R}_+^N\ s.t. x_i \in [0, l]\right\}$.

A. Stability properties of the MPC controller

A result that applies to a general class of algorithms (including the one described above) is the following:

Theorem 2: Consider system (1) and assume that there exists a $\overline{k}$ such that

$$\sup_{k_0} \sum_{k=k_0+1}^{k_0+\overline{k}} \left(\sum_{j=1}^N \phi_j(k+1) - \psi(k+1) \right) < 0.$$

Let

$$\gamma = max_{h \in [1, \overline{k}]} \sup_{k_0} \sum_{k=k_0+1}^{k_0+h} \left(\sum_{j=1}^N \phi_j(k+1) - \psi(k+1) \right),$$

and I be a real positive number. The feedback scheduling algorithm generated by solving Problem (6) attains $(\mathcal{Q}(I), \mathcal{Q}(J), \mathcal{Q}(0))$-stability with $J \leq N(I + \gamma)$.

Proof: Let $S(k)$ be defined as $\sum_{i=1}^N B_i(k)$ and $\alpha(k) = \sum_{j=1}^N \phi_j(k) - \psi(k)$. We prove that any scheduling policy such that

$$S(k+1) = \begin{cases} S(k) + \alpha(k+1) & \text{if } \alpha(k) < -S(k) \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

attains $(\mathcal{Q}(I), \mathcal{Q}(J), \mathcal{Q}(0))-$ stability with $J \leq N(I + \gamma)$. This proposition implies the thesis of the theorem, since the scheduling algorithm obtained by solving Problem (6) respects the property in Equation (8).

To prove the proposition we observe the system across time intervals of $\overline{k}$ steps of duration. As a first step we prove that, assuming that $B(k) \in \mathcal{Q}(I)$, then there exists a time k such that $B(k) = 0$. Indeed, supposing that, at some point in time k_0, we have $B(k_0) \in \mathcal{Q}(I)$, then can say $S(k_0) \leq NI$. As far as $S(k_0 + \overline{k})$ is concerned, in view of Property (8), we can say that either there exists $j \leq \overline{k}$ such that $S(k_0 + j) = 0$ or $S(k_0 + \overline{k}) = S(k_0) + \sum_{k=k_0}^{k_0+\overline{k}-1} \alpha(k)$. In the first case, the proof is ended. In the second case, let: $\sup_{k_0} \sum_{k=k_0}^{k_0+\overline{k}-1} \left(\sum_{j=1}^N \phi_j(k+1) - \psi(k+1) \right) = \delta$. Under the premises of the Theorem, we can write: $S(k_0 + \overline{k}) \leq S(k_0) + \delta \leq NI + \delta$. We can iterate this argument and arrive to the conclusion that $S(k) = 0$ for $k \leq h\overline{k}$ with $h \geq I/\delta$. Clearly $S(k) = 0$ implies $B(k) = 0$.

To end the proof, we have to prove that throughout the intervals of duration $\overline{k}$, it will never happen that $B(k) \notin \mathcal{Q}(J)$. Indeed, consider any interval of indices $[k_1,\ k_2] = [(h-1)\overline{k},\ h\overline{k}]$. As a consequence of what we have shown, we have got $S(k_1) \leq I$ and $S(k_2) \leq I$. For any $K \in [k_1,\ k_2]$, $S(K)$ is given by one of the following cases:

$$S(K) = \begin{cases} \alpha(K) \\ \sum_{k_1+1}^K \alpha(k) \\ \ldots \\ S(k_1) + \sum_{k_1+1}^K \alpha(k) \end{cases}$$

where the first case occurs if $S(K-1) = 0$, the second if $S(K-2) = 0$ and $S(k-1) \neq 0$ etc. Therefore, we can upper-bound $S(K)$ as follows: $S(K) \leq I + \gamma$. The proof is ended observing that: $S(K) \leq I + \gamma$ implies $B(K) \in \mathcal{Q}(N(I+\gamma))$. ■

V. Simulation results

To the purpose of validating the MPC feedback scheduler presented in this paper in a real-life situation, we performed a set of simulations using the ns2 [12] environment. In particular, we considered a VoIP scenario where each source of information is modelled as a two states Markov chain. The specification of the two states are summarised in Table I.

State	Bit-rate	Packet size	Interarrival time	Average dwell time
0	64000 bit/s	120 bytes	0.015s	0.937 s
1	640000 bit/s	1200 bytes	0.015s	0.5 s

TABLE I

SPECIFICATION OF THE STATES IN THE CONSIDERED SOURCE MODEL.

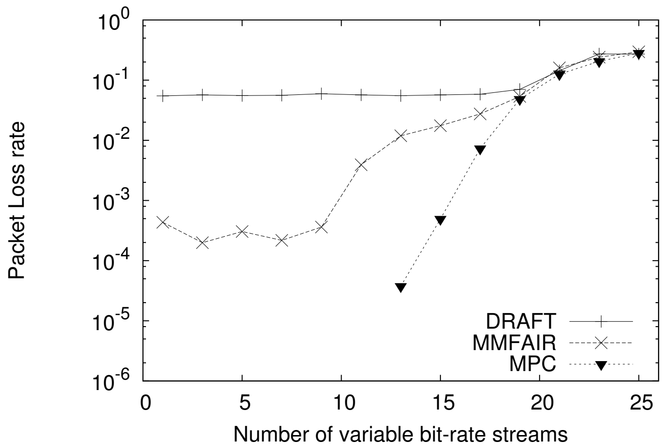

Fig. 5. Experimental results. Service interval: $50ms$, Delay bound: $100ms$.

The considered Service Interval (i.e., the periodicity of the polling cycles) was 50ms. Therefore, it is possible to assume that for about ten cycles we remain in the same state (on average). Therefore, using the last experienced value for $\mathbf{T}^s$ as a predictor is a reasonable choice. We considered a delay bound (i.e., a deadline after which a packet is dropped) of 100ms, which is typical for conversational applications. Therefore, our performance metric is only the packet loss ratio. The results are reported in Fig. 5, where we compare the packet loss ratio experienced by different algorithms for a different number of stations over a time horizon of 250s. We can see that the MPC feedback scheduler offers a significantly better performance with respect to both the reference scheduler suggested by the IEEE 802.11e standard and the MMF-A scheduler. The performance is encouraging despite the distorting effects introduced by the quantisation (the algorithm assumes a byte granularity in the allocation, but the packet size is at least two order of magnitudes greater). The beneficial effects of the algorithms are particularly evident when the system is not in sustained overload (roughly less than 18 stations); indeed, in overload the stability condition (4) is not met or is met only for very large time horizons K.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a feedback scheduling algorithm for managing traffic with QoS constraints over an IEEE802.11e wireless network. We have focused on the problem of dynamically allocating the transmission time (in the uplink direction) to different QSTA. Using a dynamic model of the system (first presented in [14]), we have set up a model predictive control problem using the level of the buffers and the predicted incoming flow as parameter. This problem can be efficiently solved and it is formally correct (with respect to the required practical stability properties). Notwithstanding the assumption of a fluid flow allocation is not realistic in many applications, the algorithm has been validated assuming a real-life situations where the packet size introduce a quantisation in the control action. As a future work, we plan to take explicitly into account the problem of quantisation and to use a stochastic variation of the model predictive approach (for those applications for which the stochastic model of the source is known to a good extent).

REFERENCES

[1] Abate A. Chen M. and Sastry S.˙ Analysis of an implementable appplication layer scheme for flow control over wireless networks. In *Proc.. of the 17th Internation Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.

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

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

[4] Wang A. and Paganini F.˙Global stability with time-delay of a primal-dual congestion control. In *IEEE Conference on Decision and Control*, December 2003.

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

[6] Blanchini F.˙Set invariance in control. *Automatica*, January 1998.

[7] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practice: a survey. *Automatica*, 25(3):335–348, 1989.

[8] Inan I. Keceli F. and Ayanoglu E. ˙ An adaptive multimedia qos scheduler for 802.11e wireless lans. In *IEEE International Conference on Communications (ICC '06)*, volume 11, pages 5623–5270, June 2006.

[9] Ieee 802.11e-2005, ieee standard for information technology–telecommunications and information exchange between systems. Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.

[10] Zafer M. and Modiano E.˙ Joint scheduling of rate-guaranteed and best-effort services over a wireless channel. In *Proc. of 44th IEEE confereence on Decision and Control and the European Control Conference*, Seville, Spain, 2005.

[11] Ramos N. Panigrahi D. and Dey S.˙Quality of service provisioning in 802.11e networks: Challenges, approaches, and future directions. *IEEE Network*, pages 14–20, July/August 2005.

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

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

[14] Larcheri P. and Lo Cigno R.˙ Scheduling in 802.11e: Open-loop or closed-loop? In *Proc. of IFIP WONS 2006*, Les Ménuires, France, January 2006.

[15] Luigi Palopoli, Renato Lo Cigno, and Alessio Colombo. Feedback scheduling in wireless lans. Technical report, Dipartimento di informatica e telecomunicazione, Trento, Italy, August 2007. DIT-07-061, Available from http://dit.unitn.it/locigno/preprints/DIT-07-061.pdf.

[16] Yang Xiao and 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*, 15(11):1041–1053, November 2004.

[17] Bin Muhamad Noh Z.A. Takahiro Suzuki, and Shuji Tasaka. A packet scheduling scheme for audio-video transmission with ieee 802.11e hcca and its application-level qos assessment. In *Proc. of Asia-Pacific Conference of Communication, 2006*, pages 1–5, August 2006.