

## Research Article

# A IEEE 802.11e HCCA Scheduler with a Reclaiming Mechanism for Multimedia Applications

**Anna Lina Ruscelli and Gabriele Cecchetti**

*TeCIP Institute, Scuola Superiore Sant'Anna, 56124 Pisa, Italy*

Correspondence should be addressed to Anna Lina Ruscelli; [a.ruscelli@sssup.it](mailto:a.ruscelli@sssup.it)

Received 31 August 2013; Revised 9 January 2014; Accepted 6 February 2014; Published 20 March 2014

Academic Editor: Mei-Ling Shyu

Copyright © 2014 A. L. Ruscelli and G. Cecchetti. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The QoS offered by the IEEE 802.11e reference scheduler is satisfactory in the case of Constant Bit Rate traffic streams, but not yet in the case of Variable Bit Rate traffic streams, whose variations stress its scheduling behavior. Despite the numerous proposed alternative schedulers with QoS, multimedia applications are looking for refined methods suitable to ensure service differentiation and dynamic update of protocol parameters. In this paper a scheduling algorithm, *Unused Time Shifting Scheduler* (UTSS), is deeply analyzed. It is designed to cooperate with a HCCA centralized real-time scheduler through the integration of a bandwidth reclaiming scheme, suitable to recover nonexhausted transmission time and assign that to the next polled stations. UTSS dynamically computes with an  $O(1)$  complexity transmission time providing an instantaneous resource overprovisioning. The theoretical analysis and the simulation results highlight that this injection of resources does not affect the admission control nor the centralized scheduler but is suitable to improve the performance of the centralized scheduler in terms of mean access delay, transmission queues length, bursts of traffic management, and packets drop rate. These positive effects are more relevant for highly variable bit rate traffic.

## 1. Introduction

Service differentiation tailored to the type of applications and stations requirements is a key issue of *Quality of Service* (QoS) provided by a network. The focus of the research on this topic is motivated by the spreading diffusion of multimedia applications that users ask the network to conveniently support. Indeed, applications like *Voice over IP* (VoIP), video streaming, video conference, and *High Digital TV* (HDTV) have different features and must be appropriately handled.

IEEE 802.11e *Medium Access Control* (MAC) *Hybrid Coordination Channel Access Function* (HCCA) [1], based on a centralized polling mechanism, has been proposed in order to introduce QoS support by means of differentiation and negotiation of stations service parameters. In particular, a new protocol parameter, *Transmission Opportunity* (TXOP), has been introduced in order to guarantee a maximum transmission time for each station, whereas the polling period is managed by the *Service Interval* (SI) parameter. The purpose of the other 802.11e MAC function, *Enhanced*

*Distributed Channel Access* (EDCA), is the same but based on *Carrier Sense Multiple Access/Collision Avoidance* mechanism and on differentiation of contention parameters.

However, since these parameters are set by the *reference scheduler* as fixed values in time and for the different stations, the network shows poor performances and the QoS management is not yet satisfactory for applications with variable bandwidth, data rate, packet size, and so forth [2–5]. Indeed, in the case of *Constant Bit Rate* (CBR) traffic, the network is yet suitable to guarantee the required service, whereas, as far as *Variable Bit Rate* (VBR) applications are concerned, it is not able to conveniently follow the traffic changes. Therefore, in order to meet the QoS applications requirements, the MAC scheduling of stations and resources must be refined by means of more specific differentiation mechanisms. The differentiation of service, in the case of IEEE 802.11e HCCA, can be based on diverse approaches, such as (1) modifying the scheduling engine to provide variable TXOP and SI; (2) adopting a simple overprovisioning method (for instance, considering worst case conditions

parameters values); (3) integrating/introducing mechanisms to tune the MAC parameters accordingly to the changing traffic profile. An example of the latter solution is the instantaneous dynamic computation of resource assignment respecting the admission control that deals with the capacity allocation constrained by the thresholds of available resources. Otherwise bandwidth reclaiming methods can be adopted: they simply recover the exceeding allocated resources, not used by some stations, and assign that to other stations, without any impact on the resource allocation performed by the admission control, but like a “legal” form of overprovisioning.

In this paper a recently proposed scheduling algorithm, *Unused Time Shifting Scheduler* (UTSS) [6], based on a simple reclaiming mechanism, is deeply analyzed. It is inspired by the consideration that in HCCA standard reference scheduler when a station has finished to transmit but it has not exhausted its assigned transmission time, that is, its TXOP, this residue of time is simply lost and the right to transmit is transferred to the next polled station. Furthermore, the resource allocation is based on the TXOP computation performed during the admission control, when the considered station asked to be associated to the *QoS Access Point* (QAP). In general, this computation takes into account mean traffic conditions, in order to have enough chances to meet stations requirements but, at the same time, avoiding expensive overprovisioning. Consequently, in the case of VBR traffic, when data rate decreases, the accorded TXOP can exceed the actually used transmission time but, if no reclaiming mechanisms are implemented, the exceeding capacity is lost. Otherwise, if it were possible to keep memory of this not spent transmission time and use that for stations with strict QoS requirements, a more efficient resources management would be possible, without affecting the overall admission control computation.

UTSS has been conceived as a cooperative centralized scheduler that can be integrated in any preexistent HCCA centralized scheduler. It simply recovers the nonexhausted portion of TXOP and allocates that to the next polled station. Thus, if this *QoS Station* (QSTA) experiences an instantaneous increase of data rate not satisfied with its TXOP, the recovered portion of transmission time can absorb the burst of traffic, without violating the QoS of other admitted stations and without eroding their accorded capacity. The behavior of UTSS will be analyzed both from the analytical and the simulation points of view; considering that, since UTSS is a supplementary mechanism that collaborates with a preexistent algorithm, each effect has to be evaluated with respect to the centralized scheduler that biases the behavior of the global scheduler. In particular, both theoretical and simulative results highlight that it does not impact on admission control and on the scheduling engine of the centralized scheduler, but that it allows a considerable improvement of network performances of the centralized scheduler in terms of access delay, throughput, packets drop rate, and transmission queues length. The analysis has been performed considering CBR and VBR traffic with increasing data rate (VoIP, video conference, video streaming, and bursty traffic) in order to reflect real world multimedia traffic and the

impact on the related TXOPs computation is investigated. In particular, the choice of VBR scenarios composed by different TSs with increasing data rate stresses UTSS just where most of the QoS schedulers proposed in literature show poor performance; moreover it highlights how and where the contribution with UTSS can be beneficial. Theoretical analysis and simulations will show that UTSS is outperforming in the case of highly variable VBR TSs whereas, in the case of CBR traffic or of traffic with low variable bite rate, its contribution is limited because the centralized QoS scheduler is suitable to efficiently serve the considered traffic streams allocating the required resource. Indeed, UTSS simply operates as a greedy algorithm that recovers unexhausted resources avoiding their waste and allocates that to the next polled station, without verifying through monitoring or statistical studies if this station really needs that. This is its flaw but also its strength since it is suitable to improve network performance without impacting on the system overhead due to  $O(1)$  computational complexity, as it will be analytically shown. Thus, even when its action is not needed, it does not overload the scheduler computation.

Furthermore, with the aim to provide a deep insight on the cooperation of UTSS with a centralized scheduler, choosing as example *Wireless Capacity-Based Scheduler* (WCBS), the obtained global scheduler has been compared with two other advanced peer methods well known in literature, *Fair HCF* (FHCF) [7] and *Real-Time HCCA* (RTH) [8].

The rest of the paper is organized as follows. In Section 2 the IEEE 802.11e HCCA protocol along with its reference scheduler are described. In Section 3 the state of arts about some scheduling algorithms with QoS, proposed as alternative to the reference one, are summarized highlighting their pros and cons, whereas in Section 4 UTSS is illustrated. Then, in Section 5 UTSS is analyzed from the theoretical point of view considering its impact on admission control and on the real-time behavior of the global scheduler; furthermore TXOP computation and management of VBR traffic like VoIP, video conference, video streaming, and bursty traffic are investigated. Section 6 reports the simulations results that confirm the analytical considerations and are corroborated by them and provides a comparison of the global scheduler against two QoS advanced methods. Finally, Section 7 draws some conclusions.

## 2. IEEE 802.11e Standard

The IEEE 802.11e standard amends the previous IEEE 802.11b with two further MAC functions, suitable to provide QoS support and coordinated by the *Hybrid Coordination Function* (HCF): the *Enhanced Distributed Channel Access* (EDCA) distributed mode and the *HCF Controlled Channel Access* (HCCA) centralized scheme.

EDCA is the evolution of the basic IEEE 802.11b *Distributed Channel Access* (DCF), where stations contend the access to the medium. EDCA adds a prioritization of service based on QoS requirements of the QSTAs that can choose between four different Access Categories. HCCA updates the

centralized polling scheme of IEEE 802.11b *Point Coordination Function* (PCF) to introduce QoS negotiation. The standard establishes EDCA and HCCA as totally compliant with the original DCF and PCF functions, ensuring compatibility with legacy devices.

Since HCCA is object of the presented study, it will be deeply analyzed in the following.

**2.1. HCCA Function.** HCCA function is managed by a QoS-aware *Hybrid Coordinator* (HC), located in the QAP, that takes action during the *Controlled Access Phase* (CAP), and that uses a polling mechanism to allocate guaranteed channel access to traffic streams, based on their QoS requirements. This scheme allows a station to transmit only if required by the QAP that sends the CF-POLL frame. In order to be included in the polling list of HC, a QSTA sends it a QoS reservation request by means of the *Add Traffic Stream* (ADDTS) QoS management frame, which collects stream information (mean data rate, mean packet size, MAC service data unit size, maximum tolerable delay, etc.). In order to implement service differentiation up to eight traffic classes, the *Traffic Streams* (TSs), with different QoS levels, are supported.

TSs parameters are collected in the *Traffic Specification* (TSPEC), whose values are negotiated between QSTA and QAP and used by the resource scheduler enabling a parameterized QoS.

**2.2. HCCA Reference Scheduler.** The IEEE 802.11e standard draws some guidelines for a nonmandatory HCCA *reference scheduler* suitable to take into account QoS requirements of QSTAs and used as reference for the design of any other HCCA scheduling algorithm. This reference scheduler deals with the computation of the main protocol parameters used to manage the access to the medium. It computes SI and TXOP as fixed values. SI is calculated as a unique value for all admitted QSTAs with the aim to globally meet their temporal service expectations. In particular, in order to ensure that each QSTA is polled at least once during the beacon interval, the scheduler establishes that SI has to be less than the beacon interval itself. Moreover, to guarantee that the polling period constraints of all traffic streams are respected, it has to be less than the minimum *Maximum Service Interval* (MSI).

Adopting a conservative approach TXOP is computed as the maximum time to transmit, at the minimum physical rate  $\Gamma_i$ , the total amount of bits enqueued during SI:

$$N_i = \left\lceil \frac{SI \cdot R_i}{L_i} \right\rceil, \quad TXOP_i = \max \left( \frac{N_i \cdot L_i}{\Gamma_i}, \frac{M_i}{\Gamma_i} \right) + O, \quad (1)$$

where  $M_i$  is the maximum *MAC Service Data Unit* (MSDU) size, that is, 2304 bytes,  $L_i$  is the nominal MSDU size,  $R_i$  is the mean data rate, and  $O$  is the transmission overhead due to interframe spaces, ACK, and CF-POLL.

As previously mentioned, the computation of TXOP and SI is necessary to distribute the available resources taking into account the QoS requirements of each station that asks to transmit. Furthermore, due to resource constraints, only

the stations that can be served with the available resources taking into account more stringent QoS needs are selected to transmit. The HCCA reference admission control test for deciding whether to admit a new stream is the following:

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=0}^k \frac{TXOP_i}{SI} \leq \frac{T - T_{CP}}{T} \leq 1, \quad (2)$$

where  $k$  is the number of yet admitted streams,  $k + 1$  indexes the newly admitted stream,  $T$  is the beacon interval, and  $T_{CP}$  is the duration of the *Contention Period*, managed by EDCA.

After the admission control phase SI and TXOP are recomputed only if a new traffic stream arrives to the station. Since their values are based on *worst case* conditions the admission control test results to be too stringent and the resource management is not optimal. Moreover, since SI is the same for all admitted QSTAs and TXOP is globally assigned to all the streams of a QSTA, all different TSs of a station are polled with the same period and are served with the same computation time. Therefore, as highlighted by numerous studies and evaluations [2–5], the reference scheduler is suitable to serve CBR traffic but unable to efficiently adapt the resource management to VBR TSs.

### 3. HCCA Scheduling Algorithms

Due to the limitations of the reference scheduler on QoS provisioning, illustrated above, many alternative scheduling algorithms have been proposed to improve the resource management [3, 9–11] and few of them are focused on the real-time support [12], that is, on temporal guarantees. In the following, to the best of our knowledge, some significant QoS schedulers are summarized. They range over different methods used to mix together efficient resource management and QoS guarantees, including queue length models, feedback-based schemes, deadlines management, integration of additional schedulers, exploitation of EDCA resources by means of the IEEE 802.11e *HCCA-EDCA Mixed Mode* (HEMM) mode that allows to jointly use both the HCCA and the EDCA MAC mechanisms, and bandwidth reclaiming methods. Moreover, for each of them we try to highlight pros and cons going deep inside into their features and performance.

The need of a variable TXOP, due to VBR traffic streams, was perceived also before the establishment of IEEE 802.11e amendment.

*Predictive-HCCA* (P-HCCA) [13] yet provided a rough mechanism to predict the mean data rate during the next polling phase and to assign a variable transmission time tailored to VBR traffic. Despite this, it does not consider the effect of this computation on the admission control feasibility test.

An example of use of traffic queues model is *Fair HCF* (FHCF) [7]. It mathematically models the uplink traffic streams queues length to estimate the global packets delay and to compute variable TXOPs with the aims to improve the fairness of both CBR and VBR traffic and the delay performances. FHCF provides a deep insight on the delay components and on queues length, modeling the queues

at the beginning of the polling phase in order to have an evaluation of the actual queues length. In particular, the packets queuing delay, influenced by the variations in packet size and data rate, is distinguished from the waiting time delay between the packet arrival time and the QSTA polling time. The investigation in the case of both CBR and VBR traffic is functional to distinguish where the reference scheduler performs well, and queues are empty since the scheduler allocated the required resources, and the case where more capacity could be needed. Being composed by two schedulers, one located at the QAP and one local to the node, FHCF is able to refine the TXOP computation. Indeed, whereas the QAP scheduler takes into account estimated and ideal queues length, the node scheduler, using its exact value, is able to recompute the needed time and redistribute the additional resources between the different TSs. Even if this complex structure impacts on system overhead and efficiency, a variable TXOP, useful in the case of VBR traffic, improves access delay.

The closed loop feedback control of *Feedback Based Dynamic Scheduler* (FBDS) [14] tries to ensure a tailored delivery of enqueued packets, limiting the maximum delay. A discrete time model of queues length at the beginning of the CAP, where HCCA is used, corrected by the actual queues length information sent by each QSTA, is used to dynamically compute TXOPs through a simple proportional controller. In particular, TXOP is assigned a value suitable to dispatch at the average depletion rate the amount of packets that can be enqueued during a CAP, whereas SIs have fixed values, as calculated during the admission control. Unfortunately the queues model and the setting of TXOP consider mean rates and MSDU size and assume the contribution of EDCA in the emptying queues, without any deeper analysis. As far as the admission control is concerned, the authors distinguish when there is network overload and when new TSs ask to be admitted to transmit. In the first case they propose to decrease all TXOPs of an amount proportional to the corresponding data rate. This solution aims not penalize stations with low data rate but does not take into account any type of QoS requirement. In the second case a new admission test is proposed that includes different TXOP values and CAP duration in the place of SI; this results in admitting a less number of streams with respect to the reference scheduler.

Instead of queues length estimation, in *Explicit Traffic Aware scheduling with Explicit Queue length Notification* (ETA-EQN) [15] QAP computes the current TXOP considering the exact queues length notification sent by the stations to the QAP at the end of their transmissions through EDCA function, (to avoid using HCCA resources). Furthermore the *TXOPLimit* standard parameter, that sets the maximum accorded TXOP considered during the admission control, is increased in order to provide instantaneous longer transmission time and to avoid excessive reduction of SI that can increase the system overhead. This is obtained by setting some unused bits of TSPEC without modifying the structure of *TXOPLimit* fields. This simple dynamic TXOP computation is suitable to increase network throughput and to reduce mean delay and packets loss rate. However, being

based on simple notifications, the proposed algorithm is agnostic of every traffic profile.

Arora et al. [16] proposed a mechanism to provide an adaptive TXOP, applicable to existing schedulers and based on the concept of link adaptation, addressed by few schedulers, to consider channel conditions. Assuming the TXOP computation provided by the centralized scheduler, the introduced algorithm takes into account the channel state and degrades the TXOP of stations with bad channel conditions until a minimum, distributing the exceeding time to the stations into good channel conditions and returning the stations to their normal TXOP when the channel state becomes good again. This allows for maximizing the overall throughput taking into account the current physical rate of nodes. Moreover, in order to ensure long-term and short-term temporal fairness, a lead-lag counter is used for accounting the transmissions state of each station with respect to the corresponding error-free service model as provided by a general centralized scheduler. Finally, two counters are used to memorize the received and the returned transmission time.

As far as the use of both HCCA and EDCA modes is concerned, if the purpose is to improve and extend in time the QoS service when HCCA has exhausted its accorded duration, a possible shortcut is trying to use EDCA to continue delivery of TSs packets. Few studies have investigated both these access functions and their behavior with different types of traffic. The Markov channel model in [17] shows that incrementing the HCCA duration increases the medium utilization of large WLAN in saturation conditions and the channel control determinism. Instead, large EDCA networks are affected by growing collisions that degrade their performance.

In [18] the economic model used to efficiently manage elastic traffic over EDCA and HCCA highlights that CSMA/CA method, *Request To Send/Clear To Send* (RTS/CTS) scheme, and the setting of  $CW_{min}$  parameter impact channel congestion and throughput. Moreover, the optimal HCCA-EDCA ratio is deduced through optimization techniques.

The *Adaptively Tuned HCF* (AT-HCF) algorithm [19] dynamically adapts the HCCA and the EDCA durations to the different types of traffic (VBR and CBR), until they converge to the optimal values to improve the delay and the throughput of the overall system. It is based on a two-step algorithm that adjusts the value of the *CapLimit* standard parameter that sets the maximum HCCA duration, without impacting on the different used schedulers but simply modifying the relative duration of the two MAC functions.

The *Overboost* local node scheduler [20] exploits EDCA bandwidth to integrate HCCA with the aim to limit the delay experienced by the traffic streams waiting for the next HCCA polling time and improve the network performance. Before the Contention Period begins, Overboost moves the TSs traffic exceeding the assigned HCCA TXOPs from the HCCA queues to the higher priority Access Category EDCA queue, providing the best QoS as available through the contention access. This scheduler can be integrated with any type of centralized HCCA scheduling algorithm that continues to manage admission control, scheduling parameters,

and polling list. Moreover it does not require any modification in the standard protocol parameters used to manage EDCA and HCCA, neither does it affect the scheduling of stations during HCCA and their accorded QoS but simply exploits EDCA to continue transmitting QoS TSs. Since EDCA is a distributed scheme, that is, without a central entity, Overboost is located in each node, allowing each QSTA to operate according to the contention rules and accessing to EDCA queues mechanism. Simulation and analytical results show that Overboost positively impacts on efficient resource management and on queues length and access delay. Indeed, the cooperation with Overboost has performance comparable with those of overprovisioning, admitting the same number of TSs with a better QoS or admitting more TSs with the same QoS, but without impacting on the admission control. From a different point of view, it allows same results than underprovisioning, saving resources but maintaining the negotiated QoS.

As example of first reclaiming mechanisms, in [21] a bandwidth reclaiming scheme for the PCF function with *Weighted Round Robin* (WRR) scheduling policy is proposed. It determines when the unused transmission time can be either used to advance the next polling opportunity or assigned to the Contention Period. Moreover, the WRR polling list is modified in order to put the stations with higher probability of generating unused time at the end of the list. This rearrangement aims to reduce the number of reclaimed stations but it makes the solution not extensible to HCCA function, where the polling order is strictly related to real-time guarantees.

In [22] an HCCA bandwidth reclaiming mechanism is presented and open-loop strategies, based on the use of TSPEC, and closed-loop strategies, that use the transmission queues information sent by stations to the QAP, are compared. Two different *max-min fairness* algorithms are proposed to reduce the HCCA delay in the case of VBR traffic by recovering spare resources through a proportional controller. A weighted proportional function assigns an additional amount of resources proportionally to the traffic class and to the buffer length. *MaxMin Fair-Adaptive* (MMF-A) assumes fixed SI, whereas *MaxMin Fair-Adaptive with Rescheduling* (MMF-AR) considers dynamic SI values. The nonlinearities that can affect the abovementioned work are overcome by [23] with an optimal controller, based on Model Predictive Control [24], that dynamically assigns resources in order to empty buffers of the stations and reduce packets loss.

Moreover, in [25] both cited proportional controller and optimal controller algorithms are applied to multiclass traffic with different priorities and to heterogenous traffic in order to find the resource assignment tailored to maximize the throughput, reducing the packet loss experienced by the different types of traffic.

*Immediate Dynamic TXOP HCCA* [26] scheduler introduces a bandwidth reclaiming scheme that tunes the assigned transmission time considering the transmission duration of previous polling of the considered station, in order to avoid providing further capacity when it is not needed.

However, since it is suitable to provide only short-term fairness in allocating recovered resources, same authors

proposed *Dynamic TXOP HCCA* [27] algorithm that refines TXOP computation integrating a mechanism suitable to estimate the required transmission time at the current polling by the use of time series forecasting applied to previous polling phases selected by a Moving Average. This provides a more accurate estimation of needed transmission time and a long-term fairness by means of a traffic monitoring window with tunable length.

The concept of deadlines is suitable to report timing constraints and to meet temporal requirements. The *Scheduling Estimated Transmission Time-Earliest Due Date* (SETT-EDD) [28] algorithm uses a token bucket of time units, or TXOP timer, to vary TXOP over time according to the node requirements. Indeed, considering the minimum and the maximum TXOP durations instead of using the mean value of TXOP, the authors propose to increase its value of a constant rate equal to the total fraction of time a QSTA can transmit during polling with respect to its minimum SI. This allows to adapt the transmission interval to VBR TSs. SI of each node is computed taking into account its traffic profile, varying from its minimum value (minimum SI) and the maximum one (maximum SI), which corresponds to its deadline. This solution, one of the first proposed for HCCA, allows for making TXOP and SI variable just exploiting the protocol parameters that set their minimum and the maximum values. Finally Delay-EDD [29], that computes deadlines considering delay bounds, determines the polling order. Variable TXOP and SI enhance the scheduler flexibility and lead to significant reduction in average transmission delay and packet loss ratio.

Instead the timer-based scheduler presented in [30] aims at providing variable and different SIs. SI of each station is computed as the smallest between the downlink and uplink deadlines that are calculated considering the different delay bounds of stations, in order to introduce a service differentiation tailored for CBR and VBR TSs. Then, traffic streams are scheduled according to *Earliest Deadline First* (EDF) [31] algorithm. This solution, even not exhaustive in terms of QoS since does not take into account the need of different TXOPs, is suitable to reduce the delay, the jitter, and the packets loss experienced by diverse types of traffic (voice and video) as well as increase the network capacity serving more video streams with respect to the reference scheduler, assuming the same packet loss.

*Real-Time HCCA* (RTH) [8] scheduler aims at ensuring each QSTA a fixed transmission time with a fixed period. In order to lighten the online scheduling, RTH has two components, the offline and the online activities. The first one performs the more complex work, executing admission control, computing transmission parameters, and scheduling timetable, where the TSPEC parameters are converted in capacity and period. Instead the online activity schedules traffic streams transmissions, simply applying the results of the first one (enforcement procedure); thus its computational complexity is  $O(1)$ . Since an EDF-based algorithm schedules TSs transmissions that introduce a prioritization of streams, RTH manages TXOP as critical section, which cannot be interrupted by higher priority flows, by using *Stack Resource Policy* (SRP) algorithm [32]. This implies the inclusion of

critical sections in the admission control schedulability test and increases the computational load. However, in the case of a new TS asking to be admitted, RTH states that the QAP can simply positively or negatively answer and postpone the transmission of the new TS in order to take the time to perform the heavy computation. Despite its complexity, RTH is suitable to improve the system efficiency, with respect to reference scheduler, admitting more TSs and living more capacity to EDCA, while ensuring the same QoS.

*Adaptive Resource Reservation over WLANs (ARROW)* [33] dynamically computes TXOPs by taking into account the actual buffered TSs data at the beginning of the polling. In particular, it simply uses the standard protocol *Queue Size* field of *QoS Data* frame to feedback the queue length at the beginning of transmission and that will be used to compute the duration of the subsequent TXOP as the sum of the maximum duration of each TSs. This allows for adapting the transmission interval to traffic profile that is useful especially in the case of VBR and bursty streams, as discussed in the previous sections. Moreover, a strict upper bound is set for MSI in order to ensure no deadline miss and delay requirements. Despite increased system overhead due to the augmented number of polling and the consequent increase of delay, the higher frequency of TXOP recomputation improves the system throughput in terms of number of served TSs, fraction of used TXOP, packets loss, and channel occupancy. Finally *Earliest Due Date (EDD)* [34] deadlines scheduling manages the QSTAs polling list.

The *Application-Aware Adaptive HCCA Scheduler* [35], derived from ARROW, distinguishes uplink and downlink schedulers, while EDF defines the QSTAs polling order taking into account the stations deadlines. The uplink scheduler assigns each QSTA a minimum and a maximum SI that imposes the respect of traffic period and delay bound, tailored to application (CBR and VBR) and network conditions and to the buffered traffic. Thus QAP continuously monitors transmission beginning and ending time plus new transmission requests and actual size of packets for each TS, and then compares these variables with the corresponding in TSPEC. In the case of downlink scheduler this computation exploits QAP buffered traffic and PHY rate. Even if this structure implies a precise signaling of this overall information and a cross-layer coordination between QSTAs and QAP, it allows improvements in terms of overall throughput, packets loss, and delay.

Since in what follows of this study *Wireless Capacity Based Scheduler (WCBS)* [36] will be used for the implementation of UTSS as example of centralized scheduler suitable to provide variable TXOP and polling order and different SI, it will be deeply analyzed in the subsequent subsection.

(1) *Wireless Capacity Based Schedule*. WCBS [36] is a deadlines-based centralized scheduler that aims at introducing a dynamic mechanism for allocating-recharging capacity taking into account used transmission time and temporal requirements. First of all, being focused on real-time expectations of TSs, it dynamically updates the polling list of TSs by means of EDF. The order of increasing deadlines allows reordering the original first-in first-out list to poll

first the stations with TSs with more strict temporal needs. Then the resource allocation and the access to the medium are managed by means of static and dynamic parameters, assigned differently to each QSTA to adapt the transmissions scheduling to its TSs characteristics. In particular, during the admission control, a pair of static parameters, unchanged until there are modifications in the service negotiation, for instance due to the association of new stations with more strict QoS requirements, are assigned to each  $TS_i$  taking into account its  $TSPEC_i$ : the budget  $Q_i$ , that is equivalent to  $TXOP_i$ , and the polling period  $P_i$ , that is, the service interval  $SI_i$ . Their ratio  $U_i = Q_i/P_i$  is the utilization factor of the stream, that is, its bandwidth that is considered in the admission control feasibility test. These two parameters set the transmission time allocation for the considered station and the general polling time limitation. In particular  $P_i$  is computed as the maximum SI for the considered TS. Instead  $Q_i$  is obtained by a weighted function of the minimum time  $Q_{\min}$  and the maximum time  $Q_{\max}$ .  $Q_{\min}$  is the time interval needed to transmit, during a period  $P_i$ , and at the mean data rate  $R_i$  a MSDU of nominal size  $L_i$ , whereas  $Q_{\max}$  is the time needed to send at peak data rate  $\Pi_i$ , a MSDU of maximum size  $M_i$ :

$$Q_{\min} = \left[ \frac{R_i P_i}{L_i} \right] \cdot t_n, \quad Q_{\max} = \left[ \frac{\Pi_i P_i}{M_i} \right] \cdot t_n, \quad (3)$$

where  $t_n$  is the nominal transmission time during a polling, computed as  $t_n = t_{\text{DATA}} + t_{\text{SIFS}} + t_{\text{ACK}} + t_{\text{SIFS}}$ .

Different from the reference scheduler, the polling periods can be different for the diverse QSTAs, in dependency of their TSs. Therefore the admission control test expressed by (2) begins as follows, where now different value of SIs are considered:

$$\frac{TXOP_{k+1}}{SI_{k+1}} + \sum_{i=0}^k \frac{TXOP_i}{SI_i} \leq \frac{T - T_{\text{CP}}}{T} \leq 1. \quad (4)$$

Instead, the dynamic parameters characterize the scheduling phase of each  $TS_i$  and its evolution. At each polling WCBS assigns a station its allocated resources, taking into account its  $Q_i$ . However, since during a transmission, especially in the case of VBR TSs, a station cannot exhaust its  $Q_i$ , WCBS, already using a recovery mechanism, does not waste the remaining transmission time but takes memory of that through the dynamic parameter  $c_i$ , the remaining time of the transmission just finished and that will be assigned to  $TS_i$  during its next polling. Different from the previously mentioned reclaiming mechanisms, WCBS preserves the saved resources only for the station that has not used that. Thus it acts like an accordion mechanism that does not waste resource but, at the same time, it is not able to follow traffic profile and add additional resources when needed. As far as the polling action is concerned a further dynamic parameter, the absolute deadline  $d_i$ , is used. It sets the absolute time within  $Q_i$  has to be exhausted, that is, it is related to the maximum delay bound that can be tolerated. Moreover it is used to determine the next polling time  $p_i$  that is updated when, during the current polling, there are no more data

to transfer or  $TXOP_i$  is exhausted. Finally the stream *state* defines its condition during the evolution of the scheduling: the state is *idle* if there are no data to send for the considered traffic stream, *polling* if this TS is in the polling list, *active* if it has been extracted from the polling queue and inserted in the EDF transmission queue, and *transmitting* if it is during its sending time.

Consequently, after these definitions, the scheduling engine is as follows.

- (i) During the admission control each  $TS_i$  is assigned the static parameters  $Q_i$  and  $P_i$ .
- (ii) At the beginning  $c_i = Q_i$  and  $p_i = d_i$ . TSs with *active* state are inserted in the EDF-ordered polling list considering their deadlines  $d_i$ .
- (iii) WCBS extracts the first TS from the polling queue and sets  $TXOP_i = c_i$ , changes its state into *transmitting*, and allows the corresponding station to transmit until there is sufficient capacity.
- (iv) When  $TXOP_i$  is exhausted, WCBS recharges the capacity  $c_i = Q_i$ , and the next polling time as  $p_i = d_i$  or postponing deadline, if the current time  $t_c > d_i$ , as  $d_i = t_c + P_i$ . The state becomes *polling*.
- (v) When  $p_i \leq t_c$  TS<sub>i</sub> will be polled again.
- (vi) If a transmission ends without exhausting the assigned budget, the remaining capacity will be simply preserved for the next polling.

## 4. Unused Time Shifting Scheduler

**4.1. Motivation.** Satisfactory QoS performance of HCCA networks for VBR traffic with real-time guarantees, typical of multimedia applications, has not yet been completely achieved and the HCCA reference polling mechanism shows some inefficiencies in the case of VBR traffic. Moreover, despite many QoS scheduling algorithms proposed as alternative to the reference one, some of them mentioned in Section 3, this type of traffic stresses the behavior of the different families of schedulers. Thus it is pressing to provide tailored and more efficient scheduling policies. For instance, EDF-based algorithms are well performing in the case of CBR traffic streams, whereas they are not able to follow the variations of the VBR ones [12]. Indeed, even if the postponing deadlines mechanism allows a more flexible scheduling in dependency of the actual resource needs, it hinders the respect of delay bounds requirements.

Furthermore, the waste of resources due to data rate variations affects the schedulers that do not implement any recovery policy. In particular, when the instant data rate drops down, a polled QSTA transmits data for an amount of time shorter than its assigned TXOP, easily dispatching the arriving traffic and the eventual enqueued packets. Then, as stated by the reference scheduler, the QoS Access Point, listening the idle channel for a time longer than a *Short Interframe Space* SIFS, assumes the control of the medium and polls the next station. Therefore the unused portion of TXOP is lost. On the other hand, when the instant data rate

goes up, the assigned TXOP, computed considering mean data rate, is not sufficient to deliver the incoming traffic and the station queue length increases. This has the side effect to increase the end-to-end delay and the number of discarded packets, due to the expiration of their validity time. A more flexible scheduling scheme could offer advantages in terms of end-to-end delay and amount of dropped packets in the case of high and variable data rate traffic.

**4.2. UTSS Description.** *Unused Time Shifting Scheduler* (UTSS) [6] aims at facing off the problem of not efficient resource management, especially in the case of VBR traffic, by reclaiming the unused transmission time. It provides a shortcut to have an instantaneous dynamic TXOP, without modifying the admission control (see Section 5). The basic idea is to remove from stations the unspent transmission time and to make that available to the next polled stations, especially for those requiring a longer transmission interval. In order to do that UTSS, that is, at its turn, a centralized mechanism, that is, it is located in the QAP, simply introduces a supplementary resource scheduling rule that just handles the recovery of the unused time, without effects on the centralized scheduling scheme. This is a valid alternative to the overprovisioning mechanism that affects admission control and efficient resource provisioning. The UTSS approach is greedy since it does not make differentiations between applications and stations but assigns the recovered resources to the next polled station, which can use this additional transmission capacity without impacting on the admission control threshold, as it will be shown in Section 5.

To illustrate its feature, as example of application of UTSS with a centralized QAP scheduler, the cooperation of UTSS with WCBS, that has been described in Section 3, is considered. Since WCBS is an EDF-based scheduler, whose limits have been explained above, it can be a meaningful case study suitable to show how UTSS works and its advantages.

UTSS integrates the action of the centralized HCCA scheduler by keeping the same admission control and scheduling algorithm. This is due to the fact that the admission control, that computes the basic protocol parameters (TXOP and SI) used during the following polling phases, is performed at the association of the stations to the QAP (it is reiterated only if a new QSTA asks to be admitted to transmit) and precedes the subsequent CAPs. It has no knowledge about the instantaneous unspent time of future transmissions and its computation is based on mean value of QoS parameters, as negotiated with the corresponding QSTAs. These considerations are valid for all centralized schedulers that do not use any type of traffic prediction mechanism.

Figure 1 summarizes the admission control activities of a centralized scheduler, like WCBS, that tests the feasibility condition and, then, computes the static parameters, the initial values of the dynamic ones, and the enqueueing of newly admitted TSs in the polling list, EDF-ordered for WCBS.

UTSS adds its further scheduling rule during the subsequent CAPs, in particular during the updating of the dynamic parameters and immediately before the polling

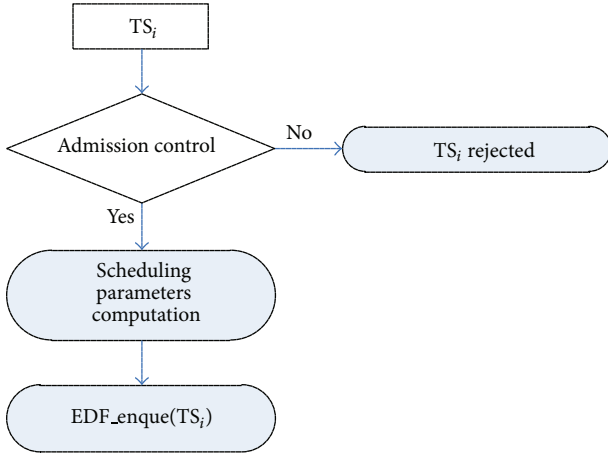


FIGURE 1: Admission control test and activity of a centralized scheduler.

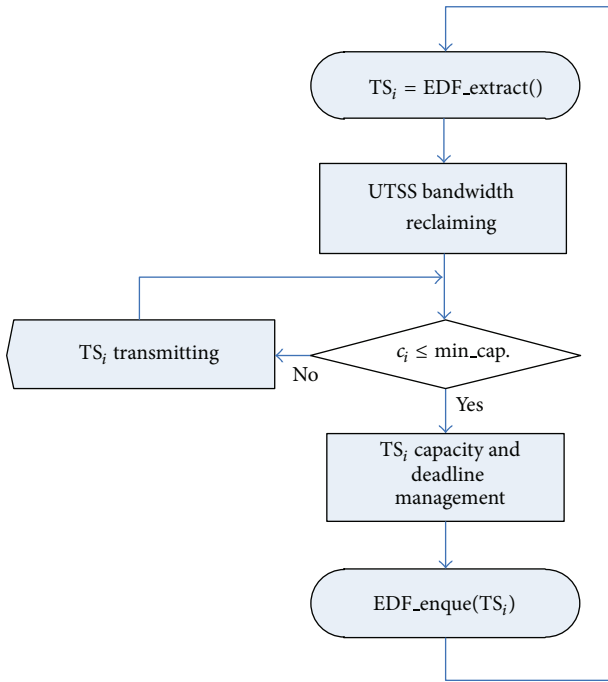


FIGURE 2: EDF-based (WCBS) + UTSS scheduling cycle.

of the considered station, by modifying the related TXOP computed during the admission control, if some unused time is available.

Figure 2 describes the interaction between the EDF-based WCBS and UTSS. The global scheduler based on their cooperation, after the extraction of the next  $TS_i$  from the polling queue, performs the UTSS bandwidth reclaiming that affects the currently assigned  $TXOP_i$ . Then, if there is enough capacity to transmit,  $TS_i$  is scheduled for transmission. Otherwise, in the considered example, WCBS operates according to its scheduling rules replenishing the capacity and, if necessary, postponing the deadline. Therefore, while WCBS uses a constant TXOP assigned during the admission

control phase, UTSS can make this parameter variable each time a TS is scheduled for transmission. In particular, each time a QSTA does not use its full allocated TXOP, UTSS assigns this unused time to the next scheduled TS extracted from the EDF queue during the current CAP. In order to illustrate this bandwidth reclaiming scheme the following notation is adopted:

$t_{end}$ : ending time of the transmission  $t_{end} = t_p + TXOP$ , when TXOP is completely exhausted ( $t_p$  is the polling time);

$T_{spare}$ : spare time of TXOP, computed as the difference  $t_{end} - t_{effective-end}$ , where  $t_{effective-end}$  is the time when QSTA has actually finished its transmission.

$T_{spare}$  is the variable used by UTSS and it is computed every time QAP polls a QSTA. In particular,  $T_{spare} > 0$  if

- (i) a station ends its transmission before  $t_{end}$  and it does not have enqueued traffic;
- (ii) a station ends its transmission before  $t_{end}$ , dispatching the incoming traffic and emptying its transmission queue before the ending of TXOP.

Then the current  $T_{spare}$  is added to the assigned  $TXOP_i$  of the next polled station  $QSTA_i$  that will receive a new  $TXOP'_i$  computed as follows:

$$TXOP'_i = \begin{cases} TXOP_i, & \text{if } T_{spare} = 0, \\ TXOP_i + T_{spare}, & \text{if } T_{spare} > 0. \end{cases} \quad (5)$$

Figure 3 better details UTSS and its interaction with the centralized scheduler, here WCBS. After the extraction from the EDF queue of the next TS to transmit, UTSS verifies if some spare time from previous transmissions is available. In such case, it adds  $T_{spare}$  to the TXOP computed by WCBS ( $TXOP_i = c_i - t_p$ ), (in general, to the TXOP computed by the centralized scheduler). Then the scheduler updates the estimated transmission ending time.

Hence next scheduled traffic streams can be transmitted not only exploiting the correspondent TXOP assigned during the admission control phase, (or the remaining capacity, if they have been already served, in the case of WCBS), but also by using the surplus  $T_{spare}$ .

Figure 4 shows a simplified scheduling example where a centralized scheduler and the centralized scheduler plus UTSS are compared. In the case of the centralized algorithm alone the unused bandwidth of  $TS_2$  is wasted whereas, with the help of UTSS,  $TS_3$  transmits its data during also the unused time of previously polled TSs, here of  $TS_2$ .

In this way it is possible to reclaim all unused TXOP portions. The total amount of  $T_{spare}$  can contribute to satisfy the temporal requirements of a highly variable traffic with a temporary load greater than the mean value used to calculate TXOP during the admission control. Hence  $T_{spare}$  can be useful to absorb traffic peaks which characterize highly variable bit rate applications, improving the obtained performance.



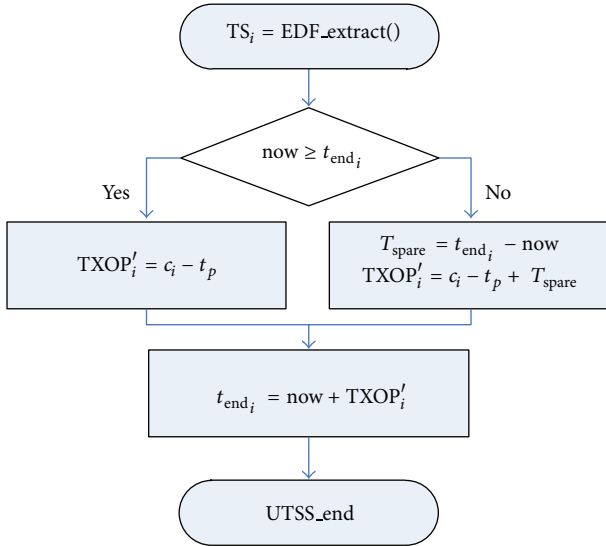


FIGURE 3: UTSS mechanism integrated with the centralized scheduler.

## 5. UTSS Scheduling Analysis

This section analyzes the UTSS algorithm in order to highlight its effects on the performance of a centralized HCCA scheduler. First of all the general properties of UTSS are studied focusing on its computational complexity, its impact on the system overhead, and its ability to preserve the temporal isolation, that is, to ensure a not interruptible transmission time. Then its strictly related real-time features are investigated. The purpose is to assess whether UTSS changes the admission control feasibility condition. Moreover, it is checked if the backlog mechanism of  $T_{\text{spare}}$  from previous transmissions can rise or not a deadline miss, jeopardizing the real-time behavior of the global scheduler, and an upper bound of the maximum acceptable  $T_{\text{spare}}$  is computed. This allows also for reducing the starvation eventually experienced by subsequent stations due to the greedy nature of UTSS.

Furthermore a deep study of the impact of UTSS on the instantaneous TXOP assigned to each station during the current polling has been performed considering different types of traffic: the aim is to highlight whether the integration of UTSS benefits the management of VBR traffic, improving flexibility and efficiency of QoS provisioning. This investigation is particularly meaningful since UTSS has been introduced to overcome the limit of some centralized HCCA schedulers when they, dealing with multimedia applications with highly variable bit rate, are not able to follow traffic variations. Different classes of data streams are considered, from CBR TSs to increasing VBR ones (from video conference to video streaming and to bursty traffic).

Finally, the impact of UTSS on transmission queues length is considered.

**5.1. General Properties.** First of all the computational complexity of UTSS is analyzed in order to evaluate if the integration of UTSS with a centralized scheduler has some

disadvantages with regard to the computational complexity of the global scheduling engine.

**Proposition 1.** *The computational complexity of UTSS is  $O(1)$ .*

*Proof.* Since UTSS simply computes  $T_{\text{spare}} = t_{\text{end}} - t_{\text{effective-end}}$  and it does not perform further operations like reordering and so forth, the computational complexity of the alone UTSS is  $O(1)$ . Thus the complexity of the global scheduler, obtained integrating UTSS, is equal to that of the centralized scheduler. The simplicity of UTSS is its basic strength and it makes UTSS a lightweight solution to improve the network performance without changing the centralized scheduler and without adverse effects from the point of view of the computational complexity.  $\square$

**Proposition 2.** *UTSS does not increase the system overhead.*

*Proof.* As deducible by the description of UTSS in Section 4, at the end of a polling this algorithm simply updates  $T_{\text{spare}}$ , whose value is used by the centralized scheduler to compute the next instantaneous TXOP. No exchange of additional information is required, nor the sending of special frames that can delay messages handshake. Thus it is possible to conclude that UTSS does not affect the centralized scheduler overhead. This deduction along with the previous Proposition 1 show that UTSS is not impacting on the basic scheduling features of the centralized scheduler.  $\square$

**Proposition 3.** *UTSS preserves the temporal isolation provided by the centralized scheduler.*

*Proof.* As temporal isolation is intended the not overlapping of transmissions of different QSTAs and the not interruption of a transmission. IEEE 802.11e guarantees the temporal isolation of the transmission of each station by introducing the concept of Transmission Opportunity. Indeed, as stated by the standard, TXOP assigned to a station cannot be consumed by other QSTAs and allows to perform a bandwidth reservation and assigns a station a maximum transmission duration, computed taking into account its QoS requirements. Moreover TXOP avoids execution overruns, due to stations aiming at transmitting more time than the negotiated one and that can jeopardize the service provided to other QSTAs.

When UTSS is scheduled in combination with a centralized scheduler, is the temporal isolation maintained? The answer is positive since UTSS preserves the TXOP set during the admission control and simply assigns an additional portion of sending time. Also this additional interval is not interruptible and it is not subtracted to other stations, jeopardizing their received service, but recovered from QSTAs that did not need that.  $\square$

These first propositions highlight how the adding of UTSS does not affect the basic scheduling features of the centralized scheduler. This makes the choice of integrating UTSS a low-cost solution to refine network performance.

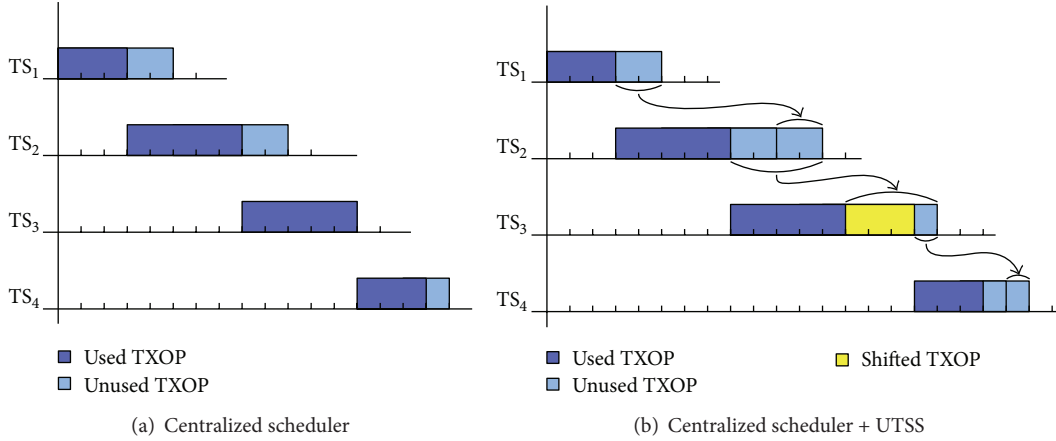


FIGURE 4: Scheduling example.

**5.2. Real-Time Analysis.** Since the introduction of UTSS aims at improving the real-time behavior of the centralized HCCA scheduler, some main real-time properties of the global scheduler are analyzed in the following. First of all the impact of UTSS on the admission control feasibility test is studied. Then, as diagnosis of deteriorated real-time performance, the presence of a deadline miss is investigated in the case of UTSS. Indeed, if an algorithm is not suitable to respect a temporal requirement, a deadline, intended as the time until the considered transmission has to be finished, is missed.

**Theorem 4.** *UTSS does not affect the admission control feasibility test.*

*Proof.* Equation (4) can be written as follows:

$$T_{\text{CAP}} + T_{\text{CP}} = \sum_{i=0}^K T_i + T_{\text{CP}} \leq H, \quad (6)$$

where  $H$  is the hyperperiod duration,  $T_{\text{CAP}}$  is the portion of  $H$  assigned to HCCA, whereas  $T_{\text{CP}}$  is the one assigned to EDCA,  $K$  is number of admitted HCCA stations, and  $T_i$  is TXOP <sub>$i$</sub>  of QSTA <sub>$i$</sub> .

Without loss of generality if it is assumed that, for instance, QSTA<sub>1</sub> does not use its whole assigned TXOP<sub>1</sub>, then  $T_{\text{spare}_1} > 0$ . Therefore, when UTSS reclaims unspent time and assigns  $T_{\text{spare}_1}$ , the following relationship holds:

$$\begin{aligned} \sum_{i=0}^K T'_i &= T_0 + T_{\text{eff}_1} + T'_2 + \cdots + T_K \\ &= T_0 + T_{\text{eff}_1} + T_2 + T_{\text{spare}_1} + \cdots + T_K \\ &= T_0 + T_{\text{eff}_1} + T_2 + T_1 - T_{\text{eff}_1} + \cdots + T_K \\ &= \sum_{i=0}^K T_i \leq H - T_{\text{CP}}, \end{aligned} \quad (7)$$

where  $T'_i$  is the new  $T_i$  computed taking into account  $T_{\text{spare}_1}$ , and  $T_{\text{eff}_1}$  is the used portion of TXOP<sub>1</sub>. This relationship

shows as UTSS simply rearranges the assignment of the transmission intervals, without using more resources than the allocated ones.  $\square$

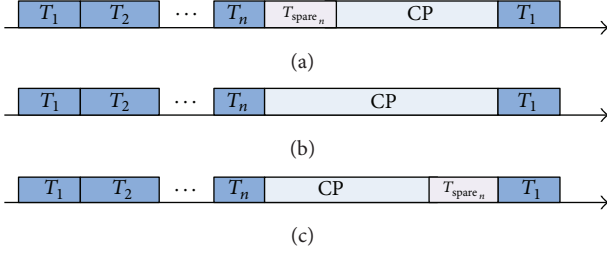
In Section 6 the analysis of null rate, polling interval, and throughput will confirm that UTSS does not affect the centralized scheduler policy.

**Theorem 5.** *The UTSS mechanism of  $T_{\text{spare}}$  assignment does not raise deadline miss.*

*Proof.* We distinguish two cases: (1) assignment of  $T_{\text{spare}}$  during a CAP, that is, the  $T_{\text{spare}}$  propagation is limited to the considered CAP and (2) assignment of  $T_{\text{spare}}$  during a CAP derived from the last polled QSTA of the previous CAP, that is,  $T_{\text{spare}}$  residue of a CAP can be transferred to the next one.

*Case 1.* During a CAP, the assignment of  $T_{\text{spare}}$  does not miss the deadlines of polled QSTAs. Indeed, the HCCA scheduling algorithms manage the temporal sequence of QSTAs access to the medium by the use of fixed interframe space intervals (*SIFS* and *PCF InterFrame Space PIFS*) specified by the reference scheduler. They set, respectively, the time interval between sending consecutive frames during a transmission and the waiting time between the end of a station transmission and the polling of the subsequent QSTA listed in the polling queue. These strict rules avoid the presence of idle blocking time between polled stations transmissions, as in general it could happen in real-time systems. Indeed, in general, in real-time systems an idle time can elapse between the processing of two consecutive tasks in dependency of their activation time, and the addition of a further slot of transmission time could jeopardize the real-time behavior of the next process to execute, raising a deadline miss. Instead, due to the mentioned IEEE 802.11e MAC scheduling rules, when  $T_{\text{spare}} \neq 0$ , there is only an advance of the transmission of next polled QSTA, without impacting its behavior, as shown in Figure 4.

*Case 2.* When  $T_{\text{spare}}$  is derived from the last polled QSTA during a CAP (see Figure 5(a)) two additional cases exist. If  $T_{\text{spare}}$  recovered from the transmission of the last QSTA in the

FIGURE 5: Example of  $T_{\text{spare}}$  propagation.

current CAP is shifted to the subsequent Contention Period, the beginning of the following CAP is not affected since, in this case, there is only a redistribution of time portions between *Contention Free Period* (CFP) and CP of the same hyperperiod; see Figure 5(b). Instead, if  $T_{\text{spare}}$  is propagated to the next CAP, by assigning the unused transmission time of the last polled QSTA to the first QSTA at the beginning of the next CAP, there is only an early polling of this QSTA of the same time interval; see Figure 5(c), whereas all other QSTAs are polled by respecting their delay bounds, thus without deadline miss. Moreover this choice increases the algorithm fairness by handling all QSTAs, including the first one of the new CAP, in the same way.  $\square$

The greedy nature of UTSS, due to the assignment of the whole  $T_{\text{spare}}$  to the next polled station, can cause starvation for the following QSTAs, especially when  $T_{\text{spare}}$ , derived from different CAPs, is large. This can result in deadline miss and reduce the global fairness. Moreover, a longer waiting time experienced by subsequent stations can imply the need of bigger buffers to collect the incoming traffic and can increase the packets drop rate. Thus, in order to avoid an unpredictable growing of  $T_{\text{spare}}$ , especially due to accumulation of a large number of unused portions of TXOPs when the  $T_{\text{spare}}$  propagation across consecutive CAPs is enabled, an upper bound  $\Theta$  is set, suitable to meet the deadlines. Furthermore this upper bound fairly mitigates the greedy nature of UTSS.

**Proposition 6.** *The upper bound of  $T_{\text{spare}}$ , suitable to avoid deadline miss, is equal to  $\Theta = d_i - t_{\text{end}_i} + \delta$ .*

*Proof.* In the case of one-hope propagation, that is, in the case of propagation of  $T_{\text{spare}_i}$  only to the following QSTA $_i$ :

$$T_i \leq T'_i \leq T_i + T_{i-1}, \quad (8)$$

where

$$T'_i = T_i + T_{\text{spare}_i}. \quad (9)$$

Instead, if all previous QSTAs of the current and, eventually, of the previous CAPs do not use their TXOPs,

$$0 \leq T_{\text{spare}_i} \leq \sum_{j=1}^{i-1} T_j. \quad (10)$$

Precisely, in this case  $T_{\text{spare}}$  is made available for the next polling after the currently polled QSTA has informed QAP

that it has no data to send by responding to CF-Po11 frame with a CF-Null frame. This handshake requires a time interval equal to  $\tau = \text{SIFS} + t_{\text{NULL}} + \text{SIFS}$ , where  $t_{\text{NULL}}$  is the time to send a CF-Null frame. Thus the general expression holds:

$$\begin{aligned} 0 &\leq T_{\text{spare}_i} \\ &\leq \sum_{j=1}^{i-1} (T_j - \tau) \\ &\leq \sum_{j=1}^{i-1} (T_j) - (i-1)\tau. \end{aligned} \quad (11)$$

This shows as  $T_{\text{spare}}$  can grow jeopardizing the respect of deadlines.

Finally, in order to avoid deadline miss, we can assume to accept  $T_{\text{spare}_i}$  if and only if it respects the upper bound  $\Theta$ :

$$0 \leq T_{\text{spare}_i} \leq d_i - t_{\text{end}_i} + \delta = \Theta, \quad (12)$$

where  $d_i$  is the absolute deadline of the polled QSTA $_i$  and  $\delta$  is a safety offset.  $\square$

The previous analysis demonstrates that, also about real-time properties, UTSS does not modify the related features of the centralized scheduler. Indeed its action is limited to the TXOP assignment, as shown in the following.

**5.3. Transmission Opportunity and  $T_{\text{spare}}$  Analysis.** Since UTSS acts on the transmission time assigned at each polling of a QSTA, it is interesting to evaluate how it impacts on the transmission time duration of different types of traffic and, in particular, on the service provided to QSTAs with VBR applications. Indeed, as previously remarked, TXOPs, computed during the admission control considering mean value parameters, may not be sufficient to follow the variations of VBR traffic, deteriorating network performance.

As mentioned in Section 2, during the admission control TXOP is computed by (1) as stated by the HCCA reference scheduler guidelines. Thus, in order to investigate the UTSS behavior with different types of traffic, VoIP, video conference, video streaming, and bursty traffic, that show increasing data rate variations, are considered. It is analyzed how the centralized scheduler computes the corresponding TXOPs and if these are suitable to provide the required service. Then, for each considered class of applications, the new expression of TXOP, modified by the introduction of UTSS, is analyzed and compared with the previous one in order to highlight any changes.

*(1) VoIP Traffic.* Since VoIP streams are characterized by ON periods, where the source generates CBR packets, and OFF periods, where no packets are sent, this type of traffic is an interesting example of how UTSS interacts with the management of CBR traffic.

**Proposition 7.** *UTSS does not interfere in the CBR traffic management, like the VoIP one, provided by the centralized scheduler.*

*Proof.* The talkspurt (ON) period of VoIP traffic is distributed according to the Weibull distribution [37] that models a one-to-one conversation. Thus the *probability density function* (p.d.f.) of its length  $T_{\text{ON}}$  is

$$p_{T_{\text{ON}}}(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \cdot \left(\frac{x}{\lambda}\right)^{k-1} \cdot e^{-(x/\lambda)^k}, & x \geq 0, \\ 0, & x < 0, \end{cases} \quad (13)$$

where  $k$  is the *shape* parameter and  $\lambda$  is the *scale* parameter depending on the streams.

The expected value of  $T_{\text{ON}}$  is  $E[T_{\text{ON}}] = \lambda \cdot \Gamma(1 + 1/k)$ , where  $\Gamma(z) = \int_0^{\infty} t^{z-1} \cdot e^{-t} dt$  is the *Gamma* function.

If during the admission control the protocol parameters are computed considering

$$\begin{aligned} \text{SI} &= T_{\text{ON}} + T_{\text{OFF}}, \\ R &= R_{\text{ON}} \cdot \frac{T}{(T_{\text{ON}} + T_{\text{OFF}})}, \end{aligned} \quad (14)$$

where  $T_{\text{OFF}}$  is the OFF period duration,  $R_{\text{ON}} = 1/T_{\text{ON}}$  is the mean data rate during the ON period, and  $T$  is the CBR traffic period, the protocol parameters reflect the traffic profile and the centralized scheduler is suitable to efficiently manage the incoming streams and to dispatch the enqueued ones. Thus, even without overprovisioning, the centralized scheduler is able to provide the needed resources and  $T_{\text{spare}}$ , if present, is not used.  $\square$

In Section 6 the analysis of queues length, mean delay, and dropped packets will confirm as, in the case of CBR traffic, the scheduler parameters and the offered service are untouched by the introduction of  $T_{\text{spare}}$ , whereas in the case of low VBR its contribution is limited. This is due to the fact that the centralized QoS scheduler is already suitable to serve the considered traffic streams allocating the required resource. Consequently, in this case the capacity added by UTSS is not necessary. Indeed, UTSS does not perform any traffic monitoring and/or prediction; thus it is not able to adapt its behavior to a different traffic profile. It simply operates as a greedy algorithm that recovers unexhausted resources avoiding their waste and allocates that to the next polled station, without verifying through monitoring or statistical studies if this station has shown a traffic behavior that can be benefited by these further resources. This is its flaw but also its strength since using a simple mechanism is suitable to improve network performance without impacting on the system overhead due to  $O(1)$  computational complexity, as shown in Proposition 1. Thus, even when its action is not needed, it is running without overloading the scheduler computation.

(2) *Video Traffic.* The analysis of video traffic is useful to verify how VBR TSs are managed by the centralized scheduler when it collaborates with UTSS. The study has been performed considering low variable (video conference) and high variable (video streaming) applications.

#### (a) Video Conference Traffic

**Proposition 8.** *UTSS positively impacts on video conference traffic management.*

*Proof.* The literature has shown that video conference traffic has a stochastic behavior and already first studies provided some analytical models of these traffic sources. In particular, we are interested in “quantify” and modeling the traffic fluctuations. For instance, already in [38, 39] it is shown that, using a Discrete Autoregressive Model, for video conference traffic over ATM networks the number of cells (packets) per frames is a stationary stochastic process distributed accordingly with the negative binomial distribution:

$$f_{k,r,p} = \binom{k+r-1}{k} p^r q^k, \quad k = 0, 1, \dots, \quad (15)$$

where  $k$  is the number of successes,  $0 < p < 1$  is the failure probability,  $q = 1 - p$  is the success probability, and  $r > 0$  is the number of failures. Its mean value and variance are equal to

$$m = \frac{r(1-p)}{p}, \quad v = \frac{r(1-p)}{p^2}. \quad (16)$$

In the same works it has been shown that also the number of cells per frame and the bit data rate can be modeled by the Gamma distribution. This distribution has a mean value  $m = s/\lambda$  and a variance  $v = s/\lambda^2$ , both depending on the values of shape  $s$  and scale  $\lambda$  parameters. In both cases, considering the reference scheduler computation of TXOP based on mean value parameters, (1) begins as follows:

$$\text{TXOP}_{i_{\text{videoconf}}} = \max\left(\frac{N_i \cdot \mu \cdot m_{ci}}{m_{ri}}, \frac{M_i}{m_{ri}}\right) + O, \quad (17)$$

where  $m_{ci}$  is the mean value of number of cells per frame,  $\mu$  is the mean dimension of cells, and  $m_{ri}$  is the mean data rate, whose expressions can be derived by the cited distributions. Even if video conference traffic is characterized by low variable bit rate, it is affected by fluctuations around its mean value, expressed by the variance  $v$ . Thus, even without taking into account deeper statistical analysis about the distribution of cells per I, B or, P frames, the computation of TXOP and SI using mean values parameters does not reflect traffic variations and underestimates the instantaneous resource needs, harming the supported QoS.

These considerations are independent from the used codec but reflect the traffic and data rate profile. Thus this type of application can benefit from the use of UTSS.  $\square$

#### (b) Video Streaming Traffic

**Proposition 9.** *The service provided to video streaming traffic may benefit from the cooperation with UTSS.*

*Proof.* Video streaming traffic is characterized by highly variable bit rate. Adopting the same approach as in the case of video conference, its basic features about data rate and frame

size distributions are analyzed. VBR MPEG4 traffic is composed by *Groups of Pictures* (GoP) that consist of twelve I, P, and B frames, following the pattern IBBPBBPBBPBB, each of them differently coded and with different lengths. Moreover, some correlations exist between GoPs and it is shown that the related probability density and autocorrelation functions are characterized by both *Short Range Dependence* (SRD) and *Long Range Dependence* (LRD), [40]. Thus, due to its complexity, the study of video streaming traffic is challenging and numerous models have been proposed in the literature [40–42], most of them derived from the basic works on *AutoRegressive Moving Average* model [43]. Furthermore an increasing attention has been recently addressed to traffic dynamic prediction, [44]. The huge effort in modeling this type of traffic trying to reflect its variations implies that, also and especially in this case, the use of mean value parameters in the TXOP computation, as stated by the reference scheduler, is really far from being able to follow VBR variations, independently from the used model. Hence, also in this case, it is possible to deduce that provide additional resources by means of UTSS can help absorb traffic variations.  $\square$

In Section 6, the analysis through simulations of queues length, mean delay, and dropped packets will show the positive impact of UTSS on video conference and video streaming traffic scheduling.

**5.4. Bursty Traffic.** Finally, analyzing the effects of UTSS on the service provided to traffic with increasing VBR, its impact on bursty traffic, that has highly variable packets interarrival time, is studied.

**Theorem 10.** *When UTSS is used, the resulting global scheduler is able to manage greater bursts of traffic, with a burstiness factor  $B'_i < B_i$ .*

*Proof.* At the polling time of a station QSTA<sub>*i*</sub> its new TXOP'<sub>*i*</sub>, computed by UTSS considering (1), is

$$\text{TXOP}'_i = \max\left(\frac{N_i \cdot L_i}{R_i}, \frac{M_i}{R_i}\right) + O + T_{\text{spare}_i}. \quad (18)$$

A metric useful to characterize the burstiness of traffic with high rate variability is the *burstiness factor* ( $B$ ), defined as the *ratio between mean data rate  $R$  during a long time interval and peak data rate  $\Pi$  during the activity interval*. Its value ranges from 1, in the case of CBR traffic, to 0, as the peak rate increases. Introducing  $B$  the expression of TXOP'<sub>*i*</sub> becomes as follows:

$$\text{TXOP}'_{i_{\text{burst}}} = \max\left(\frac{N_i \cdot L_i}{B_i \cdot \Pi_i}, \frac{M_i}{B_i \cdot \Pi_i}\right) + O + T_{\text{spare}_i}, \quad (19)$$

where  $R_i$  and  $\Pi_i$  are, respectively, mean data rate and peak data rate of QSTA<sub>*i*</sub>. When  $T_{\text{spare}_i} > 0$ , TXOP'<sub>*i*</sub> > TXOP<sub>*i*</sub>. In this case, if  $T_{\text{spare}_i}$  is incorporated in the first term of the previous equation, since  $N_i$ ,  $M_i$ ,  $\Pi_i$ , and  $L_i$  do not change being related to the considered TS, then the only element that

can vary is  $B_i$ . Consequently, since TXOP'<sub>*i*</sub> > TXOP<sub>*i*</sub>, then  $B'_i < B_i$ , and TXOP'<sub>*i*</sub> can be expressed as follows:

$$\text{TXOP}'_{i_{\text{burst}}} = \max\left(\frac{N_i \cdot L_i}{B'_i \cdot \Pi_i}, \frac{M_i}{B'_i \cdot \Pi_i}\right) + O. \quad (20)$$

This implies that UTSS is able to manage greater bursts of traffic through an overprovisioning of resources, due to  $T_{\text{spare}_i}$ , local to the current polling and without any effect on the admission control, as previously demonstrated.  $\square$

(1) *Analysis of Burstiness Admissible by the System.* The previous theorem suggests that the burstiness admissible by the system can increase when UTSS is used. In the following this effect is deeply investigated, trying to quantify this increase and its impact on the provided service.

**Theorem 11.** *UTSS is suitable to increase the maximum tolerable burstiness, locally at the current polling, and raise the service rate  $r$ .*

*Proof.* Assume to adopt a token bucket model for the enqueued traffic of a QSTA and for the network service. If a centralized HCCA scheduler, like WCBS, is considered the state of the traffic incoming and outgoing is illustrated in Figure 6, where the  $x$  axis is the time and the  $y$  axis is the packets.

In this case the maximum admissible burstiness is represented by the *bucket dept*  $b$  parameter, whereas the mean data rate  $R$  is equivalent to the mean service rate  $r$ , represented by the slope of the service curve. Furthermore the total delivered packets during the current polling are represented by the parameter  $\psi$  that takes into account the token bucket depth and the service rate. All these parameters are evaluated during the current polling, that is, during the time interval ranging from the polling time  $t_{p_i}$  to the transmission ending time  $t_{\text{end}_i}$  of QSTA<sub>*i*</sub>, that is, where UTSS takes action.

If UTSS is integrated with the centralized scheduler, when QSTA<sub>*i*</sub> is polled and  $T_{\text{spare}} \neq 0$ , the station can transmit its  $\psi$  bytes plus the  $\varphi$  bytes sent during  $T_{\text{spare}}$ . The new situation is shown in Figure 7.

Since UTSS makes available  $T_{\text{spare}}$  at the polling time  $t_{p_i}$ , if it is assumed that the service rate  $r$  is constant, this situation is equivalent to have a greater token bucket dept  $b' = b + \varphi$ . This deduction confirms Theorem 10.

Figure 7 allows a further consideration: if it is assumed that the burstiness is constant, providing  $T_{\text{spare}}$  is equivalent to have a service curve with a higher slope, that is, to increase the service rate  $r'$ . This implies an overprovisioning local to the current polling, not obtained by rerunning the admission control, and yet confirms, from a different point of view, Theorem 10. These results are valid for any type of TS and corroborate the considerations about VBR applications.

Such overprovisioning can positively influence network performance expressed in terms of end-to-end delay and can reduce the packets waiting time in the queues, the packets drop, and the queues length. This conclusion will be confirmed by delay, packets drop rate, and queue length analysis illustrated in Section 6. The risks of starvation and

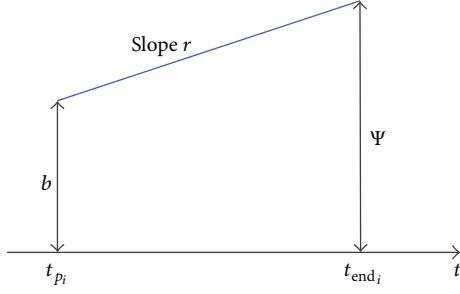


FIGURE 6: Service model of a HCCA centralized scheduler.

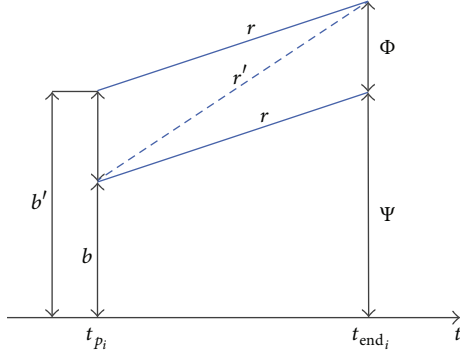


FIGURE 7: Service model of a HCCA centralized scheduler + UTSS.

increase of buffering amount and of exceeding the admission control threshold are avoided by taking advantage of UTSS features that do not impact on the admission control feasibility test (see Theorem 4) and improves the system fairness; see Proposition 6.  $\square$

In the following the maximum burstiness tolerable by the network will be computed.

**Theorem 12.** *With UTSS the maximum traffic burstiness tolerable by the system is increased and the corresponding burstiness factor  $B_{\max}$  is equal to*

$$\begin{aligned} B_{\max} &= \frac{t_{\text{end}_i} - t_{p_i}}{T_{\text{spare}_{\max}} + t_{\text{end}_i} - t_{p_i}} \\ &= \frac{t_{\text{end}_i} - t_{p_i}}{\Theta + t_{\text{end}_i} - t_{p_i}} \leq B, \end{aligned} \quad (21)$$

where  $B$  is the burstiness factor without UTSS and  $\Theta$  is the upper bound of  $T_{\text{spare}}$ ; see Proposition 6.

*Proof.* As shown by the previous results,  $T_{\text{spare}} \neq 0$  allows for dispatching more enqueued traffic, especially in the case of VBR traffic, and improving the capacity of absorbing traffic streams peaks. Thus, with regard to the assigned TXOP, UTSS has the same effect of instantaneously increasing the delivery rate. When a QSTA<sub>*i*</sub> does not use its whole  $T_i$ , then

$$\tau \leq t_{\text{end}_i} - t_{p_i} = T_{\text{eff}_i} \leq T_i, \quad (22)$$

where  $\tau = \text{SIFS} + t_{\text{NULL}} + \text{SIFS}$ , is the minimum transmission time used by a station when it has no data to send, and  $r'$  is equal to

$$r'_i = \frac{\varphi_i + r \cdot (t_{\text{end}_i} - t_{p_i})}{t_{\text{end}_i} - t_{p_i}} = \frac{r \cdot (T_{\text{spare}} + t_{\text{end}_i} - t_{p_i})}{t_{\text{end}_i} - t_{p_i}}, \quad (23)$$

where  $\varphi_i$  is the bytes sent using  $T_{\text{spare}}$ . Thus the maximum burstiness  $B_{\max}$  tolerable by the system, taking advantage by the use of UTSS, is

$$B_{\max} = \frac{t_{\text{end}_i} - t_{p_i}}{T_{\text{spare}_{\max}} + t_{\text{end}_i} - t_{p_i}} \leq B, \quad (24)$$

and, if we consider  $\Theta$ , the upper bound of  $T_{\text{spare}}$ , the following equation holds:

$$B_{\max} = \frac{t_{\text{end}_i} - t_{p_i}}{\Theta + t_{\text{end}_i} - t_{p_i}} \leq B. \quad (25)$$

$\square$

**5.5. Transmission Queues Length Analysis.** The previous analyses about the impact of UTSS on the TXOP duration suggest a deeper investigation on its effect on the stations queues length.

**Theorem 13.** *When UTSS is ON  $T_{\text{spare}} \neq 0$  allows increasing the portion of dispatched traffic during a polling phase and reducing the transmission queues length.*

*Proof.* In order to evaluate the impact of  $T_{\text{spare}}$  on the transmission queues length, it is considered, as worst case, a station QSTA<sub>*i*</sub> with backlogged traffic: in this case the station can exhaust its TXOP<sub>*i*</sub> and may need more time to dispatch the incoming traffic. In general, the traffic that can be delivered during a polling of a *i*th station, which is assigned its TXOP<sub>*i*</sub>, is equal to

$$\lambda_i = \text{TXOP}_i \cdot r, \quad (26)$$

where  $r$  is the mean transmission rate. When  $T_{\text{spare}} \neq 0$  a longer transmission time is available and the delivered traffic begins as follows:

$$\lambda'_i = (\text{TXOP}_i + T_{\text{spare}_i}) \cdot r. \quad (27)$$

Since an upper bound of  $T_{\text{spare}_i}$ ,  $\Theta_i$ , is set in order to avoid deadline miss, it limits also the maximum dispatched traffic:

$$\lambda'_{i_{\max}} = (\text{TXOP}_i + \Theta_i) \cdot r \geq \lambda_i. \quad (28)$$

Thus the expression of the maximum time gain  $g_{(t_i)_{\max}}$  for QSTA<sub>*i*</sub>, obtained with UTSS during the current polling and intended as the percentage of increase of transmission time duration, is

$$\begin{aligned} g_{t_i} &= \frac{r \cdot (\text{TXOP}_i + T_{\text{spare}_i})}{r \cdot \text{TXOP}_i} = \frac{\text{TXOP}_i + T_{\text{spare}_i}}{\text{TXOP}_i} \\ &\leq g_{(t_i)_{\max}} = \frac{\text{TXOP}_i + \Theta_i}{\text{TXOP}_i}. \end{aligned} \quad (29)$$

The enqueued traffic of QSTA<sub>*i*</sub> during the *j*th polling can be expressed as

$$Q_{i_j} = Q_{i_{j-1}} + \pi_i - \lambda_i = Q_{i_{j-1}} + \text{TXOP}_i \cdot R_i - \text{TXOP}_i \cdot r, \quad (30)$$

where  $Q_{i_{j-1}}$  is the enqueued traffic residue from the previous (*j* - 1)th polling phase,  $\pi_i$  is the incoming traffic during the current *j*th polling, and  $R_i$  is the mean data rate. When  $T_{\text{spare}} \neq 0$  the enqueued traffic  $Q'_{i_j}$  is bounded by  $Q'_{i_j \text{min}}$  and it is such that

$$\begin{aligned} Q'_{i_j \text{min}} &= Q_{i_{j-1}} + \text{TXOP}_i \cdot R_i - (\text{TXOP}_i + \Theta_i) \cdot r \\ &\leq Q'_i = Q_{i_{j-1}} + \text{TXOP}_i \cdot R_i - (\text{TXOP}_i + T_{\text{spare}_i}) \cdot r \\ &\leq Q_i. \end{aligned} \quad (31) \quad \square$$

In Section 6 the simulations results about queues length will confirm these analytical deductions.

## 6. Performance Analysis

In this section the performance of the proposed UTSS scheduling algorithm integrated with WCBS versus WCBS, FHCF, RTH, and reference schedulers are evaluated through extensive simulation results. The reference scheduler is the usual benchmark when analysing the behavior of alternative scheduling algorithms. Instead WCBS has been considered a right candidate to evaluate the effect of UTSS integration on the real-time behavior of a global scheduler since, being an EDF-based algorithm, its real-time performance is not optimal for VBR TSs [12]. Furthermore, even if its capacity recharging mechanism does not waste the accorded budget and efficiently assigns resources trying to follow traffic needs, it can be further improved through the UTSS recovery mechanism. The performances of WCBS are the starting point where UTSS takes action, like in the case of any other centralized scheduler, and, consequently, they affect the behavior of the final global scheduler. Indeed UTSS has been conceived as a cooperative scheduling algorithm. Thus, since the final performances are affected by the original scheduler, each effect has to be evaluated against the centralized one (WCBS) and, consequently, the comparison between WCBS and its integration with UTSS is an example suitable to highlight how UTSS influences and changes the scheduler behavior. As far as FHCF and RTH are concerned, their selection is based on similar goals and/or scheduling schemes. Indeed, since the aim of this section is to analyze the benefits introduced by UTSS with respect to a centralized scheduler alone, we decided to focus the comparison on algorithms showing some common features with WCBS, instead of considering algorithms belonging to different families that can show different performance. In particular, we have chosen FHCF first of all since it is a famous benchmark in the evaluation of HCCA schedulers alternative to the reference one, and, then, since it provides variable TXOPs such as WCBS. Instead

TABLE 1: MAC/PHY simulation parameters.

Parameter	Value
SIFS ( $\mu\text{s}$ )	10
DIFS ( $\mu\text{s}$ )	28
PIFS ( $\mu\text{s}$ )	19
Slot time ( $\mu\text{s}$ )	9
PLCP header (b)	24
Preamble (b)	72
Data rate (Mbit/s)	54
Basic rate (Mbit/s)	1

RTH has been selected since, even if its resource assignment is different from WCBS, it is based on a similar EDF-based polling scheduling.

In particular, the analysis takes into account the utilization efficiency of the network, the mean access delay, the discarding rate of enqueued packets with expired delay bound, and the transmission queues length. A description of the simulation tools, their settings, the traffic models, and the considered scenario precedes the simulation results analysis.

**6.1. Simulation Settings and Traffic Model.** Performance is evaluated through simulation using *ns-2* network simulator [45]. It is assumed that QSTAs communicate directly without hidden node problem; RTS/CTS mechanism, MAC level fragmentation, and multirate support are disabled. The physical layer is specified in the IEEE 802.11g standard where OFDM (*Orthogonal Frequency-Division Multiplexing*) is the mandatory modulation scheme; its parameters are listed in Table 1.

The presented results have been obtained simulating independent replications of 700 s with a warm-up time of 100 s until the 95% confidence interval is reached for each measure.

The network scenario chosen for simulations is composed by seven QSTAs and one QAP. Each QSTA transmits *one* uplink TS which is received by the QAP. Each TS has a different TSPEC. In particular, one station sends G.729A VoIP traffic, one station a video conference, and five stations video streaming applications. Finally, one station transmits data traffic with SDU of 1500 bytes through legacy Distributed Coordination Function and operates in asymptotic condition; that is, it is always backlogged in order to saturate the channel. The VoIP traffic parameters are shown in Table 2.

Video streaming traffic has been generated using pre-encoded high quality MPEG4 trace files of 60 minutes each from the Internet archive of traces [46]. Such traces are as follows: *Jurassic Park* (VS1), *Silence of the lambs* (VS2), *Mr. Bean* (VS3), *Die hard III* (VS4), and *Robin Hood* (VS5). The video conference (VC) session has been represented by the preencoded *LectureHQ-Reisslein* trace file. The video streaming and video conference parameters are summarized in Table 3.

**6.2. Efficiency Analysis.** In this paragraph the efficiency of the resource management performed by UTSS integrated with

TABLE 2: G.729A VoIP traffic stream parameters.

Parameter	Value
Frame size (B)	10
Frame per packet	2
Period (s)	0.02
Data rate (kb/s)	24
Payload size (B)	20
IP/UDP/RTP	
Header size (B)	40
SDU size (B)	60

TABLE 3: Video streaming and video conference parameters.

Parameter	VC	VS1	VS2
Mean frame size (B)	600	3800	2900
Maximum frame size (B)	11386	16745	22239
Period (s)	0.033	0.040	0.040
Mean data rate (kb/s)	158	770	580
Maximum data rate (kb/s)	2733	3300	4400
Parameter	VS3	VS4	VS5
Mean frame size (B)	2900	3500	4600
Maximum frame size (B)	15251	16960	16550
Period (s)	0.040	0.040	0.040
Mean data rate (kb/s)	580	700	910
Maximum data rate (kb/s)	3100	3400	3300

WCBS (labeled as UTSS + WCBS) is evaluated and compared with that of the mentioned schedulers used as benchmark (FHCF and RTH).

The first efficiency parameter is the *null rate* defined as the number of CF-Null packets sent by a QSTA as answer to a CF-Poll, when it has no traffic to transmit. A high value of null rate highlights a polling period more frequent than necessary for the considered application and can increase the system overhead due to unnecessary polling phases that occupy a minimum time interval.

In Figure 8 UTSS + WCBS and WCBS alone have similar values of null rate for each traffic stream: this is because they use the same polling interval, since UTSS does not affect this action. Often their null rate and that of RTH are better than that of the reference scheduler since they poll the stations using different SI for each TS: such feature lets these schedulers perform a zero null rate with some TSs (e.g., VC and VS2), or, however, small values. As expected FHCF shows a null rate less than the reference scheduler but, in most cases, higher than that of WCBS, WCBS + UTSS, and RTH since, even if it does not provide SIs different from the reference, it is more aggressive in emptying transmission queues. Also, in the case of VoIP traffic, it shows the greater null rate. For the same reason, for some video streams UTSS + WCBS, WCBS, and RTH have a null rate greater than that

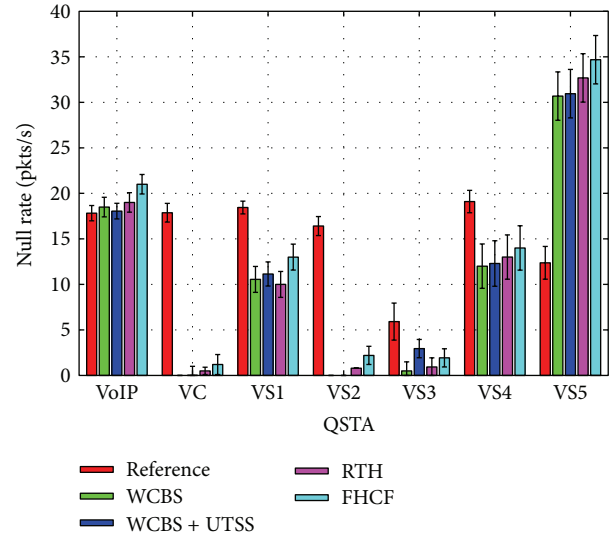


FIGURE 8: Null rate.

of the reference: since they provide tailored TXOPs they are able to dispatch more packets and the QAP can poll the stations finding empty queues; the integration of UTSS can exalt this phenomenon. In particular, this evaluation suggests a distinction: considering the reference scheduler the high values of null rate are due to the fact that it polls all TSs with the same SI, less than the minimum MSI, with the risk of polling some QSTAs more frequently than needed. Instead, focusing on the improved schedulers, a sporadic greater null rate (see FHCF) is due to the efficient computation of TXOP, tailored to the type of traffic. This is the same reason of the null rate of UTSS + WCBS greater than that of WCBS and, sometimes of RTH, in the case of some video streaming (VS1, VS3, VS4, and VS5), in particular that with high variability in the data rate. In this scenario the UTSS reclaiming mechanism adds the capacity needed to dispatch more enqueued traffic, having the same effect of a dynamic computation of TXOP tailored to the different TSs.

Finally, in the case of WCBS + UTSS, even if the polling instant can be advanced, the polling interval variation is negligible and it does not affect the average null rate. This confirms that UTSS does not affect the polling timing of the centralized scheduler.

To go deep inside this deduction Figure 9 displays the polling interval used by the schedulers with each TS. The reference scheduler and FHCF use a unique value of the polling interval for all TSs, as reported in Section 3, which is less than the minimum MSI of all admitted TSs. Instead, WCBS, WCBS + UTSS, and RTH poll each  $TS_i$  using its  $SI_i$ . In particular, WCBS and RTH show similar values since they use EDF to sort the next polling time, taking into account the deadline of each stream, whereas WCBS + UTSS has the same values than WCBS since UTSS does not care about the polling scheduling that continues to be managed by the centralized scheduler; in this case WCBS. This confirms that the integration of UTSS with WCBS does not touch its polling policy, as stated in Section 4.



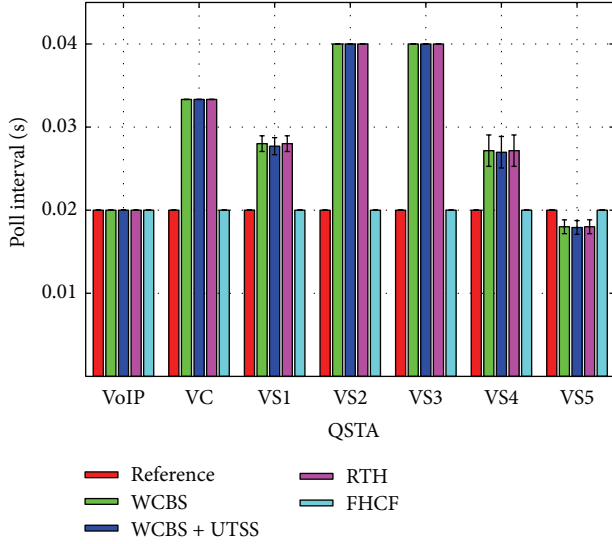


FIGURE 9: Polling interval.

Finally, Figure 10 shows that UTSS does not significantly affect the global throughput; thus global network utilization is roughly the same. However, more specifically, UTSS can produce a throughput greater than that of WCBS, using more efficiently the allocated resources and improving the behavior of the global scheduler: since it takes action especially in the case of high variable data rate, its positive effect is more relevant for high VBR TSs, like VS2 and VS3, where WCBS + UTSS is able to outperform also RTH and FHCF, unlike WCBS. In the remaining cases, where the contribution of UTSS is less meaningful, WCBS + UTSS performs better but its behavior is biased by WCBS.

As proof that UTSS is effectively suitable to recover not exhausted transmission time, in Figure 11 an example of dynamic evolution of  $T_{\text{spare}}$  is displayed. The  $x$  axis reports the scheduler tick since  $T_{\text{spare}}$  is sampled when the scheduler takes action. This Figure illustrates as, in dependency of the different type of traffic streams,  $T_{\text{spare}}$  shows extended variations, reaching some significative values around 5-6 ms. This could happen in the case of low VBR TSs, when the data rate drops down allowing the recovery of a considerable portion of transmission time.

**6.3. Delay Analysis.** In this paragraph the analysis is focused on the real-time behavior of the considered schedulers, intended as temporal performance expressed in terms of access delay. The access delay is defined as the time elapsed from when the frame reached the MAC layer until when the frame is successfully acknowledged.

Figure 12 highlights that the mean value of access delay of WCBS + UTSS is improved with respect to WCBS, RTH, and FHCF, in particular for traffic streams with higher VBR, (VS2 and VS3). This confirms the analytical results about deadline miss (see Theorem 5). Compared to the reference scheduler, WCBS and, consequently, WCBS + UTSS and RTH perform worse when serving TSs with less variable bit rate.

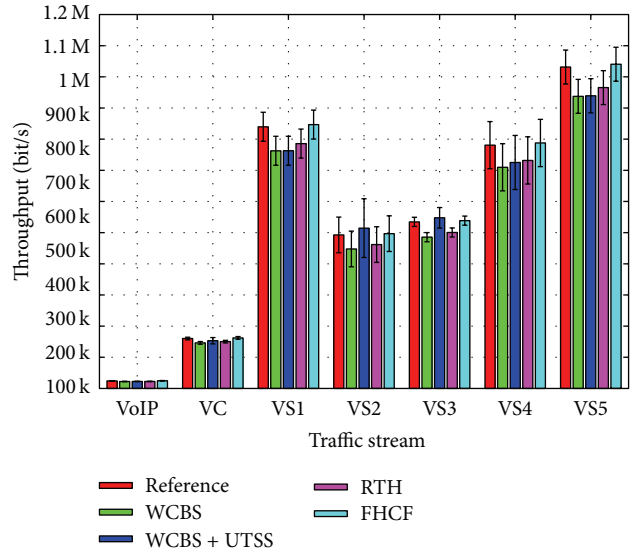
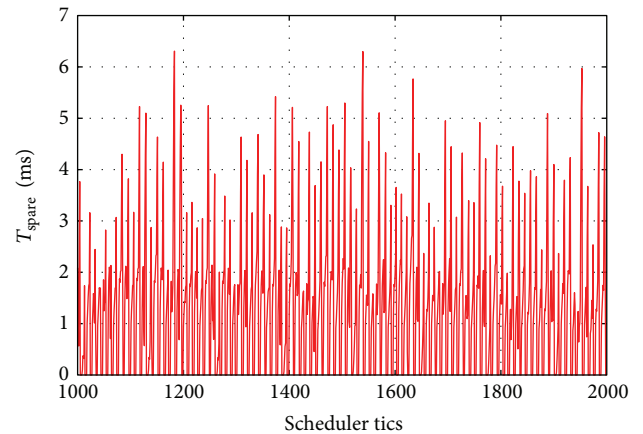


FIGURE 10: Throughput of VBR stations.

FIGURE 11: Example of dynamic evolution of  $T_{\text{spare}}$ .

In particular, WCBS is always underperforming due to its postponing deadlines and capacity recharging mechanisms that provide a variable TXOP but not tailored to the type of traffic. This result points out to the mentioned drawback of EDF-based schemes [12], since their postponing deadline mechanism improves the scheduling responsiveness but can increase the experienced delay. This is confirmed by the RTH behavior in comparison to the reference scheduler.

The previous considerations about the resource efficiency and the scheduling profile of WCBS + UTSS are not surprising since the behavior of the global scheduler is affected by that of the centralized one; thus, if this is underperforming in terms of access delay, like WCBS, this aspect can be propagated also in the global scheduler. However the cooperation with UTSS can improve the global performance in terms of the access delay with respect to WCBS, as shown in the case of all the analyzed TSs. Indeed, even if the access delay of WCBS + UTSS continues to be biased by the performance of WCBS, as previously motivated, in the case of highly VBR TSs

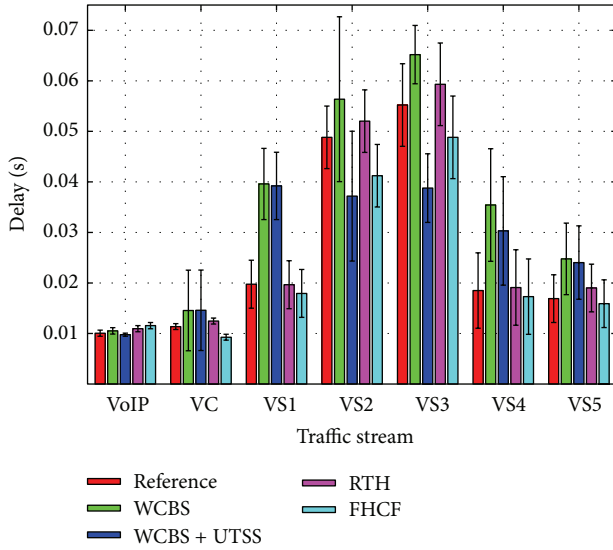


FIGURE 12: Mean access delay.

(see VS3 and VS4), where WCBS fails against the other alternative algorithms, when UTSS is added, the global scheduler is able to outperform also RTH and FHCF. This corroborates the validity of the UTSS scheme that results in being suitable to meaningfully improve the performances of WCBS in the situations where it performs worse.

This deduction suggests to further investigate the impact of UTSS on high VBR TSs in order to quantify the introduced changes. Looking at the *Cumulative Distribution Function* (CDF) of the access delay in the case of VS3, that has high VBR and showed in the previous Figure 12 an interesting improvement of access delay, Figure 13 displays that, after a time interval of 0.05 s, UTSS + WCBS allows 70% of transmitted packets whereas WCBS is limited to only 30%. This is the most important result that highlights how UTSS is suitable for improving the real-time behavior of WCBS in the case of high VBR traffic. Moreover, it corroborates the theoretical deductions illustrated in Section 5 about the effect of the UTSS integration on the TXOP computation with different types of multimedia applications with an increasing variable bit rate, and in the case of low VBR and CBR TSs (see Proposition 7). Therefore UTSS is outperforming in the case of highly variable VBR TSs, whereas, in the case of CBR or low VBR traffic, its contribution is limited because the centralized scheduler is suitable to efficiently serve the considered streams allocating the required resource. The UTSS simple greedy algorithm has its main strength in improving the network performance without impacting on the system overhead due to  $O(1)$  computational complexity, but, as shown, despite its simplicity, its effects are relevant.

**6.4. Packets Drop Analysis.** In the following the amount of packets dropped from the transmission queues, due to expiration of their delay bound, is evaluated. Such parameter is chosen taking into account the length of the play-out buffer of a typical consumer device and it is

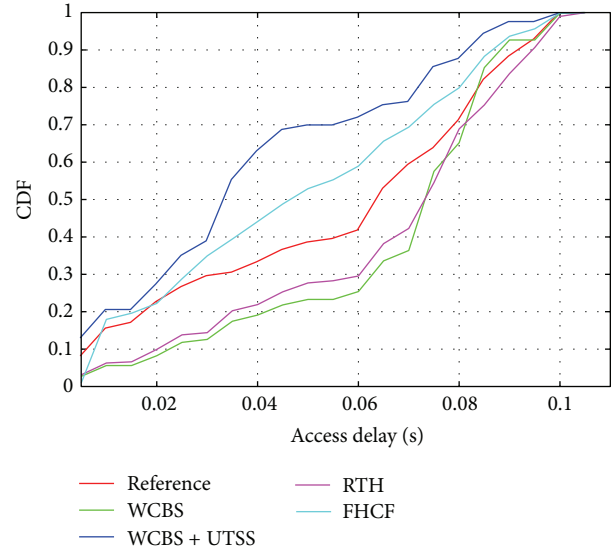


FIGURE 13: Access delay CDF of VS3 traffic stream.

tailored to the type of traffic. For this reason the analysis aims to highlight the differences between the schedulers and not the absolute value performed by each single scheduler.

Focusing on dropping rate, Figure 14 illustrates the positive impact of UTSS. Also for this performance parameter, the main consideration is that each effect has to be evaluated in comparison to the centralized scheduler (WCBS). This means that if WCBS is underperforming with respect to RTH and FHCF, this can influence the behavior of the global scheduler. First of all, looking at the considered VBR TSs in order to provide a global evaluation, UTSS is suitable to reduce the number of dropped packets since, adding a further transmission capacity, it can dispatch more traffic and reduce the waiting time in the queues. The phenomenon is more evident with respect to the mean access delay since this one is an aggregated result about all the packets of the considered TSs, whereas in the case of dropping rate, individual performances about the packets are considered. Instead, considering the CBR traffic, since WCBS is suitable to serve these TSs polling the station with the required period, UTSS does not modify the final performance, as expected. Consequently the number of dropped packets is improved by UTSS in particular when the considered TS is highly variable, since, as shown by the simulations about the access delay, UTSS, while recovering the unused time from the previous transmissions, reduces the waiting time of the scheduled TS. In this case UTSS does its best and is suitable to make the global scheduler outperform also RTH and FHCF (see VS3 and VS4). In particular, VS3 TS experiences an improvement up to 60% with respect to WCBS, RTH, and reference schedulers. This result confirms the analytical consideration about the VBR applications (see Section 5.3 and in particular about the burstiness that the network can support, (see Theorem 12) and demonstrates that UTSS is suitable to absorb data rate peaks.

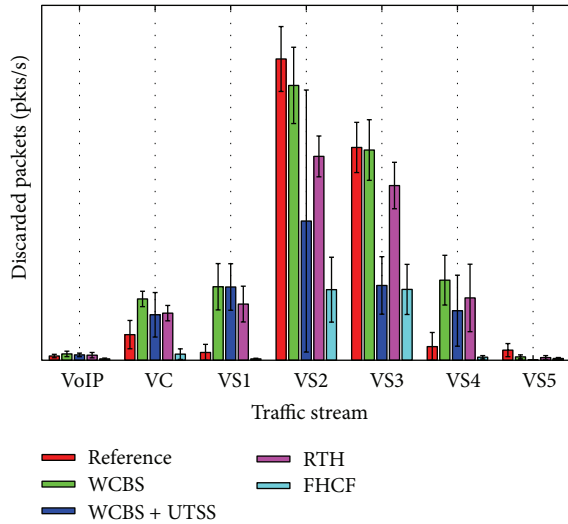


FIGURE 14: Packets drop rate.

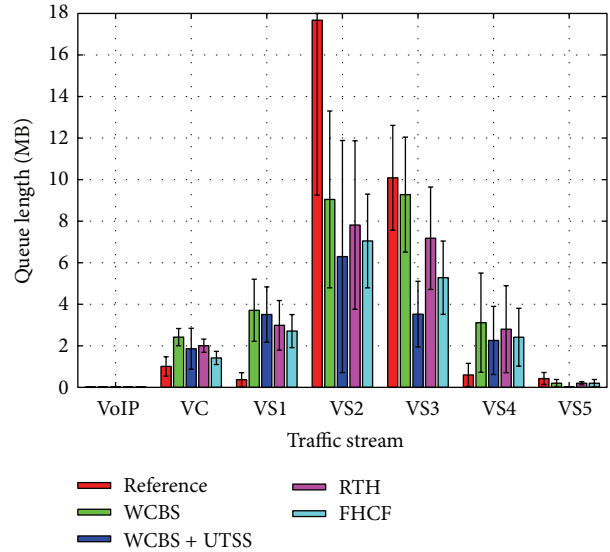


FIGURE 15: 99th percentile of HCCA buffer length.

6.5. *Transmission Queues Length Analysis.* As studied in Section 5.5, when UTSS collaborates with a centralized scheduler, there is a positive effect also on the transmission queues length. In this section the simulation is focused on the 99th percentile of queues length.

In Figure 15 it is shown that, with the exception of VoIP TS, when high VBR TSs are transmitted, as in case of VS2 and VS3, and the data rate exceeds its mean value the buffers length grows. For these TSs already WCBS, RTH and, FHCF reduce the number of enqueued packets against the performance obtained by the reference scheduler, but UTSS is able to further improve this result since the reclaimed portion of unused bandwidth helps the scheduler to transmit more frames and to shorten the occupied portion of transmission queues.

In order to deeply analyze the previous result, Figure 16 illustrates the CDF of the HCCA buffer length when VS3 traffic stream is transmitted by different schedulers. Also in this case this stream is the right candidate to evaluate the algorithms behavior when stressed by high traffic variations since it has high VBR. The simulations confirm that with high VBR TSs UTSS is able to keep the queues length shorter than the other schedulers. In particular, in the considered scenario, the greater gap between the schedulers is with a buffer length of 3 Megabytes, when UTSS shows a 66% of the probability the queue will be shorter than 3 Megabytes, while the other schedulers require about the double of this length. This is a meaningful result impacting both on the resource scheduling design as well as on the physical devices design.

6.6. *Final Considerations.* Summarizing, like other schedulers, UTSS does not aim to globally improve network performance. As proved in [12], a unique algorithm suitable to globally outperform all other solutions does not exist since each scheduler, due to its feature and scheduling rules, is tailored to face off some aspects of the complex resource management. Hence a scheduler can be focused to improve,

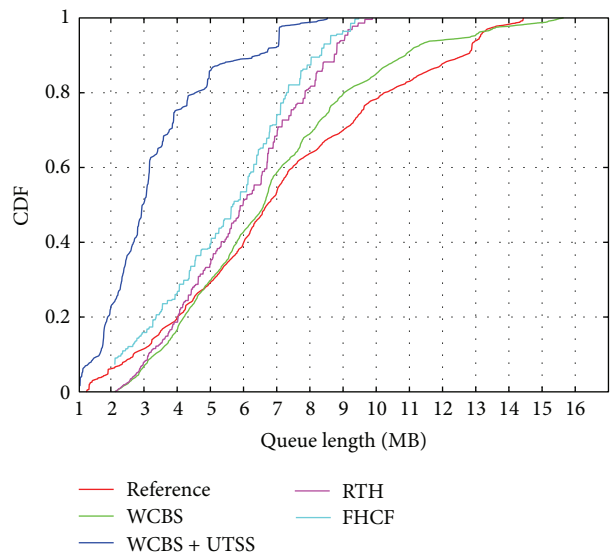


FIGURE 16: CDF of HCCA buffer length in the case of VS3 TS.

for instance, the network delay, whereas a different algorithm is better in terms of fairness or efficiency or system overhead, and so forth. In this context UTSS tries to do its best to fill in the gaps of the underlying centralized schedulers, whose features, however, continues to condition the global algorithm behavior. Indeed, the global scheduler performance suffers the imprinting of the centralized scheduling scheme. UTSS gives its contribution starting from this situation and, since it is conceived as a cooperative scheduling algorithm, the actual improvement can be perceived only comparing the performance of the centralized scheduler along with the global scheduler, obtained from the cooperation with UTSS. In particular some deductions can be made as follows.

- (i) Considering the different types of traffic, since the centralized IEEE 802.11e scheduler, reference of alternative, is suitable to serve CBR traffic, in this case the contribution of UTSS is limited but, due to its  $O(1)$  computational complexity, it does not impact on the global scheduling complexity and does not overload the system. Instead, UTSS is beneficial in the case of VBR TSs that stress the centralized schedulers.
- (ii) Focusing on the efficiency, UTSS does not modify the scheduling policy and timing, but it is able to improve the use of allocated resources by means of its reclaiming mechanism. In particular, it is beneficial especially in the case of highly VBR TSs since, dispatching more packets, it shows a local increase of null rate and throughput.
- (iii) Considering the mean access delay and taking into account that the behavior of the global scheduler is affected by that of the centralized one, the simulation results show that the cooperation with UTSS can improve the global performance. Indeed, even if the access delay of WCBS + UTSS continues to be biased by the performance of WCBS, in the case of highly VBR TSs, where WCBS fails with respect to the other alternative schedulers, when UTSS is added, the global scheduler is able to outperform RTH and FHCF.
- (iv) Focusing on packets drop rate, UTSS is suitable to reduce the number of dropped packets since, adding a further transmission capacity, it can dispatch more traffic and reduce the waiting time in the queues, especially in the case of highly VBR TSs, where UTSS does its best and is suitable to make the global scheduler outperform also RTH and FHCF.
- (v) Finally, in the case of transmission queues, UTSS is able increase their reduction obtained by the alternative schedulers.

## 7. Conclusions and Future Works

Multimedia applications with VBR traffic are a challenge for the QoS provisioning of IEEE802.11e networks: they require a flexible scheduling policy suitable to follow traffic variations.

In this paper a bandwidth reclaiming scheduling algorithm, *Unused Time Shifting Scheduler* (UTSS), is deeply analyzed. UTSS cooperates with a HCCA centralized scheduler and dynamically computes the transmission time by recovering the exceeding time from previous transmissions, providing an instantaneous resource overprovisioning.

The analytical and simulations results, comprising its comparison against some advanced QoS algorithms, highlight that it does not violate the admission control feasibility test and does not affect the centralized scheduler behavior. By means of its greedy reclaiming mechanism with  $O(1)$  computational complexity, it positively impacts on the efficiency of the resource management improving network performance. In particular, being a cooperative mechanism, its positive effects can be appreciated in comparison to the performance

of the centralized scheduler: it is suitable to improve the global scheduler behavior in terms of mean access delay, throughput, transmission queues length, bursts of traffic management, and packets drop rate. These positive effects are more relevant for highly VBR traffic.

Future works include the study of the UTSS behavior using an elastic traffic model, as well as its integration with Overboost algorithm with the aim to investigate how the collaboration of these cooperative mechanisms, each one suitable to improve the provided QoS seamlessly with respect to preexistent HCCA schedulers, may amplify their individual results. Additional object of future investigation is the comparison of UTSS with the recently proposed reclaiming mechanisms IDTH and DTH in terms of short-term and long-term fairness.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] IEEE802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," IEEE, Piscataway, NJ, USA, 2007, <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>.
- [2] A. Grilo and M. Nunes, "Performance evaluation of IEEE 802.11E," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, vol. 1, pp. 511–517, Lisboa, Portugal, September 2002.
- [3] S.-L. Tsao, "Extending earliest-due-date scheduling algorithms for wireless networks with location-dependent errors," in *Proceedings of the 52nd IEEE Vehicular Technology Conference (VTC-Fall '00)*, vol. 1, pp. 223–228, Boston, Mass, USA, September 2000.
- [4] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e wireless LAN for quality of service," in *Proceedings of the European Wireless*, vol. 1, pp. 32–39, Florence, Italy, February 2002.
- [5] J. Cowling and S. Selvakennedy, "A detailed investigation of the IEEE 802.11e HCF reference scheduler for VBR traffic," in *Proceedings of the 13th International Conference on Computer Communications and Networks (ICC '04)*, pp. 453–459, Chicago, Ill, USA, October 2004.
- [6] A. L. Ruscelli, G. Cecchetti, G. Lipari, and A. Mastropaolo, "A greedy reclaiming scheduler for IEEE 802.11e HCCA real-time networks," in *Proceedings of the 14th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '11)*, pp. 223–230, Miami, Fla, USA, November 2011.
- [7] P. Ansel, Q. Ni, and T. Turletti, "FHCF: a simple and efficient scheduling scheme for IEEE 802.11e wireless LAN," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 391–403, 2006.
- [8] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea, "Design and performance analysis of the Real-Time HCCA scheduler for IEEE 802.11e WLANs," *Computer Networks*, vol. 51, no. 9, pp. 2311–2325, 2007.

- [9] H. Fattah and C. Leung, "An overview of scheduling algorithms in wireless multimedia networks," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 76–83, 2002.
- [10] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 473–489, 1999.
- [11] A. Grilo, M. Macedo, and M. Nunes, "A service discipline for support of IP QoS in IEEE 802.11 networks," in *Proceedings of the IFIP TC6/WG6.8 Working Conference on Personal Wireless Communications (PWC '01)*, Laapenranta, Finland, August 2001.
- [12] G. Cecchetti and A. L. Ruscelli, "Real-time support for HCCA function in IEEE 802.11e networks: a performance evaluation," *Wiley Security and Communication Networks*, vol. 4, no. 3, pp. 299–315, 2011.
- [13] L. Yang, "P-HCCA: a new scheme for real-time traffic with QoS in IEEE 802.11e based networks," in *Proceedings of the APAN Network Research Workshop*, Cairns, Australia, July 2004.
- [14] G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo, "Feedback-based control for providing real-time services with the 802.11e MAC," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 323–333, 2007.
- [15] K. Y. Lee, K. S. Cho, and W. Ryu, "Efficient QoS scheduling algorithm for multimedia services in IEEE 802.11e WLAN," in *Proceedings of the IEEE 74th Vehicular Technology Conference (VTC Fall '11)*, San Francisco, Calif, USA, September 2011.
- [16] A. Arora, S.-G. Yoon, Y.-J. Choi, and S. Bahk, "Adaptive TXOP allocation based on channel conditions and traffic requirements in IEEE 802.11e networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1087–1099, 2010.
- [17] C. Kuan and K. Dimiyati, "Utilization model for HCCA EDCA mixed mode in IEEE 802.11e," *ETRI Journal*, vol. 29, no. 6, pp. 829–831, 2007.
- [18] V. A. Siris and C. Courcoubetis, "Resource control for the EDCA mechanism in multi-rate IEEE 802.11e networks," in *Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, pp. 1–6, Boston, Mass, USA, 2006.
- [19] W. K. Lai, C.-S. Shieh, and C.-S. Jiang, "Adaptation of HCCA/EDCA ratio in IEEE 802.11e for improved system performance," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 11, pp. 4177–4188, 2009.
- [20] A. L. Ruscelli, G. Cecchetti, A. Alifano, and G. Lipari, "Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks," *Ad Hoc Networks*, vol. 10, no. 2, pp. 147–161, 2012.
- [21] J. Lee, I. Shin, G. Park, W. Song, J. Kim, and J. Hong, "An efficient bandwidth reclaim scheme for the integrated transmission of real-time and non real-time messages on the WLAN," in *Computational Science—ICCS 2007*, vol. 4490 of *Lecture Notes in Computer Science*, pp. 925–932, 2007.
- [22] P. Larcheri and R. L. Cigno, "Scheduling in 802.11e: open-loop or closed-loop?" in *Proceedings of the 3rd Annual Conference on Wireless on-demand Network Systems and Services (WONS '06)*, Les Ménuires, France, January 2006.
- [23] L. Palopoli, R. Lo Cigno, and A. Colombo, "Control and optimisation of HCCA 802.11e access scheduling," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 4427–4432, New Orleans, La, USA, December 2007.
- [24] C. E. García, D. M. Pretti, and M. Morari, "Model predictive control: theory and practice: a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [25] R. Lo Cigno, L. Palopoli, and A. Colombo, "Analysis of different scheduling strategies in 802.11e networks with multi-class traffic," in *Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN '07)*, pp. 455–462, Dublin, Ireland, October 2007.
- [26] G. Cecchetti, A. L. Ruscelli, A. Mastrotaolo, and G. Lipari, "Providing variable TXOP for IEEE 802.11e HCCA real-time networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '12)*, Paris, France, April 2012.
- [27] G. Cecchetti, A. L. Ruscelli, A. Mastrotaolo, and G. Lipari, "Dynamic TXOP HCCA reclaiming scheduler with transmission time estimation for IEEE 802.11e real-time networks," in *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '12)*, Paphos, Cyprus, October 2012.
- [28] A. Grilo, M. Macedo, and M. Nunes, "A scheduling algorithm for QoS support in IEEE 802.11e networks," *IEEE Wireless Communications*, vol. 10, no. 3, pp. 36–43, 2003.
- [29] D. Ferrari and D. C. Verma, "A Scheme for real-time channel establishment in wide-area networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 368–379, 1990.
- [30] Y. Fan and C. Huang, "Real-time traffic scheduling algorithm in WLAN," in *Proceedings of the 4th Generation Mobile Forum (GMF '05)*, San Diego, Calif, USA, 2005.
- [31] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, pp. 46–61, 1973.
- [32] T. P. Baker, "Stack-based scheduling of realtime processes," *Real-Time Systems*, vol. 3, no. 1, pp. 67–99, 1991.
- [33] D. Skyrianoglou, N. Passas, and A. K. Salkintzis, "ARROW: an efficient traffic scheduling algorithm for IEEE 802.11e HCCA," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3558–3567, 2006.
- [34] J. R. Jackson, *Scheduling a Production Line to Minimize Maximum Tardiness*, Management Science Research Project, University of California, Los Angeles, Calif, USA, 1955.
- [35] C. Inan, F. Keceli, and E. Ayanoglu, "An adaptive multimedia QoS scheduler for 802.11e wireless LANs," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 5263–5270, Istanbul, Turkey, July 2006.
- [36] G. Cecchetti and A. L. Ruscelli, "Performance evaluation of Real-Time schedulers for HCCA function in IEEE 802.11e wireless networks," in *Proceedings of the 4th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '08)*, pp. 1–8, Vancouver, Canada, October 2008.
- [37] P. T. Brady, "A model for generating on-off speech pattern in two way conversation," *Bell System Technical Journal*, vol. 48, pp. 2445–2472, 1969.
- [38] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 49–59, 1992.
- [39] A. Elwalid, D. Heyman, T. V. Lakshman, D. Mitra, and A. Weiss, "Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1004–1016, 1995.
- [40] P. Wan, Z. Du, and W. Wu, "A simple and efficient mpeg-4 video traffic model for wireless network performance evaluation," in

*Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, Atlanta, Ga, USA, March 2004.

- [41] A. Adas, "Traffic models in broadband networks," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 82–89, 1997.
- [42] Y. Sun and J. N. Daigle, "A source model of video traffic based on full-length VBR MPEG4 video traces," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '05)*, pp. 766–770, St. Louis, Mo, USA, December 2005.
- [43] R. Grunenfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, pp. 284–293, 1991.
- [44] D. Jun and L. Jun, "VBR MPEG video traffic dynamic prediction based on the modeling and forecast of time series," in *Proceedings of the 5th International Joint Conference on INC, IMS and IDC (NCM '09)*, pp. 1752–1756, Seoul, Korea, August 2009.
- [45] "Network Simulator 2," 1995, <http://www.isi.edu/nsnam/ns/>.
- [46] 2005, <http://trace.eas.asu.edu/>.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

