WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
*Wirel. Commun. Mob. Comput.* 2013; **13**:95–110
Published online 20 March 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.1098

RESEARCH ARTICLE

# Selectivity function scheduler for IEEE 802.11e HCCA access mode

Najah Abu Ali[1]*, Ashraf Bourawy[2] and Hossam Hassanein[2,3]

[1] Faculty of IT, UAE University, Al-Ain, U.A.E.
[2] School of Computing, Queen's University, Kingston, Canada
[3] Department of Computer Science, King Saud University, Riyadh, Kingdom of Saudi Arabia

## ABSTRACT

In this paper, we present a scheduling algorithm that enhances the performance of the standard IEEE 802.11e scheduler for the Hybrid Coordination Function Controlled Channel Access mode. The main contribution in designing the proposed scheduler is the ability to accommodate multiple streams with different levels of Quality of Service requirements concurrently running on the same station. This is achieved by dynamically calculating the Transmission Opportunities of each active traffic stream (TS) and the appropriate Service Interval of each active station. The proposed algorithm optimizes the utilization of the scarce bandwidth resources by only polling active stations. The algorithm incorporates a selectivity function to assign polling priorities to the active streams only based on their diverse requirements and their link-attainable transmission rates. The performance of the proposed Selectivity Function Scheduler (SFS) scheme is evaluated against the standard scheduler. Simulation results show that the SFS outperforms the standard scheduler in terms of enhancing streams' throughput, reducing packet dropping ratio and maintaining high fairness amongst the admitted TS. 

**KEYWORDS**
IEEE 802.11e; HCCA; scheduling; radio resource management; QoS; HCF; service interval

***Correspondence**
Najah Abu Ali, Faculty of IT, UAE University, Al-Ain, U.A.E.
E-mail: najah@uaeu.ac.ae

## 1. INTRODUCTION

Wireless local area networks (WLAN) based on the IEEE 802.11 standard have gained an increasing popularity due to their simplicity, low cost and easy deployment. The explosive growth of multimedia and real-time applications has motivated research in Quality of Service (QoS) support for WLANs. As a step towards providing QoS support at the MAC sub-layer in WLANs, the IEEE task group e has introduced the IEEE 802.11e standard [1]. In essence, the IEEE 802.11e defines a Hybrid Coordination Function (HCF), which comprises two main medium access mechanisms, namely, Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA). The HCF synthesizes functions from legacy IEEE 802.11 with several extended functionalities and enhanced QoS-specific mechanisms. These QoS mechanisms are used in both contention and contention-free periods (CP and CFP) of the beacon interval. An access point (AP) and stations (STA) that support QoS functionality are called QoS-Access Point (QAP) and QoS-Station (QSTA), respectively.

A significant new feature of HCF is the concept of transmission opportunity (TXOP). A TXOP is an interval of time in which a QSTA is allowed to transmit a burst of data frames onto the wireless medium. A TXOP is defined by two main parameters, a starting time and a maximum duration. This feature allows the QAP to control access to the medium and precludes unpredictable transmission time of a polled QSTA. A defined parameter, called $TXOP_{limit}$, controls the maximum time a QSTA is allowed to transmit.

In this paper, we propose a scheduling scheme as a radio resource management tool for the IEEE 802.11e HCCA mode, called the Selectivity Function Scheduler (SFS). The SFS allocates adaptive TXOPs to admitted traffic streams (TS) based on actual requirements and the observed physical rate. This is achieved by facilitating a procedure for computing the number of variable-sized packets not only by considering the new arrivals, but also accounting for the packets remaining in the queue due to channel conditions. The SFS utilizes a selectivity function (SF) that assigns polling priorities to TS based on several parameters such as delay bound, channel quality, queue size and fairness.

The main merit of the SFS scheduler is the accommodation of multiple concurrent TS per QSTA with diverse QoS requirements. The performance of SFS is evaluated through simulation using OPNET modeler and compared with the standard scheduler of the IEEE 802.11e.

The remainder of this paper is organized as follows: Section 2 provides an overview of the IEEE 802.11e MAC enhancements. The SFS scheme is introduced in Section 3. Simulation results are presented and discussed in Section 4. Concluding remarks are presented in Section 5.

## 2. THE IEEE 802.11E HCCA

The main enhancement defined at the MAC sub-layer in the IEEE 802.11e is the HCF. The HCF multiplexes two access methods, namely, the Enhanced Distributed Coordination Function (EDCF) and HCCA. The HCCA access mechanism is defined for parameterized QoS support, and implemented by a centralized coordinator, called the Hybrid coordinator (HC). The HC is collocated within the QAP and has a higher priority to access the medium than the EDCF. TXOPs provide limited-duration Controlled Access Phase (CAP) for contention-free transmission. Unlike the Point Coordinator in legacy 802.11, the HC can access the medium during both CP and CFP periods. Moreover, the HC grants QSTAs different TXOPs with specified durations. Therefore, QSTAs may transmit multiple data frames within their HCCA-TXOPs, subject to the $\text{TXOP}_{\text{limit}}$.

When the HC needs to access the channel to start a CFP or a TXOP in CP, it senses if the medium is idle for one point coordination function inter-frame space (PIFS) period. If the medium is idle, the QAP transmits the first frame during the CFP or the TXOP period. In the CFP or the TXOP in CP during HCCA access mechanism, a short inter-frame space (SIFS) is required between each two consecutive frames. If the medium remains idle for a PIFS period, the QAP reclaims the channel. However, the CAP period ends when the QAP does not reclaim the channel after a PIFS period after the end of a TXOP. In HCCA access mechanism, the QoS assurance is based on the traffic specification (TSPEC) negotiated between the QAP and the QSTAs. Each QSTA can establish up to eight TS with different priorities. In order to setup a TS connection between a QSTA and the QAP, a set of TSPEC parameters is required to be sent by the QSTA to the QAP. The QAP's scheduler, in turn, allocates TXOP for each QSTA, if a TS is admitted.

### 2.1. Standard scheduler

The IEEE 802.11e outlines a reference scheduler designed for QoS requirements [1]. The HC maintains a list of QSTAs to be polled during controlled access periods. In order to be included in the polling list of the HC, a QSTA must send a QoS request frame, Add Traffic Stream (ADDTS), to initiate a connection. The QAP shall respond with an ADDTS response frame indicating whether to accept this TS connection or not. The ADDTS request and response frames include the TSPEC information element. The main components of the TSPEC element are:

- *User priority*: priority to be used for the transmission of packets, if prioritization is required. It is based on IEEE 802.1d priority levels and ranges between 0 (lowest) and 7 (highest).
- *Nominal MAC Service Data Unit* (MSDU) *size* ($L$): nominal size of the packets (bytes).
- *Maximum MSDU size* ($M$): maximum size of the packets (bytes).
- *Maximum Service Interval* (MSI): maximum time between two consecutive TXOPs allowed by the application ($\mu$s).
- *Mean data rate* ($\rho$): average bit rate for transmission of the packets (bps).
- *Minimum PHY rate* ($R$): the desired minimum PHY rate to use for this TS (bps).
- *Delay bound* ($D$): maximum delay allowed to transfer a packet across the wireless interface ($\mu$s).

The scheduling process at the QAP comprises three main steps: computing the scheduled service interval (SI), estimating the number of packets for each TS and computing the corresponding TXOPs. After these steps, the admission control unit (ACU) algorithm is invoked to check whether this TS can be admitted.

To compute the SI, the scheduler calculates the minimum of MSI for all admitted streams. If this value equals to $m$, then the scheduler shall choose a number $n$ that is a submultiple of the superframe satisfying the following inequality:

$$\text{SI} = \frac{(\text{Beacon Interval})}{n} \leq m \tag{1}$$

Upon calculating the SI, the standard scheduler estimates the number of packets that arrive in an SI time for $\text{TS}_i$. The standard scheduler uses some of the TSPEC parameters, namely, mean data rate ($\rho_i$) and the nominal MSDU size ($L_i$). Let $N_i$ denote this number of packets, then:

$$N_i = \left\lceil \frac{\text{SI} \times \rho_i}{L_i} \right\rceil \tag{2}$$

The TXOP for $TS_i$ is computed using some of the TSPEC parameters, including minimum physical rate ($R_i$), nominal MSDU size ($L_i$) and maximum allowed MSDU ($M$). Overheads in time units are also included in the computation of TXOPs, which comprises IFSs, ACK and CF-Poll frames.

$$\text{TXOP}_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O\right) \tag{3}$$

If a new stream requests admission, the scheduler performs the same steps described above. Consequently, the ACU determines that the stream can be admitted if the


Wirel. Commun. Mob. Comput. 2013; 13:95–110 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

following inequality is satisfied:

$$\frac{\mathrm{TXOP}_{k+1}}{\mathrm{SI}} + \sum_{i=1}^{k} \frac{\mathrm{TXOP}_i}{\mathrm{SI}} \leq \frac{T - T_{\mathrm{CP}}}{T} \tag{4}$$

where $k$ is the number of existing streams, $k+1$ is the newly arriving stream, $T$ indicates the beacon interval duration and $T_{\mathrm{CP}}$ is the time for contention access (EDCA).

### 2.2. Related work

Several researchers evaluated the performance of the standard scheduler [2–4] and show that it is only efficient for CBR traffic, but ineffective for VBR traffic as the standard scheduler is based on fixed transmission parameters. Hence, it is not capable of coping with the variability behaviour of the VBR traffic. Several enhancements in the literature are proposed to improve the performance of the standard scheduler. For example, the work in Refs. [5,6] focused on enhancements of the EDCA. On the other hand, the work in Refs. [7–9] address the deficiency of the polling algorithm, where all stations are polled irrespective of having packets to send or not. Enhancing polling – taking into account the packets in the queue – improves the performance of the scheduler in meeting the flows delay bounds; i.e. if the flows without backlogged packets are not polled, the flows with packets are polled faster, hence, improving the scheduler delay performance. Other proposals addressed the void in the standard scheduler to accommodate traffic variability and real-time applications' QoS requirements. Some of these enhancements comply with the IEEE 802.11e standard specifications by supporting the flows' data rate together with the delay traffic requirements [10–17]. Others depart from the standard specifications – as is the case with most proposals in the literature – and focused on one QoS requirement, namely, the delay or the packet loss, for example, Refs. [18–23], or focused on one type of application, for example video in Refs. [24–27] and voice in Refs. [28,29].

In this Section, We focus on the first category as our proposed algorithm is designed to comply with the IEEE 802.11e standard specifications in meeting the bandwidth and delay requirements, and to support several types of applications; real-time and non-real-time application. The authors of Ref. [10] propose a fair scheduling algorithm, which comprises a QAP scheduler and QSTA scheduler. The scheme utilizes the queue length estimation to assign excess time for TXOP duration to adapt to traffic fluctuation. If the time required adjusting the TXOP is insufficient, the QAP scheduler allocates the additional time between QSTA in a fair manner. However, this scheduling algorithm still considers constant SI that cannot adapt to traffic fluctuations. The work presented in Ref. [11] introduces a real-time scheduling algorithm, which aims at minimizing polling overhead while maintaining feasible complexity. This scheduling algorithm is divided into off-line and online procedures: an admission control algorithm, a timetable computation algorithm and an enforcement procedure. The key idea is to perform all complex computations off-line except for the enforcement procedure, which is the sole online activity. Since the off-line timetable is calculated when a new flow is admitted, this scheduler is appropriate only for CBR traffic and therefore, it is ineffective to handle variable data rate traffic. Same authors of Ref. [11] proposed a scheduler based on the Timed Token Protocol used in FDDI networks in Ref. [12]. The proposed scheduler allocates fixed data rate for CBR traffic and utilizes the residual bandwidth to provide VBR flows with their required data rate. The scheduler guarantees the VBR traffic their required data rate, but does not provide for the delay bound guarantees. Foronda et al. [13] presented a scheduler based on the token bucket shaper to provide for delay and data rate guarantees for the admitted flows. Foronda algorithm is executed periodically, which render the algorithm inflexible in meeting the delay bound as it may increase the latency of the scheduling time. The work presented in Refs. [14,15] is not designed on flow bases, which results in the drawback that these schedulers are not able to guarantee the delay bound for all packets of a flow. The algorithm named ARROW presented in Ref. [16] schedule flows according to buffered traffic. Information about the queue size is signalled to the AP. Hence, the algorithm achieves efficient utilization of the wireless link capacity. However, the algorithm still calculates the SI based on the traffic mean values, which may render the algorithm inefficient for VBR traffic.

The aforementioned proposals that comply with the IEEE 802.11e standard can be further categorized into two categories. The first category assumes that all flows of the same application type (e.g. VoIP) have the same level of QoS requirements. For example, they provide for a single delay bound requirement for all real-time traffic sessions. Hence, they are not able to provide the required QoS guarantees per each flow, for example Ref. [23]. The Second category may, for example, provide for multi delay bounds of one application type. However, it assumes that each QSTA can accommodate one real-time session at a time. However, in a practical environment, a QSTA may be required to simultaneously serve multiple real-time sessions with different delay bounds and data rates QoS requirements. Moreover, almost all proposals overlook the fact that in practical networks the real-time traffic is mixed with non-real-time traffic (namely TCP applications such as FTP). Accommodating non-real-time traffic equivalently requires providing slack level of QoS guarantees. In this paper, we propose a scheduler that is able to provide for different levels of QoS requirement, accommodate more than a single session at the QSTA, and cater for real-time and non-real-time QoS requirements.

## 3. THE SELECTIVITY FUNCTION SCHEDULER

We propose a scheduler for IEEE 802.11e to accommodate for the voids in the standard IEEE 802.11e by incorporating the following functionalities:

- Accommodates different types of traffic including CBR and VBR traffic and provides diverse QoS assurance.
- Considers the channel conditions and QoS requirements of different traffic types.
- Provides QoS assurance to users based on their individual service requirements. The proposed SFS scheme incorporates several components and parameters in order to provide service differentiation and scheduling for the admitted streams. The SFS considers the delay bounds of real-time applications, queue size of each TS and channel condition of QSTAs with active traffic flows.
- Computes the number of packets for each TS, calculates adaptive TXOPs and variable SI to assign polling priorities to TS based on their requirements.
- Incorporates a fairness mechanism in order to balance the tradeoff between maximizing system throughput based on users with good channel conditions and precluding starvation of users with bad channel conditions.

The proposed scheme is discussed below.

## 3.1. Computing number of packets

An essential part of allocating TXOPs for admitted TS is the computation of the number of packets in TS queues. The SFS differentiates the number of packets into two main components: packets remaining in the queue from previous TXOP, and new packets arrived during the scheduled SI.

### 3.1.1. Packets remaining in the queue.

In order to determine the number of packets remaining in the TS queue from previous TXOP, the SFS calculates the probability of error based on channel conditions. According to the IEEE 802.11e standard, the queue size is included in the data frames sent by each QSTA to the QAP. The queue size field indicates the amount of buffered traffic for a given TS at the QSTA. As depicted in Figure 1, the QAP is aware of the queue size of each TS at the beginning of each TXOP, denoted by $Q_b$ in the figure. This $Q_b$ consists of packets remaining in the queue from previous TXOP and new arrived packets during an SI period.

To compute the number of packets remaining in the queue at the end of a previous TXOP out of the $Q_b$, the SFS scheme computes the probability that a packet is received correctly using the Bit Error Rate (BER) of the wireless channel as follows:

$$P\{\text{Packet is received correctly}\} = (1-\text{BER})^L \tag{5}$$

where $L$ is the packet size in bits.

We can compute the probability that $k$ packets remained in the queue out of $N_i$ packets that were in the queue $i$ at the beginning of TXOP$_i$ as follows:

$$P\{F = k|N_i\} = \binom{N_i}{k}(1-P_s)^k P_s^{N_i-k} \tag{6}$$

where $P_s$ is the probability of receiving a packet successfully, $F$ is a binomial distributed random variable denotes the number of packets remaining in the queue.

### 3.1.2. New arrivals in an SI interval.

The SFS records the queue length of each TS in every SI to compute the number of new arrivals. The queue length is recorded at the beginning of each TXOP and is denoted by ($Q_b$). The value of $Q_b$ comprises two components: the packets remaining in the queue from previous TXOP and the new arrived packets. Since the SFS scheme already computed the packets remaining in the queue from previous TXOP, the new arrivals can be calculated as follows:

$$N_r = Q_b - N_f \tag{7}$$

where $N_f$ is calculated from Equation (6).

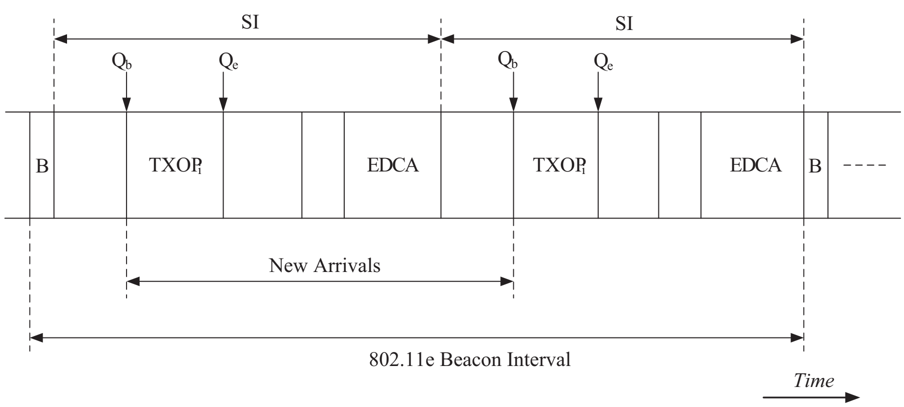



**Figure 1.** Beacon interval analysis.


Wirel. Commun. Mob. Comput. 2013; 13:95–110 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

The SFS scheme uses an exponential weighted moving average to compute the average value of the new arrived packets.

$$\overline{N}_{r_{(t+1)}} = (1-\alpha)\overline{N}_{r_{(t)}} + \alpha N_{r_{(t+1)}} \quad (8)$$

where $\alpha$ is a weighting factor.

Finally, the number of packets $N_i$ that will be in the $TS_i$'s queue at the beginning of TXOP is estimated as follows:

$$N_i = \overline{N}_{r_{(t+1)}} + N_f \quad (9)$$

## 3.2. TXOP computation

Physical rate of a QSTA plays a crucial role in computing TXOPs for TS. The SFS scheme uses the observed PHY rate ($R_{obs}$) in computing adaptive TXOPs. During communication with QSTAs, the QAP has feedback information about the PHY rate of each QSTA, which is used in TXOP's computation.

$$\mathrm{TXOP}_i = \max\left(\frac{N_i \times L_i}{R_{i_{obs}}} + O,\ \frac{M}{R_{i_{obs}}} + O\right) \quad (10)$$

where $L_i$ is the frame size in bits, $R_{i_{obs}}$ is the observed physical rate of $TS_i$, $O$ is the overhead in seconds and $M$ is the maximum frame size in bits.

The SFS scheme computes TXOPs for each individual TS at the beginning of each SI. This allows the SFS scheme to opportunistically adapt to the variations in the number of packets generated and the observed PHY rate. These adaptive TXOPs help utilizing the network resources efficiently by allocating accurate time durations needed by different applications.

## 3.3. Selectivity function

The SFS scheme makes selections from the admitted TS that will be polled in the current SI. All computations are done at the beginning of each SI. The SF coordinates the polling mechanism and utilizes the WLAN resources during the HCCA access. The SFS scheme incorporates the SF in order to assign priorities to admitted TS and reorder their polling times. This mechanism can effectively enhance the system performance and provide QoS assurance for different traffic types.

### 3.3.1. Selectivity function features.

The main features of the proposed SF are:

- Accommodating VBR traffic fluctuations in terms of channel quality and variable data rate.
- Incorporating a fairness measure based on achieved throughput by TS to prevent starvation of TS with degraded channel quality.
- Providing QoS assurance for different traffic types by incorporating several important parameters, such as delay bound, queue size and channel quality.
- Considering a system priority factor for each TS type to allow the service provider to assign different priorities based on user preferences.
- Enabling the SFS scheme to select TS to be polled in the current SI and to decide the SI length.

### 3.3.2. Selectivity function formulation.

Before proceeding with describing the formulation of the SF, it is imperative that we make the following definitions:

- $t_i$: Queueing delay of the packet at the head of TS $i$'s queue.
- $\mathrm{DB}_i$: Delay bound as specified in the TSPEC element for TS $i$.
- $t_i/(\mathrm{DB}_i)$: This ratio is used per TS to account for delay bound constraints for each TS, which allows the QSTA to accommodate different streams with different delay bounds. Since voice and video traffic are delay-bounded, this ratio component is included in the SF formula when computing SF values for voice and video. However, this ratio is not included when computing SF for FTP streams.
- $R_{i_{obs}}$: The observed PHY rate of TS $i$.
- $R_{max}$: The maximum observed PHY rate among all admitted TS.
- $\left(R_{i_{obs}}\right)/R_{max}$: This ratio is used in the SF computation to enhance the system performance in terms of throughput. Incorporating the channel quality in the SF is considered as an advantage for users with good channel quality. In other words, the better the channel quality, the higher the value of SF is, and a higher priority will be assigned to this TS.
- $Q_i$: Queue size of TS $i$.
- $Q_{max}$: The maximum queue size among all admitted streams.
- $Q_i/Q_{max}$: This fraction is used in the SF formula to account for queue size parameter. This renders TS with large queue sizes having high SF values, which in turn increases their probability of polling.
- $\delta_i$: A priority factor for different TS and is a system parameter where the provider can set different priorities to different users based on users' preferences.
- $S_{av_i(t)}$: The average throughput for TS $i$ at the beginning of $t$th SI. It is computed using the exponential smoothed filter as follows:

$$S_{av_i(t)} = \begin{cases} (1-\alpha)S_{av_i(t-1)} + \alpha S_{i_{(t)}}, \\ \quad \text{if } \mathrm{TS}_i \text{ is selected} \\ (1-\alpha)\ S_{av_i(t-1)}, \\ \quad \text{Otherwise} \end{cases}$$

  where $S_{i_{(t)}}$ is the throughput of TS $i$ at the $t$th SI.
- $RQ_{\mathrm{DR}_i}$: The required data rate requested by TS $i$.

- $e^{-\left(\frac{Sav_i(t)}{RQ\mathrm{DR}_i}\right)}$: A fairness measure incorporated by the SF. This exponential function serves as a fairness measure between the admitted TS in terms of achieved throughput. Since the denominator of this fraction is fixed and is obtained through the TSPEC element, the fraction depends only on the numerator, which is the achieved average throughput by TS $i$. When the achieved throughput is low, this exponent fraction increases, which leads to increase in the value of the SF.

Having stated all the important factors and parameters for constructing the SF, we can incorporate these factors to produce an SF formula comprising three cases for different traffic types. The first case is designated for voice (VO) traffic, the second case considers video (VI) traffic, and the third case is for File Transfer Protocol (FTP) traffic. The SF is given by the following formula:

$$SF_i(t) = \begin{cases} \left[\frac{t_i}{\mathrm{DB}_i} + \frac{R_i}{R_{\max}} \cdot e^{-\left(\frac{Sav_i}{RQ\mathrm{DR}_j}\right)}\right] \cdot \delta_i & , \quad i \in VO;\ t_i < \mathrm{DB}_i;\ R_i > 0 \\ \left[\frac{t_i}{\mathrm{DB}_i} + \frac{R_i}{R_{\max}} \cdot \frac{Q_i}{Q_{\max}} \cdot e^{-\left(\frac{Sav_i}{RQ\mathrm{DR}_i}\right)}\right] \cdot \delta_i & , \quad i \in VI;\ t_i < \mathrm{DB}_i;\ R_i > 0 \\ \left[\frac{R_i}{R_{\max}} \cdot \frac{Q_i}{Q_{\max}} \cdot e^{-\left(\frac{Sav_i}{RQ\mathrm{DR}_i}\right)}\right] \cdot \delta_i & , \quad i \in \mathrm{FTP};\ R_i > 0 \end{cases} \tag{11}$$

The SF, $\mathrm{SF}_i(t)$, given in Equation (11) meets the objectives stated in Section 3.3.1 as follows: the delay bound is taken into account in the fraction $t_i/\mathrm{DB}_i$, maximizing the throughput is expressed in the ratio $(R_i/R_{\max})$, the fairness measure is included and given by the exponent, $e^{-\left(\frac{Sav_i(t)}{RQ\mathrm{DR}_i}\right)}$ and the queue size, given by the normalized value $(Q_i/Q_{\max})$, is considered to reduce dropping ratio and to give higher priority to streams with large queue sizes. Note that Equation (11) is mainly based on proportional fairness formulas.

Figure 2 depicts numerical SF values plotted against queuing delay in order to evaluate the formulation of the SF function. We plot three curves for each traffic type, where each curve represents a different value of the fraction $\left(\frac{Sav_i(t)}{RQ\mathrm{DR}_i}\right)$, the values are set to 0.1, 0.5 and 1. $\delta_i$ is fixed at 1 for voice (VO) streams, at 0.7 for video (VI) flows and at 0.5 for FTP flows. Other factors are fixed to 1. The effect of the fraction $R_i/R_{\max}$ on SF is illustrated in Figure 3, where the value of this fraction is set to 0.1, 0.5 and 1, respectively. Note that the legend key VO 0.1 means the curve is plotted considering the value 0.1 for voice traffic. This applies to the other two curves; the video (VI) and FTP.

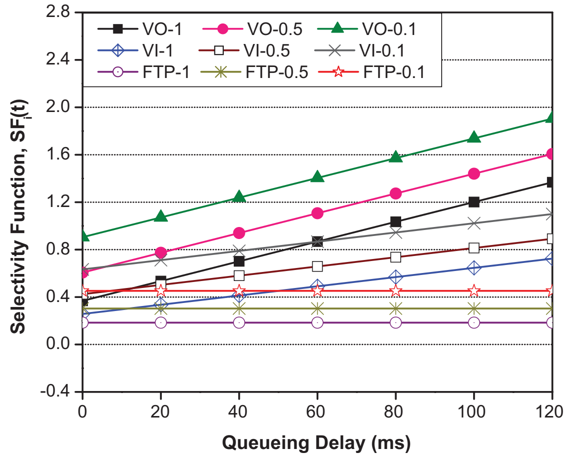



**Figure 2.** SF vs. queueing delay and fairness measure.

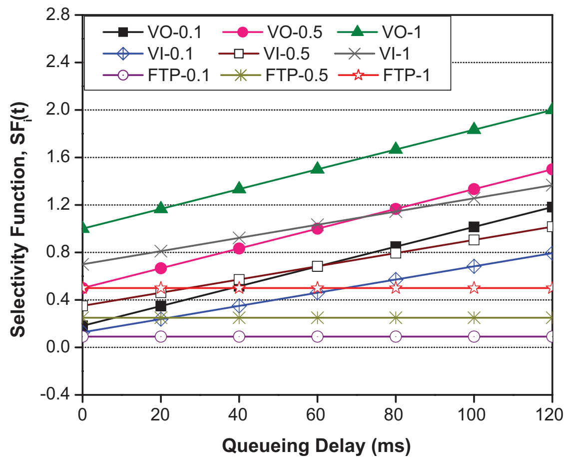



**Figure 3.** SF vs. queueing delay and channel quality.

## 3.4. Computing the SI length

The computed number of packets $N_i$, transmission opportunity $\mathrm{TXOP}_i$ and the $\mathrm{SF}_i(t)$ for traffic stream $i$ $(\mathrm{TS}_i)$ are made available for the SFS scheme to compute an appropriate SI length. Contrasting the standard scheduler, which polls all TS, the SFS scheme and based on the SF values polls a subset of the TS during the current SI. This polling mechanism ensures to poll QSTAs that have pending data frames to transmit, which best utilizes the scarce bandwidth resources.

The SFS constructs a five-column list that contains all admitted streams TS. Each entry in this list corresponds to a TS's parameter that includes: node ID (NID), traffic stream ID (TSID), traffic stream type $(\mathrm{TS}_{\mathrm{Type}})$, the computed TXOP for this TS and the SF($t$) value for this TS. The list is then sorted in a descending order based on the SF values of all TS. Note that the complexity of the SFS scheduler depends mainly on the sorting algorithm employed in realizing the SFS. The computation procedure of variable SI is as depicted in Figure 4.


*Wirel. Commun. Mob. Comput.* 2013; **13**:95–110 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

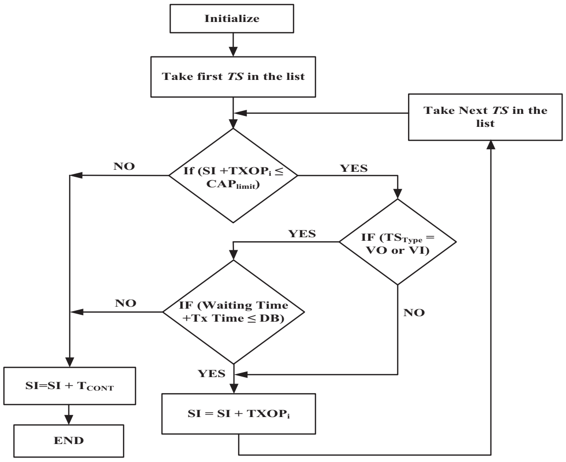



**Figure 4.** Service interval computation.

Initially, the SFS chooses the first TS in the list, which corresponds to the highest priority, and checks if the available resources can accommodate this TS. If the time reserved for controlled channel access is exceeded, this ends the loop and sets the SI length. Otherwise, if $CAP_{limit}$ for the current SI is not violated, the SFS checks the type of this TS. If the selected TS is of type voice or video, the SFS verifies if the delay bound is not exceeded. Violating the delay bounds results in terminating the current loop, consequently determining the SI length. If the delay bound is not violated, this TS is considered in the polling list. If the selected TS is of type FTP, this stream is included in the polling list because FTP streams have no delay constraints, given that the $CAP_{limit}$ is not exceeded. Upon choosing TS to be polled in the current SI, the time for contention-based access mechanism ($T_{CONT}$) is added to the SI length. $T_{CONT}$ is the time required to send one maximum MSDU as specified by the standard. Finally, the QAP polls all QSTAs with TS in the polling list.

## 4. PERFORMANCE EVALUATION

The SFS scheme is implemented and evaluated using the OPNET Modeler 14.5 (http://www.opnet.com). To compare the performance results, we consider the IEEE 802.11e based on the 802.11b physical layer. To realize the SFS scheme, we modified the IEEE 802.11e scheduler in the OPNET modeler to include the extended aforementioned functionalities of the SFS, such as computing the number of packets, SF and polling mechanism procedure. The beacon interval generated by QAP is set to 200 ms.

### 4.1. Simulation setup

An infrastructure network is configured comprises a QAP and a number of QSTAs. The number of QSTAs varies between 5 and 15 depending on the simulation scenario.

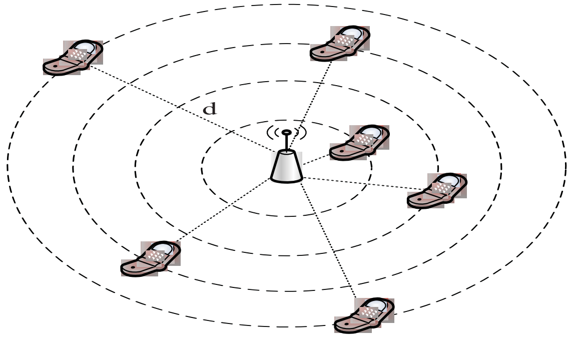



**Figure 5.** Simulation model - setup.

**Table I.** Simulation parameters.

| Parameter | Value |
|---|---|
| Physical layer | IEEE 802.11b |
| Bandwidth | 2.4 GHz |
| Node placement | Random |
| Beacon interval | 200 ms |
| Simulation time | 25 seconds |

**Table II.** PHY and MAC parameters.

| Parameter | Value |
|---|---|
| SIFS ($\mu$s) | 10 |
| PISF ($\mu$s) | 30 |
| DIFS ($\mu$s) | 50 |
| Slot time ($\mu$s) | 20 |
| MAC header size (Octets) | 32 |
| QoS-ACK frame size (Octets) | 16 |
| QoS-CF-Poll frame size (Octets) | 36 |
| PLCP header length (bits) | 48 |
| PLCP preamble length (bits) | 144 |
| PHY rate (Mb/s) | 11 |
| Minimum PHY rate (Mb/s) | 2 |
| CCA time ($\mu$s) | 15 |
| TxRxTurnaround time ($\mu$s) | 15 |

The QSTAs are randomly distributed around the QAP with different distances ranging between 5 and 250 m forming a star topology as shown in Figure 5. We utilize the channel model provided by the OPNET modeler.

The network system parameters used in our experiments are listed in Table I and the MAC and PHY layer parameters are summarized in Table II.

### 4.2. Transmission modes

We use four transmission modes in our implementation as shown in Table III, in conformance with the IEEE 802.11b PHY layer. These transmission modes are then used to compute the BER based on the signal to noise ratio and the observed physical rate by the QAP.

**Table III.** IEEE 802.11b transmission modes.

| Data rate ($R$) | Code length (type) | Modulation |
|---|---|---|
| 1 Mbit/s | 11 (Barker sequence) | DBPSK |
| 2 Mbit/s | 11 (Barker sequence) | DQPSK |
| 5.5 Mbit/s | 8 (CCK) | DQPSK |
| 11 Mbit/s | 8 (CCK) | DQPSK |

**Table IV.** Traffic specifications.

| Application | Voice | Video-H.261 | FTP |
|---|---|---|---|
| Packet size (bytes) | 160 | 660 | 1200 |
| Minimum rate (kb/s) | 25 | 150 | 250 |
| Mean rate (kb/s) | 64 | 200 | 350 |
| Maximum SI (ms) | 25 | 30 | 35 |
| Delay bound (ms) | 120 | 180 | $\infty$ |

## 4.3. Traffic model

We implement three different traffic sources, each one represents a different priority class. The three traffic types are considered as follows: a high priority voice traffic, a medium priority VBR video traffic and a low priority FTP data traffic. The voice flows are modelled by ON-OFF sources with parameters corresponding to a typical phone conversation using OPNET built-in traffic generators. VBR video traffic flows are modelled as video-H.261 sources using OPNET traffic generators. FTP flows are also obtained using the built-in generators in OPNET network simulator. The TSPEC of each individual flow are summarized in Table IV.

## 4.4. Simulation variables

In order to study the performance of the SFS scheme against the standard scheduler, we considered the following simulation variables.

- *Network Load*: Varying the network load by changing the number of TS. This variable is studied under three different scenarios: the effect of voice streams, video streams and FTP streams. However, the results included in this paper are the results of the video streams scenario – as a sample of the three scenarios' results.
- *Throughput Averaging Window*: Varying the averaging window size used in computing the average throughput in the fairness measure of the SF.
- *Number of Associated QoS Stations*: Increasing the number of associated QoS stations, while fixing the network load. The objective of this variable is to evaluate the effect of overheads caused by increasing the number of QSTAs on the performance of SFS.
- *Retransmission Limit*: This variable is used to study the impact of the packet retry limit on the performance of the SFS. Computation of the number of packets depends on BER and channel conditions, while, the packet retry limit controls whether this packet is transmitted or discarded, consequently, impacts the average throughput and delay.

## 4.5. Performance evaluation metrics

The performance of the SFS scheme is evaluated using the following metrics: *Throughput* ($S$)*:* the number of bits successfully received by the potential receiver per time unit. The value of throughput is expressed in kbps.

*Packet Loss Ratio* (PLR): The per cent of packets dropped from the queue out of all the packets enqueued.

*Average Delay (Latency)* ($\overline{D}$): the average of delays of all packets from the same class. The delay is calculated as the time difference between the reception of this packet by its potential receiver and the enqueue time. The value is represented in ms.

*Fairness Index*: Fairness is measured between users of the same traffic class (intra-class fairness) and as an overall measure between all traffic classes (inter-class fairness). We use Jain's fairness index [30] to compute inter-class fairness, whereas we use Min–Max fairness index to calculate intra-class fairness. The Min–Max fairness index is very sensitive to service degradation and service unfairness [31]. The fairness indices are computed as follows:

$$\text{Jains Fairness Index} = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \times \sum_{i=1}^{n} x_i^2} \tag{12}$$

$$\text{Min-Max Fairness Index} = \frac{\min\{x_i\}}{\max\{x_i\}} \tag{13}$$

In order to compute the inter-class fairness using Jain's index, we use normalized throughput for $x_i$. The throughput of an individual TS ($S_i$) is normalized with respect to the Required Data Rate ($RQ_{\mathrm{DR}_i}$) of this TS, i.e. $x_i = S_i/RQ_{\mathrm{DR}_i}$. Min–Max fairness calculates fairness between the TS with the minimum throughput ($\min\{x_i\}$) and the TS with the maximum throughput ($\max\{x_i\}$) in the same class.

## 4.6. Simulation results

### 4.6.1. Network load.

In this scenario, we evaluate the effect of varying the network load by changing the number of admitted video streams, while fixing the number of voice and FTP streams, each to five streams. The number of VBR video streams ranges between 7 and 12. Seven QSTAs are associated with a QAP. Simulation results are shown in Figure 6.

Throughput of the three traffic types is shown in Figure 6a. Both schedulers succeed in meeting the required voice data rate due to the highest priority of voice streams, low data rate requirements and small packet sizes of the

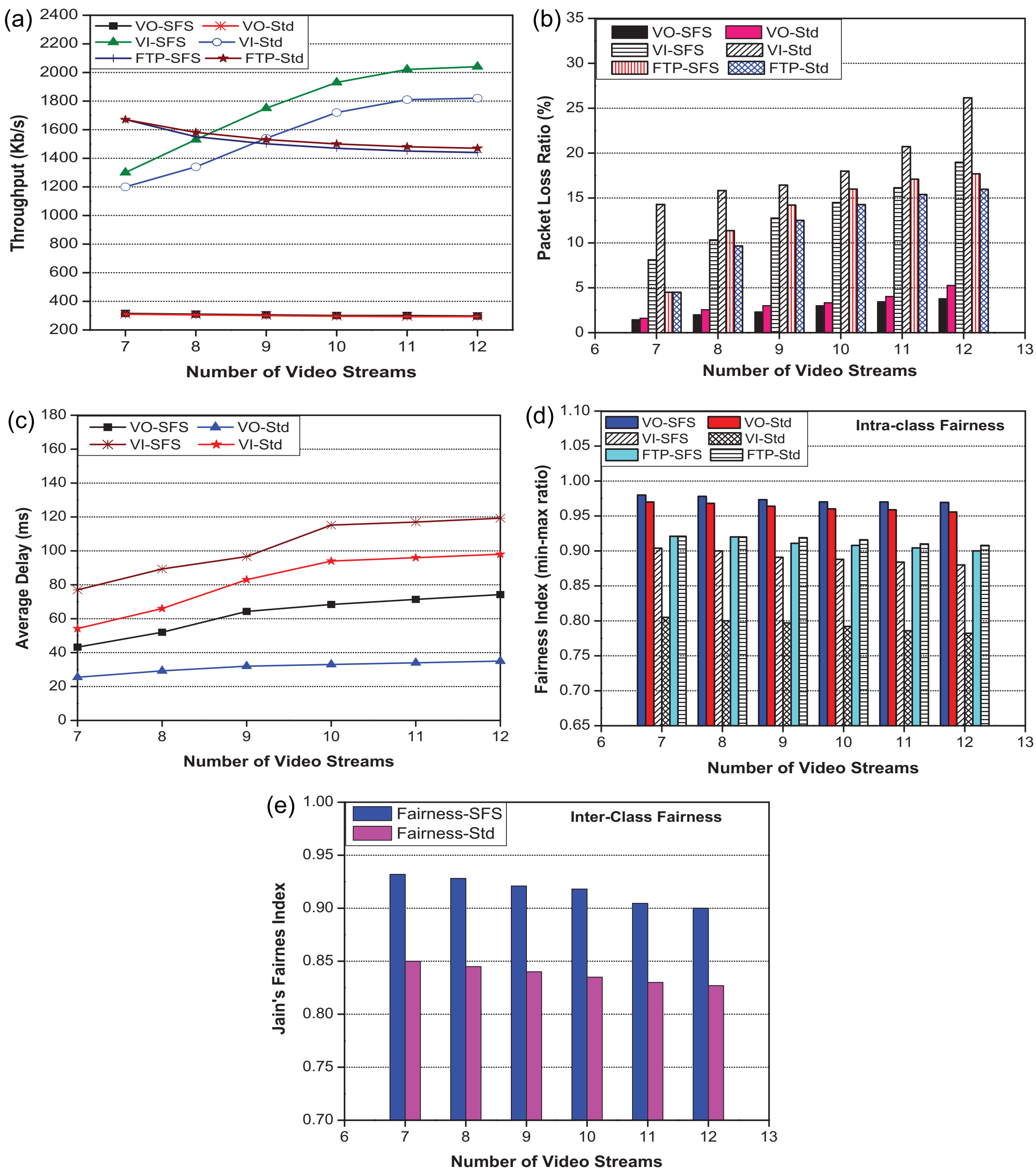



**Figure 6.** The effect of varying the network load while keeping the number of associated QoS stations fixed. (a) Number of video streams against the average throughput in kbps; (b) number of video streams against the packets drop ratio; (c) number of video streams against the average delay in ms; (d) number of video streams against the Min-Max fairness index; and (e) number of video streams against the Jain's fairness index.

voice TS. It is noted that the SFS outperforms the standard scheduler in having higher throughput for VBR video traffic as the number of video streams increases. This result is expected because the SFS considers the packets remaining in the queue and adapts to channel conditions fluctuations and video traffic burstness.

Although both schedulers achieve high and almost the same throughput for FTP traffic, the FTP throughput experiences degradation as the number of video streams increases. The degradation in the SFS performance is caused by the SF as it gives higher priority to video streams over FTP streams. The level of degradation can be controlled by the network administrator by changing the parameter $\delta_i$ in the SFS function.

The SFS succeeds in maintaining lower packet dropping ratio for all TS as shown in Figure 6b. We observed that VBR video traffic experiences high dropping ratio in the standard scheduler due to the inefficient support to the VBR traffic in the standard scheduler computations.

The SF assigns high polling priority to streams with high queuing delay. Hence, The SFS results in high average delay for voice and video traffic without exceeding their delay bounds as shown in Figure 6c. The SF defers scheduling of streams as long as their delay bound is not violated. On the other hand, the standard scheduler indicates high average delay of video traffic, which corroborates that packets remaining in the queue from previous TXOPs need to be considered. However, the average delay in both schedulers experiences a lower increase when 10 or more video streams are admitted, because of the coupling effect of delay and packet dropping. When the dropping ratio increases, the packets remaining in the queue afterwards practice lower delay, which limits the increase in the average delay.

It is desirable that streams from the same traffic class should be treated equally. Hence, the Min–Max fairness measure is used to calculate the intra-class fairness due to its high sensitivity towards service unfairness and degradation. Figure 6d shows that the SFS scheme achieves better fairness for all classes over the standard scheduler. The standard scheduler does not differentiate among flows of the same class. on the other hand, the SFS scheme incorporates fairness parameters in calculating the flows' priorities among flows of the same class. Hence the fairness performance of SFS outperforms that of the standard scheduler.

Jain's fairness index is used to calculate the inter-class fairness among TS. We observe in Figure 6e that the SFS scheme maintains higher fairness than the standard scheduler, because the differences in average throughput are higher in the standard scheduler, especially among video streams. The standard scheduler provides differentiated QoS for the different traffic classes. However, it gives the voice traffic strict priority without incorporating fairness measure to accommodate other traffic types. The SFS considers a fairness measure in the SF to allow TS experiencing low average throughput to have higher priority over TS with high average throughput to avoid degrading the service of these TS.

### 4.6.2. Averaging window size.

In this scenario, we evaluate the fairness measure incorporated by the SF, which is realized by varying the averaging window size in computing the average throughput used in the SF. In particular, we vary the value of $\alpha$ used in computing the average throughput achieved by each TS. In this experiment, seven QSTAs are associated with a QAP. Each QSTA sends three TS simultaneously, voice, video and FTP. The weighting parameter $\alpha$ is being varied between 0.02 and 0.22, where 0.02 represents large averaging window and 0.22 represents small averaging window. These values correspond to how much weight is considered for the history through varying the averaging window size. Furthermore, these numbers reflect how many samples are taken into consideration, while each sample reflects the average throughput achieved in that SI.

Simulation results of this scenario is shown in Figure 7. The three traffic types in the standard scheduler are not affected by changing the averaging window because it is not considered by the standard scheduler. The VBR video traffic scheduled by the standard scheduler shows some fluctuations, however, this behaviour is due to the VBR burstness. The SFS shows improvement in video and FTP throughput when the averaging window increases as shown in Figure 7a, because when the average window is small, more weight is given to the current attainable throughput by the TS. This affects the fairness measure negatively in the SF by decreasing its value, $\left(S_{av_i}(t)/RQ_{\mathrm{DR}_i}\right)$. Hence decreasing the probability of this TS to be scheduled in the current SI. When the value of $\alpha$ is 0.1 or less, i.e. large averaging window, the video and FTP throughput remains steady, since the average throughput will not be affected much by the long throughput history.

The standard scheduler shows in Figure 7b high dropping ratio among video and FTP streams due to overlooking the effect of the burstness of VBR traffic and the variations of the channel quality. Conversely to the standard scheduler, the SFS succeeds in reducing the packets dropping ratio of video and FTP streams. When the history of average throughput is given more weight in the fairness measure of the SF (small values of $\alpha$), the video packets drop ratio decreases, which increases the throughput.

Video traffic in Figure 7c has high average delay – but within the delay bounds required – for the SFS scheme when large averaging windows are considered, because of the ratio $\left(S_{av_i}(t)/RQ_{\mathrm{DR}_i}\right)$ in the SF, which allows the SF to give more realistic priority values to different TS based on their achieved average throughput. This also explains the improvement in video throughput under large averaging windows, i.e. small values of $\alpha$.

Both schedulers show high intra-class fairness index for voice traffic, Figure 7d. The standard scheduler indicates almost steady voice fairness index because varying the averaging window does not affect voice throughput and the standard scheduler lacks the fairness measure in its computations. For SFS, the fairness index of voice traffic increases with decreasing the averaging window. As the average window size decreases, the SF prioritize voice traffic over other types of traffic, because the history of throughput is not given much weight.

Video intra-class fairness index in the SFS scheme increases with the increase of average throughput consideration, i.e. averaging window. Video traffic in the standard scheduler shows lower fairness index due to the VBR behaviour and the absolute priorities in the standard scheduler where voice streams are always dominant. FTP intra-class fairness index in the SFS encounters improvement with the increase of averaging window (low values of

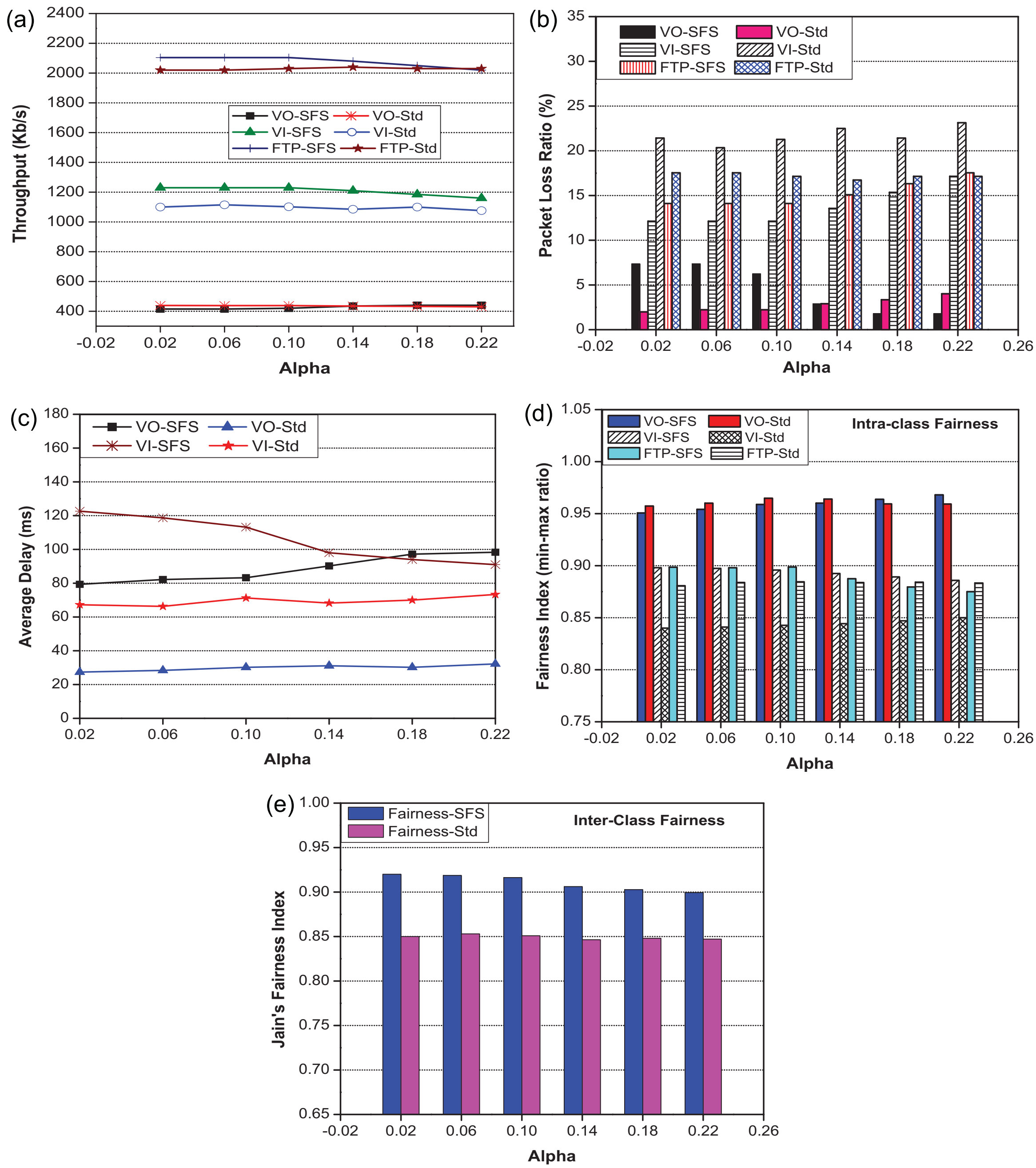



**Figure 7.** The effect of varying the averaging window size while keeping the network load fixed. (a) Averaging window size against the average throughput in kbps; (b) averaging window size against the packets dropping ratio; (c) averaging window size against the average delay; (d) averaging window size against the min-max fairness index; and (e) averaging window size against the Jain's fairness index.

$\alpha$). This indicates the effect of fairness measure incorporated in the SF, which increases the chances for FTP flows when their average throughput is low.

The SFS shows higher inter-class fairness index than the standard scheduler. However, the fairness index decreases slightly when small averaging windows are considered as shown in Figure 7e.

### 4.7. Number of QoS stations

The objective of this experiment is to study the effect of increasing the number of QoS stations. Increasing the number of QoS stations leads to increasing the overheads caused by the polling mechanism and inter-frame spacing times. In this experiment, we fix the load of the network and increase the number of QoS stations at each time. The ratio between the three traffic types is fixed to 2:2:1 (voice:video:FTP).

In general, the throughput of all traffic types degrades with increasing the number of QSTAs as shown in Figure 8a. The degradation of throughput is caused by the overheads of polling frames (QoS-CF-Poll), ACK frames and SIFS times. Voice traffic in the standard scheduler achieves a slightly higher throughput than that of the SFS, because the standard scheduler grants the highest priority to voice streams without considering fairness among the different TS. However, both schedulers succeed in providing voice traffic with its required data rate. The SFS scheme manages to reduce the overheads effect by computing the TXOPs dynamically at the beginning of each SI, which are adaptive to the burstness of the video traffic. On the other hand, the standard scheduler computes the TXOPs calculations statically. Consequently, the video traffic throughput experiences a higher degradation than that scheduled by the SFS scheme.

In Figure 8b, the voice packet drop ratio of the SFS is slightly higher than the drop ratio of the standard scheduler because the SFS scheme uses a fairness measure to balance the opportunities among different traffic types. The standard scheduler, however, associates voice traffic with the highest priority and does not include a fairness measure for other traffic types.

The SFS shows higher FTP drop ratio than the standard scheduler, because the SF of the SFS schedules voice and video flows based on their delay bounds besides their other QoS requirements. Consequently, the FTP streams have a lower probability of being scheduled because they are delay tolerant. In addition, FTP traffic is characterized by large packet sizes which increases the probability of dropping those packets due to channel conditions.

The voice average delay of the standard indicates in Figure 8c a low increase, while the voice average delay in the SFS encounters a higher delay but within the delay bounds. The SF chooses streams based on their delay bound requirement, which results in increasing the average delay of voice traffic in the SFS. On the other hand, video average delay indicates a high increase in both schedulers. The SFS scheme defers transmission of video traffic as long as their delay bound is not imminent, which explains the high average delay. The high delay encountered by the standard scheduler can be referred to the miscalculation of number of packets. In addition, the overheads caused by increasing the number of QoS stations affect the average delay as these overheads consume portion of the TXOPs, which results in increasing the number of packets dropped.

Intra-class fairness index shows different values for different traffic types as shown in Figure 8d. Voice streams achieve high fairness, although it decreases with the increase of number of QSTAs associated with QAP. This is due to different channel quality values for different QSTAs, where in this case streams achieve different throughput, which in turn decreases the fairness index. Video streams in the standard scheduler indicate low intra-class fairness index, because of the variability in channel quality and the fact that the duration of TXOPs are the same for the same traffic type even though it is associated with different QSTAs (that have different channel quality). Both schedulers indicate high fairness among FTP streams due to the low number of admitted FTP flows.

The inter-class fairness, Figure 8e shows that the SFS scheme succeeds in achieving higher fairness index than the standard scheduler, because the SFS deals with fairness based on TS rather than QSTAs. The degradation in video throughput in the standard scheduler affects the inter-class fairness, which results in lower Jain's fairness index than the SFS scheme.

## 4.8. Packet retransmission limit

The objective of this experiment is to study the performance effect of varying the packet retry limit. The packet retry limit, in essence, affects the computation of the number of packets, specifically the packets remaining in the queue at the end of previous transmission opportunity. The effect of packet retry limit on TS throughput is illustrated in Figure 9a. The retry limit has almost no effect on voice traffic in both schedulers because of the voice streams high priority and small packets size.

Video streams show some throughput improvement with the increase of the retry limit in both schedulers, because packets have more chances to be delivered as long as their delay bound is not exceeded. We observe that when the retry limit is six or more, the increase of the packet retry limit is not necessarily followed by throughput increase, because the channel conditions and delay bound play a substantial role in the packet transmission besides the packet retry limit. The SFS scheme succeeds in providing higher throughput than the standard scheduler, since the SFS scheme accommodates the burstness of VBR video traffic by dynamically computing the TXOPs.

Both schedulers show enhancements in FTP throughput as shown in Figure 9a. Because the FTP packets are delay tolerant, the increase of the retry limit allows both schedulers to accommodate more packets.

In general, the packet loss ratio of all traffic flows decreases with the increase of the retry limit as shown in Figure 9b. Voice traffic indicates almost no drop in both schedulers, which explains the steady throughput as depicted in Figure 9a.

When the retry limit is six or more, the SFS scheme manages to decrease the drop ratio down to 5% as a consequence of the adaptive computation of TXOPs coupled with the SF prioritization. The FTP drop ratio in the SFS scheme is higher than voice and video drop ratio. This is expected as a result of the SF mechanism. However, the SFS succeeds in maintaining low drop ratio for FTP streams because of the fairness measure incorporated by the SF.

The packet retry limit has a clear effect on the average delay as shown in Figure 9c. Voice traffic in the SFS experiences high average delay increase, because of the retry limit


*Wirel. Commun. Mob. Comput.* 2013; **13**:95–110 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

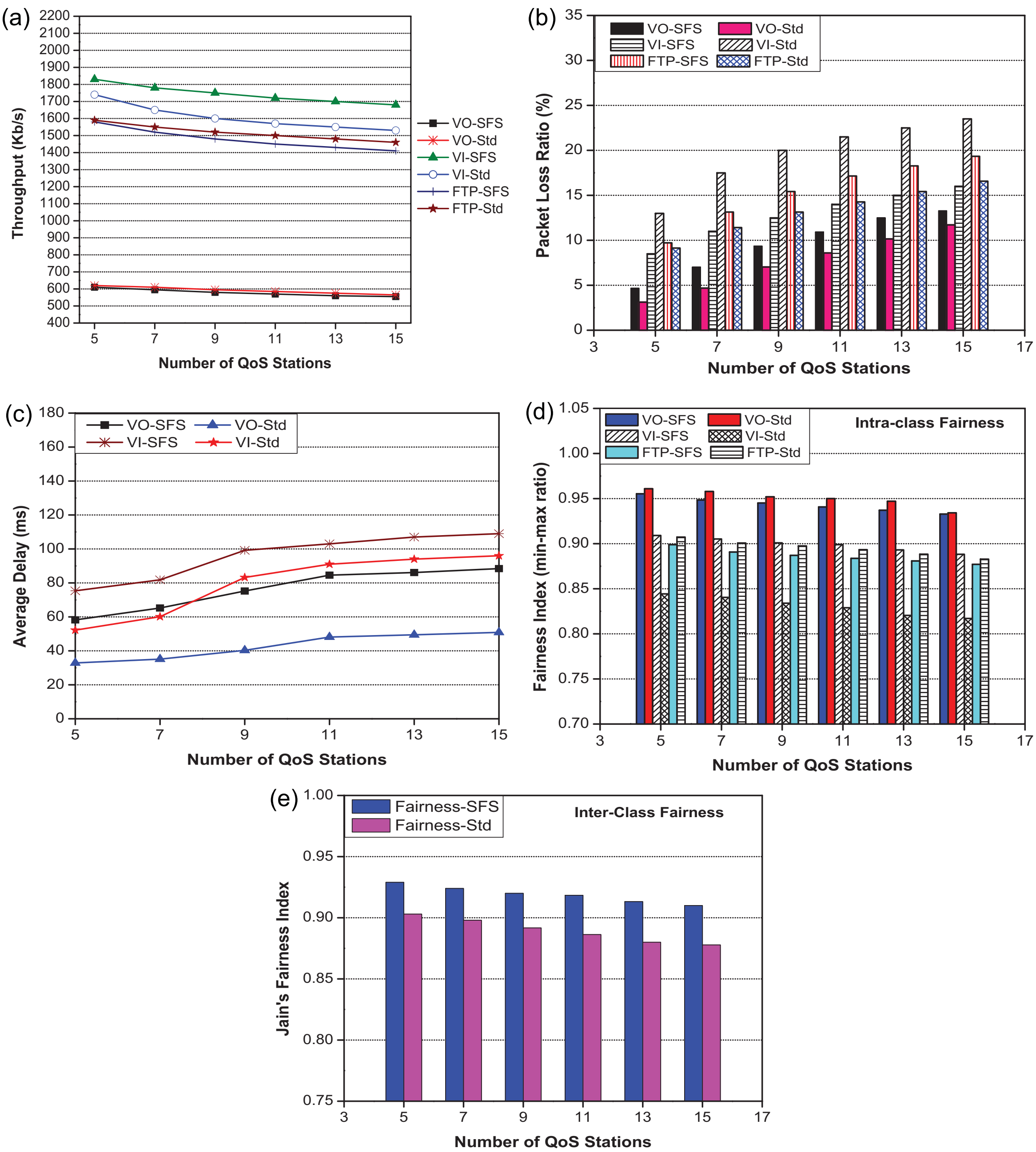



**Figure 8.** The effect of increasing the number of associated QoS stations while keeping the network load fixed. (a) Number of stations against the average throughput in kbps; (b) number of stations against the packets dropping ratio; (c) number of stations against the average delay in ms; (d) number of stations against the Min-Max fairness index; and (e) number of stations against the Jain's fairness index.

of packets remaining in the queue from previous TXOPs. The standard scheduler shows a lower increase in voice average delay because voice streams have high priority and small packet sizes.

Video average delay in both schedulers experiences high delay as shown in Figure 9c. We observe that when the retry limit is six or more, the video average delay increases sharply, which explains the effect of retry limit on the delivered packets as well as the channel conditions. In addition, the delay indicated by the SFS scheme is caused also by the mechanism of the SF. The increase in video traffic average delay shown by the standard scheduler is due to the

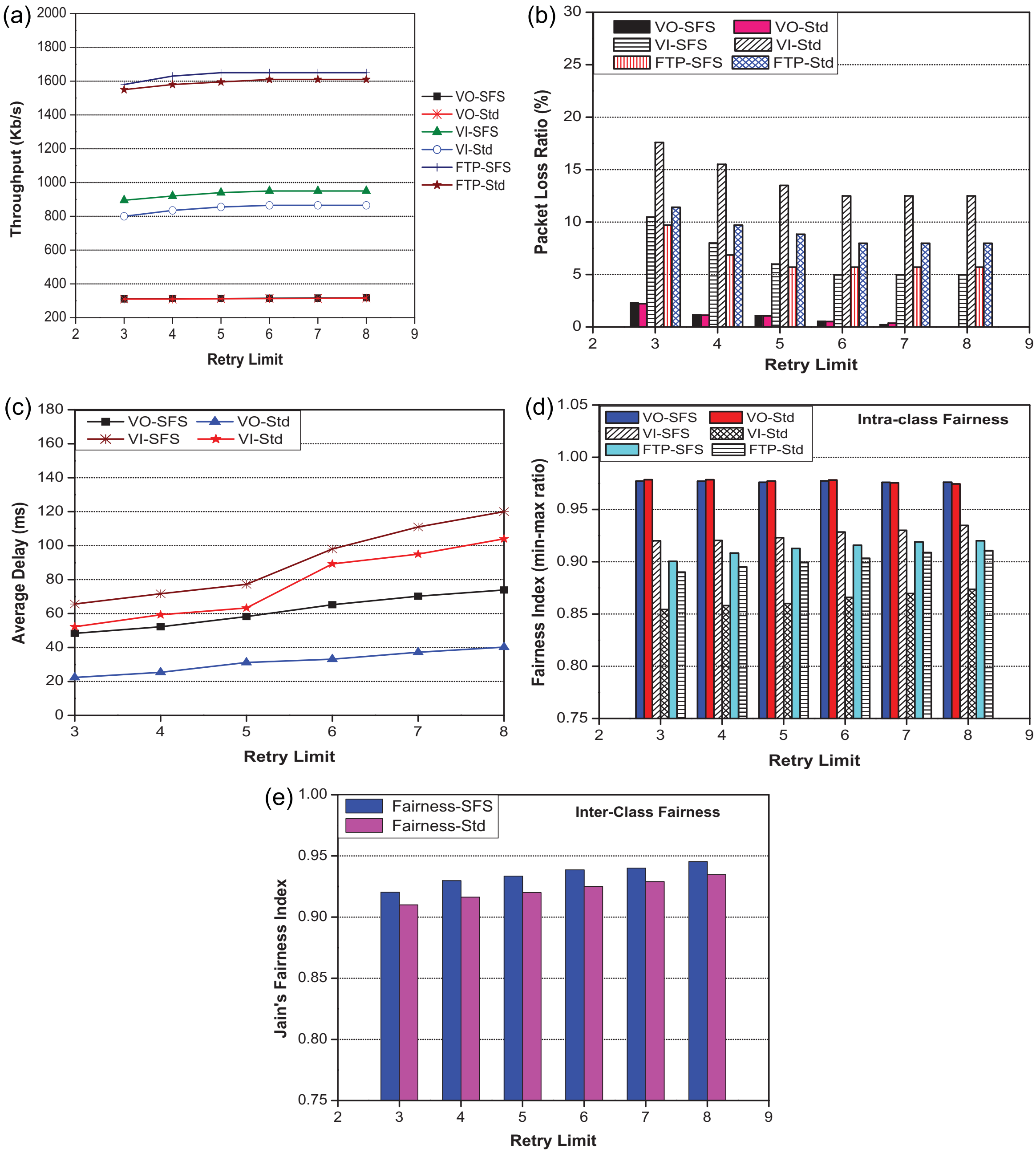



**Figure 9.** The effect of increasing the packet retry limit. (a) Packet retry limit against the average throughput in kbps; (b) packet retry limit against the packets dropping ratio; (c) packet retry limit against the average delay in ms; (d) packet retry limit against the Min-Max fairness index; and (e) packet retry limit against the Jain's fairness index.

VBR burstness, the lack of channel quality consideration and static computation of TXOPs.

The intra-class fairness index of voice traffic indicates high fairness in both schedulers as shown in Figure 9d, because both schedulers succeed in providing the required throughput for voice traffic. The SFS scheme indicates better intra-class fairness index for VBR video traffic. The SFS scheme shows higher FTP fairness index than the standard scheduler due to the better consideration of channel quality and the dynamic computation of the TXOPs.

The increase of the retry limit enhances the inter-class fairness as shown in Figure 9e, because the average throughput of video and FTP traffic flows is increased. However, the standard scheduler shows lower inter-class fairness index because of the lack of fairness among the different streams' types, specifically, video against voice streams.

## 5. CONCLUSIONS

We proposed a SFS for the IEEE 802.11e HCCA mode. The objective of the scheduler is to accommodate multiple TS per station with diverse QoS requirements. The SFS scheme is designed to adapt to the variations of the channel quality, diversity of QoS requirements, and provide fairness among the different TS. The scheme is evaluated implementing different scenarios through a comprehensive simulation. Results revealed that SFS succeeds in achieving higher performance in terms of throughput and packet dropping ratio compared to the standard scheduler, while maintaining fairness among different TS. As for the future work, we are in the process of the practical implementation of the proposed schedular into a WiFi test-bed. The scheduler will be integrated into an open-source Linux based operating system of 802.11g AP.

## 6. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers whose feedback helped improve the quality of this paper.

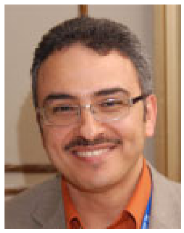
## REFERENCES

1. IEEE Part 11. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements, 2005.
2. Cowling J, Selvakennedy S. A detailed investigation of the ieee 802.11e hcf reference scheduler for vbr traffic. In *ICCCN 2004*, 2004.
3. Fiore M, Casetti C, Chiasserini C, Garetto M. Notes on the inefficiency of 802.11e hcca. In *2005 IEEE 62nd Vehicular Technology Conference, VTC-2005-Fall*, Vol. 4, 2005; 2513–2517.
4. Xiao. Y. Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e. *IEEE Transaction on Wireless Communications* 2005; **4**(4): 1506–1515.
5. Yang X, Frank L, Ming L, Jingyuan Z, BoL. L. Dynamic bandwidth partition schemes for integrated voice, video, and data traffic in the ieee 802.11e distributed wireless lans. *Wireless Communications and Mobile Computing* 2010; **23**(3): 391–412.
6. Çiftçioglu E, Gürbüz Ö. Scheduling for next generation WLANs: filling the gap between offered and observed data rates. *Wireless Communications and Mobile Computing* 2009; www.interscience.wiley.com. DOI:10.1002/wcm.808.
7. Sharon O, Altman E. An efficient polling mac for wireless lans. *IEEE/ACM Transactions on Networking* 2001; **9**(4): 439–451.
8. Hsieh R-R, Lee T-H, Kuo Y-W. Energy-efficient multi-polling scheme for wireless lans. *IEEE Transactions on Wireless Communications* 2009; **8**(3): 1532–1541.
9. Kim B-S, Kim SW, Fang Y. Wong TF, Two-step multipolling mac protocol for wireless lans. *Selected Areas in Communications, IEEE Journal* 2005; **23**(6): 1276–1286.
10. Ni Q, Ansel P, Turletti T. Fhcf: a simple and efficient scheduling scheme for ieee 802.11e wireless LAN. *Mobile Networks and Applications* 2006; **11**(3): 391–403.
11. Cicconetti C, Lenzini L, Mingozzi E, Stea. G. Design and performance analysis of the real-time HCCA scheduler for IEEE 802.11e WLANs. *Computer Networks* 2007; **51**(9): 2311–2325.
12. Mingozzi E, Cicconetti C, Lenzini L, Stea. G. An efficient cross layer scheduler for multimedia traffic in wireless local area networks with ieee 802.11e hcca. *ACM SIGMOBILE Mobile Computing and Communications Review* 2007; **11**(3): 31–46.
13. Foronda A, Higuchi Y, Ohta C, Yoshimoto M, Okada Y. Service interval optimization with delay bound guarantee for hcca in ieee 802.11e wlans. *IEEE Transactions on Communications* 2007; **11**: 3158–3169.
14. Fallah Y, Alnuweiri H. Hybrid polling and contention access scheduling in ieee 802.11e wlans. *Journal of Parallel and Distributed Computing* 2007; **67**(2): 242–256.
15. Wolf J, Heckmüller S, Wolfinger B. Dynamic resource reservation and QoS management in IEEE 802.11e networks, SPECTS 2005, Philadelphia, USA, pp. 149–161.
16. Skyrianoglou D, Passas N, Salkintzis AK. Arrow: an efficient traffic scheduling algorithm for ieee 802.11e hcca. *IEEE Transactions on Wireless Communications* 2006; **5**(12): 3558–3567.
17. Zhao L, Zhang J. Hub-polling-based ieee 802.11 pcf with integrated qos differentiation. *Wireless Communications and Mobile Computing* 2009; **9**(9): 1220–1230.
18. Boggia G, Camarda P, Grieco LA, Mascolo S. Feedback-based bandwidth allocation with call admission control for providing delay guarantees in ieee 802.11e networks. *Computer Communications* 2005; **28**(3): 325–337.
19. Fan WF, Gao D, Tsang DHK, Bensaou S. Admission control for variable bit rate traffic in ieee 802.11e wlans. In *Communications, 2004 and the 5th International Symposium on Multi-Dimensional Mobile Communications*

*Proceedings. The 2004 Joint Conference of the 10th Asia-Pacific Conference*, Vol. 1, August–1 September 2004; 272–277.
20. Boggia G, Camarda P, Grieco LA, Mascolo S. Feedback-based control for providing real-time services with the 802.11e MAC. *IEEE/ACM Transactions on Networking* 2007; **15**(2): 323–333.
21. Grilo A, Macedo M, Nunes M. A scheduling algorithm for QoS support in IEEE 802.11e networks. *IEEE Communication Magazine* 2003; **10**: 36–43.
22. Ramos N, Panigrahi D, Dey. S. Dynamic adaptation policies to improve quality of service of real-time multimedia applications in ieee 802.11e wlan networks. *Wireless Networking* 2007; **13**(4): 511–535.
23. Hantrakoon S, Phonphoem A. Priority based hcca for ieee 802.11e. In *International Conference on Communications and Mobile Computing (CMC), 2010*, Vol. 3, 2010; 485–489.
24. van der Schaar M, Andreopoulos Y. Zhiping H. Optimized scalable video streaming over ieee 802.11 a/e hcca wireless networks under delay constraints. *IEEE Transactions on Mobile Computing* 2006; **5**(6): 755–768.
25. Chen C-L, Lai C-H. Implementation and esl verification of new earliest-deadline-first scheduler for video streaming over wireless networks. In *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09*, 2009; 390–393.
26. Kim Y, Park H, Suk J. Bandwidth guarantee per session for video transmission over ieee 802.11e hcca. In *7th IEEE Consumer Communications and Networking Conference (CCNC), 2010*, 2010.
27. Noh Z, Suzuki T, Tasaka S. Application-level qos and qoe assessment of a cross-layer packet scheduling scheme for audio-video transmission over error-prone ieee 802.11e hcca wireless lans. *IEICE TRANSACTIONS on Communications* 2010; **E93-B**(6): 1384–1394.
28. Qaimkhani I, Hossain E. Contention-free approaches for wifi mac design for voip services: performance analysis and comparison. *Wireless Communications and Mobile Computing* 2009; **9**(8): 1089–1101.
29. Qaimkhani I, Hossain E. A novel qos-aware mac protocol for voice services over ieee 802.11-based wlans. *Wireless Communications and Mobile Computing* 2009; **9**(1): 71–84.
30. Jain R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley/Interscience, New York, NY, USA, 1991.
31. Dianati M, Shen X, Naik. S. A new fairness index for radio resource allocation in wireless networks. *IEEE Wireless Communications and Networking Conference* 2005; **2**: 712–717.

## AUTHORS' BIOGRAPHIES

**Najah Abu Ali** received her B.S. and M.S. degrees in Electrical Engineering from University of Jordan, Amman, Jordan and her Ph.D. degree in 2006 in Computer Networks in Electrical Engineering department at Queen's University, Kingston, Canada. She is an Associate Professor at the College of Information Technology, United Arab Emirates University (UAEU) with the Computer Networks Engineering track. She had a postdoctoral fellowship at the School of Computing, Queen's University. Her research interests comprise wired and wireless communication networks. Specifically, analytical and measurement-based network performance management and Quality of Service and resource management of single and multihop wireless networks. Abu Ali is an expert on Broadband Wireless Networks architecture, design, QoS provisioning and performance and has published extensively in the area.



**Hossam Hassanein** is with the School of Computing at Queen's University working in the areas of broadband, wireless and variable topology networks architecture, protocols, control and performance evaluation. Hassanein obtained his Ph.D. in Computing Science from the University of Alberta in 1990. He is the founder and director of the Telecommunication Research (TR) Lab (www.cs.queensu.ca/∼trl) in the School of Computing at Queen's. Hassanein has more than 350 publications in reputable journals, conferences and workshops in the areas of computer networks and performance evaluation. He has delivered several plenary talks and tutorials at key international venues, including Unconventional Computing 2007, IEEE ICC 2008, IEEE CCNC 2009, IEEE GCC 2009, IEEE GIIS 2009, ACM MSWIM 2009 and IEEE Globecom 2009. Hassanein has organized and served on the program committee of numerous international conferences and workshops. He also serves on the editorial board of a number of International Journals. He is a senior member of the IEEE, and is currently chair of the IEEE Communication Society Technical Committee on Ad hoc and Sensor Networks (TC AHSN). Hassanein is the recipient of Communications and Information Technology Ontario (CITO) Champions of Innovation Research award in 2003. He received several best paper awards, including at IEEE Wireless Communications and Network (2007), IEEE Global Communication Conference (2007), IEEE International Symposium on Computers and Communications (2009), IEEE Local Computer Networks Conference (2009) and ACM Wireless Communication and Mobile Computing (2010). Hassanein is an IEEE Communications Society Distinguished Lecturer.


DOI: 10.1002/wcm