Adaptive TXOP Allocation based on Channel Conditions and Traffic Requirements in IEEE 802.11e Networks

Abhinav Arora, Sung-Guk Yoon, Young-June Choi, and Saewoong Bahk

Abstract— **IEEE 802.11e has established a new access mechanism, *Hybrid Coordination Function (HCF)*, as a step towards provisioning QoS support. In this mechanism, a simple HCF scheduler is introduced to take QoS requirements of admitted flows into account and allocate *Transmission Opportunities (TXOP)*, the time under which a station can send its burst of data packets, to stations. Although many approaches have been taken in order to address the mechanism's inherent problems, unsubstantial effort has been afforded in tackling a channel-aware allocation mechanism. A station can be in either good or bad channel state in any feasible data rate, which can be modeled as a Markov chain of two states for each data rate. In view of this, we propose a mechanism of adaptive TXOP allocation applicable to existing scheduling algorithms. Our method works in accordance with channel and traffic conditions and complies with the link adaptation mechanism. Extensive simulation results verify that our method shows improved performance while ensuring long term fairness among stations and being adaptive to the channel conditions and underlying physical transmission rates.**

Index Terms— **IEEE 802.11e, transmission opportunities (TXOP), resource allocation, quality of service, link adaptation**

I. Introduction

Past few years have witnessed a phenomenal growth in wireless technologies. Among these technologies, IEEE 802.11 Wireless LAN (WLAN) has become a great success for data applications in hotspots owing to low cost, robustness and easy deployment. On the other hand, the number of multimedia applications have increased tremendously, which demand some quality of service (QoS) support such as guaranteed bandwidth, bounded delay and jitter. Providing such QoS support in 802.11 is challenging since the original 802.11 standard does not take QoS support into account.

In order to enhance the support of QoS, IEEE 802.11e [1] has developed a new protocol that uses differentiation mechanisms at the medium access control (MAC) layer. A number of studies have evaluated the current standard by analytical evaluation and simulations [2]. It uses a new medium access method called *Hybrid Coordination Function (HCF)* that combines a contention-based *enhanced DCF access mechanism (EDCA)* and a controlled *HCF channel access mechanism (HCCA)* in a single function. Recent performance evaluations of 802.11e HCF [3] show that HCF is more flexible than *Distributed Coordination Function (DCF)* and *Point Coordination Function (PCF)*.

Although 802.11e supports QoS demands to a certain extent, there are a number of challenges that must be addressed to enable comprehensive QoS support. HCCA is a very crucial mechanism in meeting QoS demands and thus designing a scheduler for HCCA has been an active objective of research. Apart from several other drawbacks [4], it is shown in [5] that the HCF scheduling algorithm is only efficient for flows with strict Constant Bit Rate (CBR) characteristics. There are some problems with schedulers that are identified in [4]. A major problem cited is the independence of scheduler operation from the channel condition. Varying channel conditions because of propagation loss, multipath effect, and interference can lead to packet drops and retransmissions, thereby increasing latency while degrading throughput. In terms of providing QoS differentiation, the channel condition is an important factor for consideration because it can potentially weaken the service differentiation. IEEE 802.11e provides a means of differentiation only when nodes experience the same channel conditions. Thus there is a need to focus on the impact of time varying network conditions in scheduling by HCCA.

Moreover, with the advent of technologies like adaptive modulation and coding (AMC) schemes, the usual two state Markov channel model [6] is invalid. Most of scheduling algorithms rely on this model for QoS provisioning which is no longer accurate with the introduction of AMC or link adaptation. When the link adaptation is used, a station (STA) that experiences a bad channel, i.e., low signal-to-noise ratio (SNR), may transmit and receive with a lower rate.

The key idea of our proposal is to exploit the channel conditions to increase system efficiency. Challenges while designing a scheduler for IEEE 802.11e networks are various QoS constraints imposed on the traffic flow and TXOP allocation by the standard draft. In this paper, we follow an approach different from other opportunistic schedulers (e.g., [7]) that run per-packet basis. We develop an adaptive TXOP allocation method applicable to the existing schedulers like

This research was supported by the MKE/IITA, Korea, under the ITRC support program (grant no. IITA- 2009-C1090-0902-0006) and the IT R&D program (grant no. 2008-F-007- 02)

Abhinav Arora was with the Department of EE and INMC, Seoul National University, Seoul, Korea, 151-742, when he was conducting this research (abhinav@netlab.snu.ac.kr). He is now with the Samsung Electronics, Suwon, Korea. Sung-Guk Yoon and Saewoong Bahk are with the Department of EE and INMC, Seoul National University ({sgyoon, sbahk}@netlab.snu.ac.kr). Young-June Choi is with the School of Information & Computer Engineering, Ajou University, Woncheon-dong, Youngtong-gu, Suwon, Gyunggi-do, Korea, 443-749. Correspondence to Young-June Choi, Phone: 82-31-219-2634, Fax: 82-31-219-1621, email: yjchoi@ieee.org.

standard scheduler [8] or Grilo scheduler [9]. Our TXOP allocation algorithm adjusts the length of TXOP adaptively and assigns the residual resource to other STAs according to their channel conditions.

We design a TXOP allocation policy which will not only try to increase system performance but also ensure long term fairness among STAs. A foremost feature of HCCA mechanism is its weighted temporal fairness and hence this feature needs to be preserved in the long-term at least. Our proposal allocates the minimal length of TXOP to the STA suffering from bad channel condition and lending its TXOP to the STA with a better channel condition. Following the conventional method to achieve fairness [6], we maintain a $lead/lag$ counter for each STA which specifies the amount by which STA is lagging behind or leading compared to its normal service amount. Based upon the $lead/lag$ value, each STA is made to give up or receive extra TXOP. We also ensure that a STA does not jeopardize QoS of any STA while giving up its TXOP to any of its peer. For this we explore the $QueueSize$ field of IEEE 802.11e header to adapt TXOP to actual traffic condition.

Mathematical analysis justifies the claim that our scheme ensures long term fairness among STAs hence preserving the property of HCCA. Extensive simulation results have been discussed to prove the efficiency of the proposed scheme over corresponding conventional implementations.

The rest of this paper is organized as follows. Section II provides the overview of the limitations and the summary of new mechanisms for QoS support defined in 802.11e. Section III illustrates the existing scheduling algorithms that have been proposed for 802.11e. In Section IV, we develop our adaptive TXOP algorithm. Section V presents the simulation results of our algorithm, and Section VI concludes.

II. IEEE 802.11e: Challenges and Approaches

A. IEEE 802.11e background

This section will briefly summarize the IEEE 802.11e standard draft [1] and its limitation in providing QoS support. The IEEE 802.11e working group has proposed a new MAC mechanism for supporting QoS called HCF which consists of EDCA and HCCA.

1) EDCA: EDCA supports priority differentiation among STAs and flows by using different back-off parameters. It introduces *Traffic Categories (TCs)* and gives different priorities to different TCs. EDCA has two priority schemes: one is *Interframe Space (IFS)* priority scheme, and the other is contention window (CW) priority scheme. A STA can send a data packet or start to decrease its backoff counter after it detects the channel being idle for some IFS. The 802.11e introduces a new interval, *Arbitration Interframe Space (AIFS)*, in addition to the existing two IFSs, DIFS (DCF IFS) and PIFS (PCF IFS). The AIFS can be adjusted for each TC according to the corresponding priority. CW priority scheme implements service differentiation by using different CWs for different TCs, which gives different backoff numbers to different priority classes. The backoff value is set to a counter, which is a random number from the interval [1, CW+1], where CW is initially set to a minimum value (CWmin) and increased whenever the node involves in a collision up to a maximum value known as CWmax.

The new CW for EDCF is defined as [10]

$$newCW[AC] = ((oldCW[AC] + 1) * 2) - 1. \qquad (1)$$

2) HCCA: HCCA provides a centralized polling scheme to allocate guaranteed channel access to traffic flows based on their QoS requirements. In HCF, the superframe is divided into the Contention Free Period (CFP) that starts with every beacon, and the Contention Period (CP). During the CP, access is governed by EDCA, though the Hybrid Coordinator (HC) can initiate controlled access periods (CAPs) at any time.

A CAP is formed by a sequence of *transmission opportunities (TXOPs)*. A TXOP is a period of time in which a STA or the HC can transmit a burst of data frames separated by a short interframe space (SIFS) interval. The HC starts an uplink TXOP by issuing a poll request to a STA. A TXOP ends when at least one of the following conditions is met:

- Transmission of a data frame with the nonfinal flag is set to 0.
- The TXOP duration, given by the variable dot11DefaultCPTXOPLimit, is expired.
- A polled STA allowed the wireless medium to remain idle for a PIFS interval in an uplink TXOP.

The rate and proportion of contention-free bursts are given by the variables dot11CAPRate and dot11CAPMax, respectively.

B. System model

This paper focuses on adapting TXOP under HCCA polling mechanism with respect to varying channel conditions. We do not consider direct transmission between two stations as allowed in the IEEE 802.11e standard. Moreover, with the schemes like AMC, it is possible for a station to be at different PHY rates depending on the channel conditions. For example, IEEE 802.11a/g and IEEE 802.11b support eight and four data rates, respectively. However, unlike cellular networks, IEEE 802.11 system doesn't support any explicit feedback channel to adopt the transmission rate according to the varying channel conditions. In this model, we adopt an *Auto Rate Fallback (ARF)* Algorithm as a link adaptation mechanism to adjust the transmission rates according to the channel conditions. In the ARF algorithm, when no acknowledgment is received after two consecutive data frame transmission attempts, the sender decreases the bit rate. If ten consecutive data transmissions are successful, the sender tries to increase the bit rate. Therefore, it may take some time to update the channel condition in the IEEE 802.11 network.

III. Scheduling in IEEE 802.11e Networks

A. QoS parameters in IEEE 802.11e

An important component of HCCA framework is the scheduling policy implemented to generate TXOPs for each STA. The policy runs at QoS access point (QAP) for each flow with its traffic specifications (TSPEC). A separate reservation must be made for each traffic stream (TS), which is a unidirectional stream of MAC service data units (MSDUs) requiring QoS guarantees. A minimum set of TSPEC parameters is specified in [1] which in turn helps scheduler determine a schedule for the stream to be admitted. Some of the parameters are briefly explained as below:

- Mean data rate (ρ): average bit rate for transfer of the packets in unit of bits per second.
- Nominal MSDU size (L): nominal size of a packet in octets.
- Minimum PHY rate (R): physical bit rate for transmit time and admission control calculations in bits per second.
- Surplus Bandwidth Allowance (SBA): specifies the excess allocation of time over and above the stated application rates required to transport a MSDU belonging to this stream; this field takes into account retransmissions.
- Delay bound (D): maximum delay allowed to transport a packet across the wireless interface (including queuing delay), in milliseconds.

  A STA can provide more parameters to QAP for better scheduling decisions in its TSPEC. Some of them are:
- Maximum MSDU size (M): maximum size of a packet in octets.
- Maximum Burst Size (MBS): maximum size of the data burst that can be transmitted at the peak data rate in octets.

The polling mechanism is supervised by the HC. Although its decision is most based on individual TS characteristics, the STA is then responsible for allocating the allotted time to each TS individually. As described in [8], HC uses the following parameters, which can be derived from the individual TSPECs of a STA j with n individual TSs.

- Minimum TXOP duration (mTD): the minimum TXOP duration that can be allocated to a STA and equals the maximum packet transmission time for any active TSPECs.

$$mTD_j = \max(\frac{M_i}{R}), \quad i = 1, \cdots, n. \tag{2}$$

- Maximum TXOP duration (MTD): the maximum TXOP duration that can be allocated to a STA. It should be less than or equal to the transmission time of aggregate maximum burst size (AMBS) of a STA. The AMBS is the sum of the maximum burst size of all TSPECs of a STA, i.e.,

$$AMBS_j = \sum_{i=1}^{n} MBS_i \tag{3}$$

  and

$$MTD_j \leq AMBS_j/R. \tag{4}$$

- Minimum Service Interval (mSI): the minimum time gap required between the start of two successive TXOPs allocated to a STA (in μs). Given that the average interval between the generation of two successive MSDUs for a TS is L/ρ, the mSI is calculated as the minimum of these intervals for all TSs:

$$mSI_j = \min(L_i/\rho), \quad i = 1, \cdots, n. \tag{5}$$

TABLE I

LIST OF SYMBOLS

Symbols	Definition
ρ	mean data rate
L	nominal MSDU size
R	minimum PHY rate
D	delay bound
M	maximum MSDU size
mTD	minimum TXOP duration
MTD	maximum TXOP duration
SI	service interval
mSI	minimum service interval
MSI	maximum service interval

- Maximum Service Interval (MSI): the maximum time interval allowed between the start of two successive TXOPs allocated to a STA (in μs). A reasonable assumption is that MSI should be related to the lowest delay bound D of all STA's TSPECs. Hence we have

$$MSI_j \leq \min[D_i - MTD_i], \quad i = 1, \cdots, n. \tag{6}$$

Symbols are also listed in Table I. We next discuss some schedulers which will use these parameters in deriving system parameters.

B. Reference scheduler

Reference scheduler is presented in [8], which will be referred to as *reference scheduler*. This scheduler uses only the mandatory TSPEC parameters in calculating two additional scheduling parameters:

- Service Interval (SI): the time between the start of successive TXOPs allocated to a QoS STA, which is the same for all the STAs. To calculate this first, QAP determines the minimum value of all MSIs for all TSs and QSTAs. Then the scheduler chooses SI as the highest submultiple value of the 802.11e beacon interval.
- TXOP Duration ($TXOP$): the time needed to transmit all the packets that arrive during an SI in a TS queue at the minimum rate R. If N_i be the number of packets of mean length L_i that arrive in SI with the mean rate ρ for TS i, we have

$$N_i = \lceil \frac{SI * \rho}{L_i} \rceil \quad i = 1, \cdots, n. \tag{7}$$

  Therefore the TXOP duration for each TS_i, denoted by $TXOP_i$, can be calculated as

$$TXOP_i = N_i \cdot \left(\frac{L_i}{R} + 2 \cdot SIFS + ACK \right), \; i = 1, \cdots, n. \tag{8}$$

Note that the reference scheduler in [8] is updated to consider power-savings in the final standard, which is not considered in this work for comparison with our scheduler.

The admission control unit (ACU) is trivial in the reference scheduler. ACU admits a stream if it satisfies the following inequality:

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=1}^{k} \frac{TXOP_i}{SI} \leq \frac{\text{dot11CAPRate}}{64[\mu s]}, \tag{9}$$

where k is the number of existing streams and $k+1$ is used as the index for the newly arriving stream and dot11CAPRate variable specifies the fraction of time that can be used for contention-free bursts, expressed in unit of microseconds per $64\mu s$.

From a simple observation it can be noted that this scheme allocates fixed length TXOPs for each STA at constant interval. This simple HCF scheduling can be efficient if the traffic is strictly CBR. However, when real-time applications generate VBR traffic, it may cause the average queue length to increase, possibly resulting in packet drop.

C. Grilo scheduler

Many variants of the reference scheduler have been proposed to overcome the shortcomings of the simple scheduler. The scheduling algorithm proposed by Grilo *et al*. [9] follows a very novel approach and has been quite popular.

Grilo Scheduler extends the functionality of HC by allowing it to:

1) allocate TXOPs of variable length (instead of fixed TXOPs), and
2) poll each STA at variable and different service intervals (instead of polling all STAs with period SI).

It provides flexibility in allocating TXOPs while maintaining the average duration of $TXOP_i$ equal to TD_i by implementing a TXOP timer (or equivalently a token bucket mechanism). The TXOP timer of STA i increases at a constant rate equal to TD_i/mSI_i, which corresponds to the total fraction of time the STA can spend in polled TXOPs.

If a STA i is polled at time t, the next poll should be issued at time t' that satisfies the relation:

$$t + mSI_i \leq t' \leq t + MSI_i. \tag{10}$$

The HC scheduler has to decide which STA to poll first, among those that satisfy (10) at a given moment. Grilo scheme follows Delay-Earliest-Due-Date algorithm (EDD) in selecting the STA which satisfies the above equation. The deadline for the start of a TXOP here is $(t + MSI)$.

D. Other approaches

There has been other related work which tries to address the issues of traffic variability and varying application requirements. In [5], *flexible HCF (FHCF)* is proposed to adapt to the traffic variability by adjusting the TXOP of each flow using queue length estimation. The AP uses the queue length information to calculate the current demands of the flows. If the current demand is more than the reserved time, the TXOP is increased and vice versa.

In [11], the same problem is countered using another approach. Rather than adjusting the TXOP, the adaptation proposes additional polling to avoid delay for nodes scheduled later in the polling list. The decision to perform additional polling is made by the AP and is based on queue length information received from the nodes and the timestamp of when the node was given extra time in the HCCA. Fairness is ensured by incorporating the weight of each flow, which factor was adapted in the last time.

In [12], the authors extend the Grilo scheduler by exploiting *Queue Size* and *TXOP Duration* fields. QAP capability is enhanced to allocate TXOP in a more intelligent manner. In [13], the authors try to adapt TXOP to provide throughput fairness among STAs with different physical rates. To summarize, as we can see most of the related research work has mainly focused on adapting TXOP with respect to traffic variability and varying application requirements. No substantial effort has been observed which tries to adapt the scheduling according to channel conditions of the STAs. This motivated us to design a channel adaptive scheduler that satisfies each STA's QoS requirements and complies with the standard.

IV. Adaptive TXOP Allocation Scheme

A. Motivation

Most of the scheduling algorithm relies on the discrete-time Markov chain with two states for QoS provisioning. The channel moves between the two states - error-free (good), and error-prone (bad) [6] - according to a certain transition probability matrix.

This simple theoretical model does not take into account that the *bit error rate (BER)* can be lowered, for the same SNR, at the expense of physical bit rate. Reduction of physical bit rate can be achieved by selecting modulation and coding schemes such as AMC or link-adaptation [9], [15]. This scheme, consequently, invalidates the simple Markov model and demands a model that consider a good and bad channel state for each possible PHY rates.

Hence, it is imperative for a scheduler, for better system throughput and performance, to be consistent with the above mentioned improved model and be responsive to the underlying channel conditions. However, as discussed earlier, all of the scheduling algorithms for IEEE 802.11e networks do not exploit channel conditions while allocating resources like TXOPs. Although Grilo Scheduler studies the impact of link adaptation, it allocates TXOP irrespective of the physical conditions. Another observation to be made is that while allocating TXOPs, the scheduler always assumes the minimum physical rate rather than the actual one. Thus, TXOP allocation is not efficient with respect to channel conditions and underlying physical rate.

Our approach addresses these issues by adapting our system model to take into account Link Adaptation and basing our scheduling algorithm on the idea of any opportunistic scheduler-exploiting time-varying channel condition to maximize system throughput. However, unlike cellular networks, IEEE 802.11 systems do not support any feedback channel, therefore, a link adaptation algorithm - ARF algorithm - is used to adapt the transmission rate to the channel conditions. Under this algorithm, any station suffering from bad channel will decrease its physical rates till it has the threshold SNR. During this duration, scheduler policy will allocate just enough TXOP to the STA to send only one packet and the rest of the TXOP will be allocated to STAs with good channel conditions thereby improving the overall system performance in terms of throughput.

However, a scheme designed only to maximize the overall throughput could be very biased in terms of fairness among

TABLE II

PARAMETERS USED IN THE ALGORITHM

Parameter	*Definition*
c_i	Amount of additional service received by a lagging STA
f_i	Amount of additional service received by a non-lagging STA

STAs. So we design the TXOP allocation policy that will not only try to increase system performance but also ensure long term fairness among STAs. Therefore, scheduler ensures that when the STA gets to good channel condition, its lost service will be returned by graceful degradation of other STAs. To give back TXOP without jeopardizing its own QoS, we exploit its traffic requirements to return back the excess service whenever it can.

Scheduler aims to provide the temporal fairness rather than throughput fairness among stations for it allows the system to provide desirable "performance isolation" and to avoid "performance anomaly". The main components of the proposed temporal fair scheduler are listed below. These components are described in more detail in the following section.

- An error-free service model that describes how the algorithm provides service to sessions with error-free channels.
- A lead/lag counter for each session that indicates whether the session is leading or lagging its error-free service model and by how much.
- A compensation model that makes a lagging session compensated at the expense of leading sessions when its link becomes error-free again.
- A means for monitoring and predicting the channel state for every backlogged session.

B. Algorithm formulation

Now we discuss how the scheduler manages the lost and excess service. In order to account for the service lost or gained by a STA due to errors, we associate each system S with a reference error-free system S^r. Then, a session is classified as leading, lagging, or satisfied with respect to S^r. Several new parameters are introduced and their definitions are given in Table II.

Our important observations that can be made at this point are:

- The scheduling decisions by S^r are made by an existing scheduler like Grilo. Our scheduler allocates TXOP to the STA scheduled by the existing scheduler.
- To ensure fairness, we introduce the term lag_i which is the difference between the service that session i should receive in S^r and the service it has received in S. Thus, to achieve perfect fairness, the algorithm (work conserving) always maintains

$$\sum lag_i = 0. \tag{11}$$

- In the IEEE 802.11e standard, a STA is required to have pre-defined limit up to the maximum TXOP that can be allocated. So it cannot have infinite service.

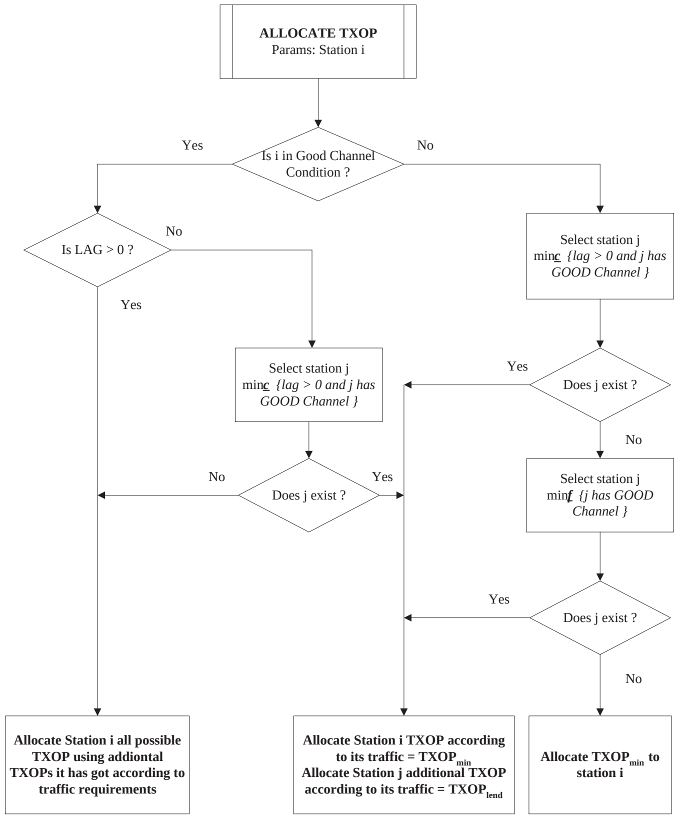


Fig. 1. Main Algorithm.

The main algorithm is depicted in Fig. 1. To provide short term fairness, we distinguish two types of additional service in the algorithm: *excess service* and *compensation service*. Excess service is made available when a session receives more service than required, while compensation service is made available due to a leading session giving up its lead.

First of all, lagging sessions have higher priority to receive additional services to expedite their compensation. By this way a lagging session is guaranteed to catch up, no matter what the lags of the other sessions are, and the short term fairness property is ensured among lagging sessions during compensation. This policy is implemented by keeping a new virtual time c_i that keeps track of the amount of additional services received by session i while it is lagging.

When additional service is available, the lagging session j with the minimum c_j that can send is chosen to receive it. Session j's c_j is then updated accordingly. However, if such session j does not exist and there are active sessions that can send left in the system, then this excess service is distributed among all non-lagging sending sessions. This policy is implemented by keeping a virtual time f_i that keeps track of the normalized amount of excess services received by session i while it is non-lagging. To distribute the excess service, the non-lagging session j with the minimum f_j that can send is chosen to receive it.

Let's take a very simple example to walk through the algorithm. Consider three stations i, j and k. Assume that station i is in the bad channel condition for time (t_1, t_2) and be in the good condition then after. Without loss of the generality, assume that station j and k are in good channel condition as ever and are always hungry for more bandwidth or more TXOP. For example they are serving a file transfer application.

By definition of the algorithm, station i will be given $TXOP_{\min}$ for the time duration (t_1, t_2). Therefore, quite clearly lag_i can be calculated as $TXOP_{normal}$ - $TXOP_{\min}$ * (number of times station i was chosen by the scheduler) where $TXOP_{normal}$ is the $TXOP$ that station i would have got in normal channel conditions.

Whenever station j or station k are selected by the scheduler in the time (t_1, t_2), they will be entitled to have that extra $TXOP$ (according to their f values) which is available to the scheduler. This clearly will ensure higher throughput for this duration in comparison to the case where station i was allowed to have the whole $TXOP$ that was initially allocated based upon its traffic requirements.

This scheme ensures temporal fairness as at t_2, lag_i will be positive and lag_j and lag_k will be negative. With this algorithm station i will reclaim its $TXOP$ (as and till lag_i is < 0) as other station will shed off the extra $TXOP$ available as $TXOPs$ are calculated due to the traffic requirement of stations j and k and their actual physical rates. Owing to the fact they are given $TXOP$ based on their actual physical rate rather than minimum physical rate, which allows scheduler to give the rest to station i to make up for the lost $TXOP$. We can summarize the behavior of the different stations over the course of time as:

Station i will be getting $TXOP_{\min}$ for the duration (t_1, t_2) and $TXOP_{normal}$ with extra $\Delta TXOP$ then after till its lag equals zero. For station j and k they would have $TXOP_{normal}$ with extra $\Delta TXOP$ for duration (t_1, t_2) and $TXOP_{normal}$ then after. In summary, the algorithm ensures in (a) long term fairness and (b) enhanced throughput.

The algorithm moreover also ensures graceful degradation by not disturbing the normal $TXOP$ of the stations in good channel conditions by always allocating enough $TXOP$ according to the traffic requirements and the actual physical conditions rather than the minimum physical rate. Example taken above only shows cases when there is one lagging station but it can be generalized to multiple lagging non-lagging stations by choosing the station for extra $TXOP$ using the parameters c, f and lag as explained in detail in the formulation of the algorithm.

We prove that our algorithm insures long term temporal fairness, starting with two lemmas giving bounds for the difference between the virtual compensation times (c_i's) and the virtual excess times (f_i's) of any two active STAs.

Lemma 1: The difference between the virtual compensation times of any two error-free STAs that are both lagging is bounded as follows:

$$| c_i - c_j | < TXOP_{\max}. \tag{12}$$

The proof is given in Appendix A.

Lemma 2: The difference between the virtual excess times of any two error-free STAs i and j that are both non-lagging is bounded as follows:

$$| f_i - f_j | < TXOP_{\max}. \tag{13}$$

The proof follows as above.

TABLE III
IEEE 802.11A MAC PARAMETERS

PHYSICAL PARAMETERS	VALUE
aSlotTime	$9\mu s$
Beacon Interval	100ms
aFragmentationThreshold	1024 octets
aRTSThreshold	500 octets
SIFS	$16\mu s$
PIFS	$25\mu s$
DIFS	$34\mu s$
aShortRetryLimit	7
aLongRetryLimit	7
dot11DefaultCPTXOPLimit	$3000\mu s$
dot11CAPRate	$21\mu s$
dot11CAPMax	$8000\mu s$
CAP timer update time	$5120\mu s$

Theorem 1: STA i has become error-free after time t_1, hence lagging is guaranteed to catch up after at most Φ units of time where

$$\Phi \leq \frac{1}{\beta} \frac{N \cdot (lag_i(t_1) + 2 \cdot TXOP_{\max})}{TXOP_{\max} - TXOP_{\min}}, \tag{14}$$

where N is the number of STAs, $lag_i(t_1)$ is the lag at the moment that STA i became error-free, and $\frac{1}{\beta}$ is the average rate at which STA can lend TXOP.
The proof is given in Appendix B.

C. Algorithm complexity

In the proposed algorithm, there are mainly four operations involved: (1) a session becoming active, (2) a session being selected to receive service, (3) an active session entering error mode, and (4) an active session becoming error-free. It is easy to deduce from the main algorithm in Fig. 1 that these operations eventually reduce to the following basic set operations: adding, deleting, and querying the element with the minimum key from the set. All of these operations are efficiently implemented in $O(\log N)$, where N represents the number of STAs in the network, by using a binary tree data structure, which maintains the tree based on f_i and c_i, respectively. More precisely, one tree will maintain all non-lagging error-free STAs based on f_i, and the other one will maintain all lagging error-free sessions based on c_i. Since all the four operations involve only a constant number of operations, they can be implemented in $O(\log N)$.

V. SIMULATION

To characterize the behavior of our adaptive TXOP allocation scheme, we conducted extensive simulation experiments and compared our algorithm with Grilo and reference scheduling. We implemented the proposed scheduler by using *ns-2 (Network Simulator 2)* [16] with the implementation code of the IEEE 802.11e/D12.0 MAC layer [1], [14]. We designed our simulation model to have only one TS per STA. In our experiments, the destination of all the flows is the QAP for fair comparison. We adopt the IEEE 802.11a PHY layer for the simulations and Table III summarizes the MAC parameters.

We measured throughput when there are three types of traffic: VoIP, CBR video, and best-effort (BE) traffic. In this

TABLE IV
TSPECS PARAMETERS

TSPEC	VoIP	CBR Video(MPEG-4)
Mean Data Rate	64 kb/s	320 kb/s
Delay bound	40 ms	120 ms
Nominal MSDU Size	160 octets	800 octets
Maximum MSDU Size	160 octets	800 octets
Peak Data Rate	64 kb/s	320 kb/s
Minimum PHY Rate	24 Mb/s	24 Mb/s

paper, we only plot the throughput of BE traffic, because the performance of VoIP and video traffic is affected not only by our TXOP algorithm but also by their own characteristics. This means that the performance of BE traffic explains the behavior of our algorithm well. We change the number of BE flows, when there are three flows of VoIP traffic and two flows of CBR video traffic. Audio traffic has been modeled using on-off sources with parameters corresponding to a typical phone conversation [17]. CBR traffic is modeled by MPEG-4, and Table IV summarizes the TSPEC reservation for VoIP and CBR Video. The BE traffic reservation only guarantees minimum throughput which is 1Mbps. VoIP traffic has the highest priority and BE traffic has the lowest priority.

To observe how lagging and non-lagging STAs exchange TXOPs according to our algorithm, three flows, each in the three traffic, enter bad channel state with different intervals. An audio flow and a CBR video flow stay in bad channel state between 20 and 30 sec, and between 30 and 40 sec, respectively. A BE flow then goes into bad channel state between 40 and 50 sec.

First, we consider a simple channel error model of uniform errors; the error rate is 0.7 during the bad state. Although this channel error model may not be realistic, it facilitates observation of throughput in terms of TXOP exchange. After presenting performance in the uniform error model, we will show performance in a Rayleigh channel model.

In the uniform error model, the mean throughput of BE traffic is shown in Figs. 2, 3, and 4, each for one, two, and ten BE flows. In Fig. 2, the BE flow achieves higher throughput in our adaptive algorithm, compared to the pure Grilo or reference scheduler. The BE flow obtains some TXOPs from the VoIP flow between 20 and 30 sec when the VoIP flow is in bad channel state. Then the BE flow obtains more TXOPs from the video flow between 30 and 40 sec when the video flow is in bad channel state. Throughput increase is higher between 30 and 40 sec, because the video flow lends more TXOPs to the BE flow than the VoIP flow does. Between 40 and 50 sec, the BE flow experiences channel errors, so its throughput decreases, but this is compensated after 50 sec.

When there are two BE flows and one of these stays in bad channel state, the throughput performance of the two flows is shown in Fig. 3. The performance of the reference scheduler is only exhibited, because the Grilo scheduler shows a similar tendency. The overall curve of the BE flow that suffers channel errors in Fig. 3 (a) is similar to Fig. 2 (b). However, throughput increase between 20 and 40 sec is less than that in Fig. 2 (b), because the two BE flows share TXOPs which are borrowed

TABLE V
PERFORMANCE IMPROVEMENT BY THE PROPOSED ALGORITHM

Velocity	Grilo	Reference
0.1 km/h	1.12%	9.94%
3 km/h	13.13%	7.43%
100 km/h	-0.23%	-12.95%
	max: 15–18%	

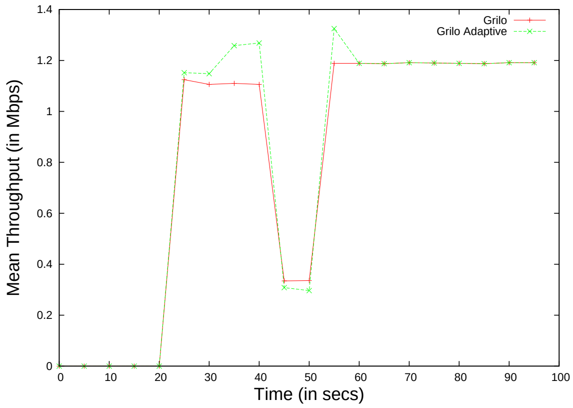
from other traffic. The BE flow also gives and takes TXOPs from the other BE flow that is always in good state. Hence, TXOPs are compensated more quickly between 60 and 70 sec compared to Fig. 2 (b). Also as shown in Fig. 3 (b), the BE flow that is always in good state borrows TXOPs from the other BE between 40 and 50 sec, accordingly achieving better throughput. Then, it receives less TXOPs after 50 sec while achieving more than its minimum throughput (1Mbps).

When there are ten BE flows and one of them suffers from channel errors, the throughput performance of the BE is plotted in Fig. 4. Ten flows share TXOPs between 20 and 40 sec, so the average throughput increases a little, and the number of additional TXOPs is quickly reduced after 50 sec.

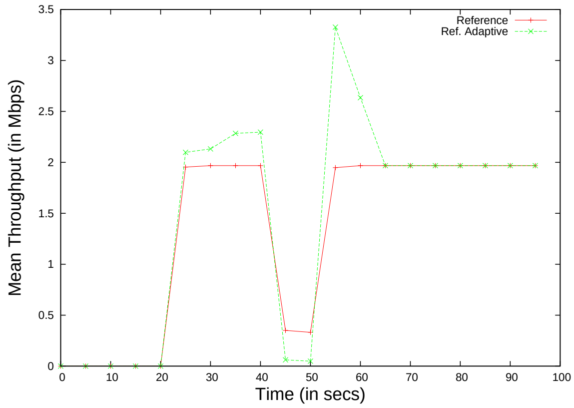
The mean throughput of VoIP traffic is plotted in Fig. 5. Unlike BE traffic, VoIP traffic switches between ON and OFF modes, resulting in fluctuation of the mean throughput. Between 20 and 30 sec, the VoIP traffic suffers from the bad channel. As mentioned earlier, the performance improvement for real-time flows in our algorithm has not been clearly observed, because their performance is affected by the scheduling algorithm as well as the service requirement such as delay bound. Overall the average performance of our algorithm for VoIP and CBR video traffic is similar to that of the algorithm without adaptive TXOP allocation. The adaptive algorithms improve 1% up to 7% on the average in Fig. 5.

Next, we present the performance under a Rayleigh channel model. We implemented the Jakes model [18] to emulate Rayleigh fading. When a data rate is decreased by the ARF algorithm, the channel state enters a bad state. A good state is recovered after successful transmissions. In the ARF algorithm, there are two cases to decrease the date rate: two consecutive transmission failure and the probation packet transmission failure. We define the bad state only for the consecutive failure. When there are ten BE flows, we plot the throughput of a BE flow in Figs. 6, 7, and 8, each with the velocity of 0.1 km/h, 3 km/h, and 100 km/h, respectively. Each curve without applying our adaptive algorithm follows the varying channel state. The case of velocity 0.1 km/h shows the variation of channel state very well, while it seems that the channel does not vary between the velocity 3 km/h and 100 km/h. This is because as the velocity increases, the channel fluctuates in a shorter time scale, but the throughput is averaged out more easily.

As shown in Fig. 6, the adaptive algorithm adjusts very well to the channel fluctuation when the velocity is 0.1 km/h. For both schedulers, the average throughput of our adaptive algorithm is higher (or lower) than that of the legacy algorithm when a rate is high (or low), by enabling to lend TXOPs when the channel becomes worse, and get more TXOPs when the channel becomes better. The overall performance

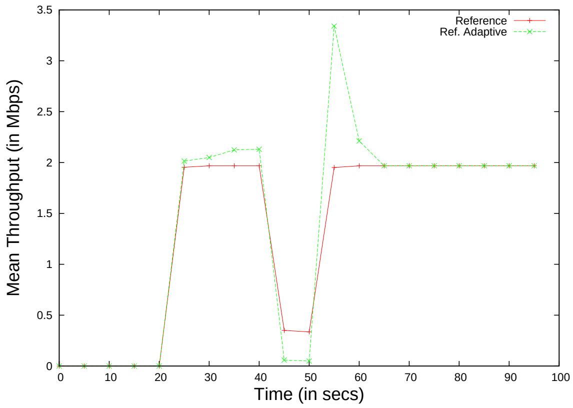


(a) Grilo scheduler

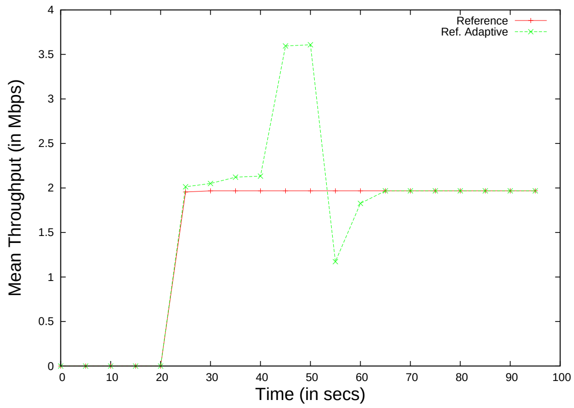


(b) Reference scheduler

Fig. 2. Performance comparison in the uniform error model, when there is one best-effort flow.

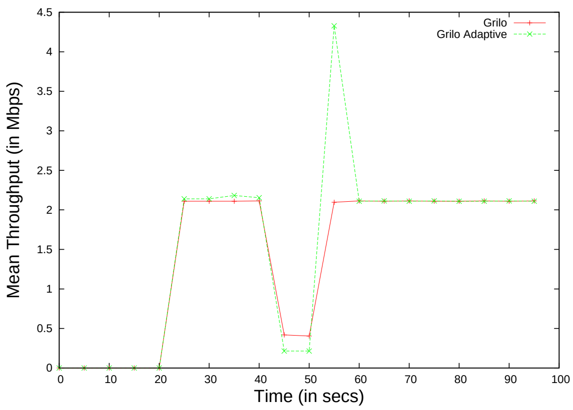


(a) The BE flow staying in bad state between 40 and 50 sec

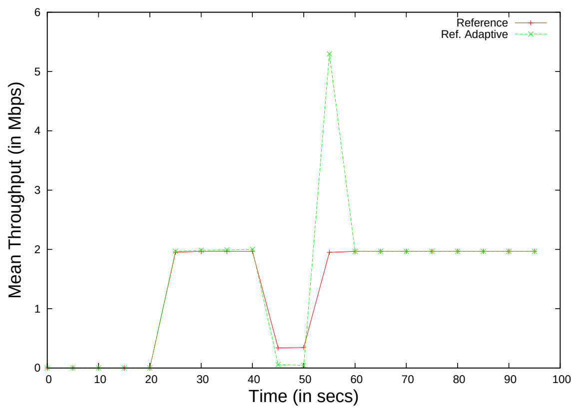


(b) The BE flow staying always in good state

Fig. 3. Performance comparison in the uniform error model, when there are two best-effort flows (Reference scheduler).

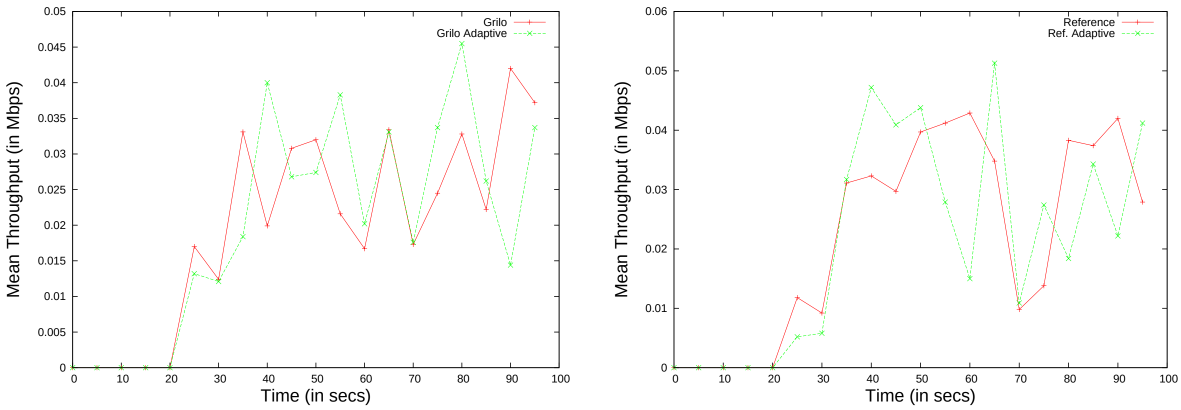


(a) Grilo scheduler



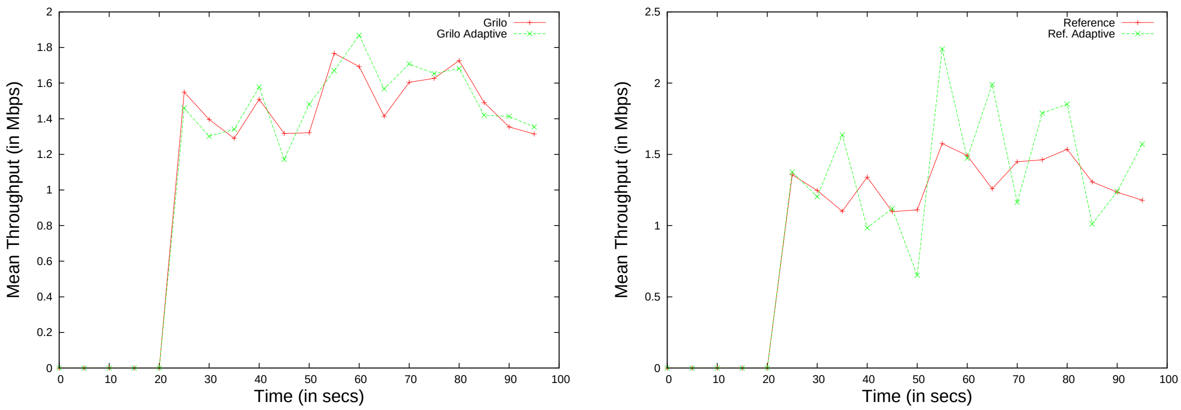
(b) Reference scheduler

Fig. 4. Performance comparison in the uniform error model, when there are ten best-effort flows.



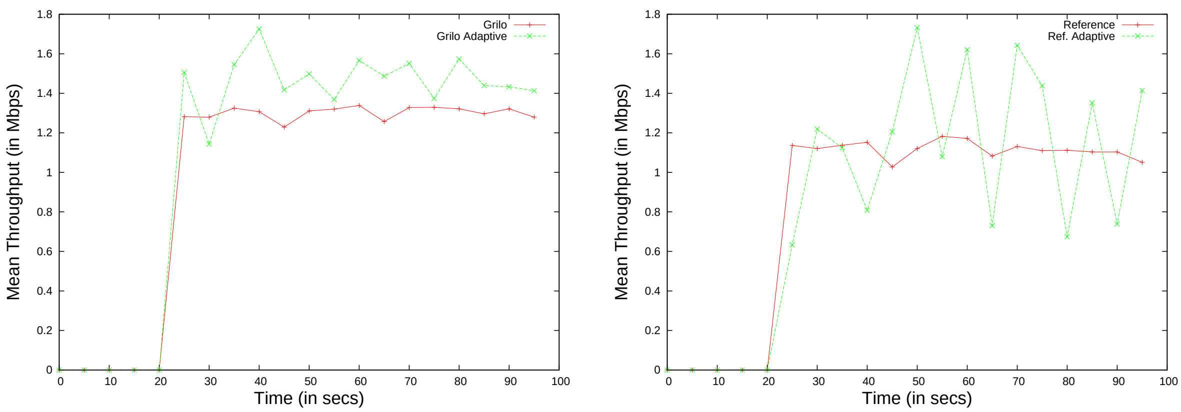
(a) Grilo scheduler (b) Reference scheduler

Fig. 5. Performance comparison in the uniform error model for VoIP traffic.



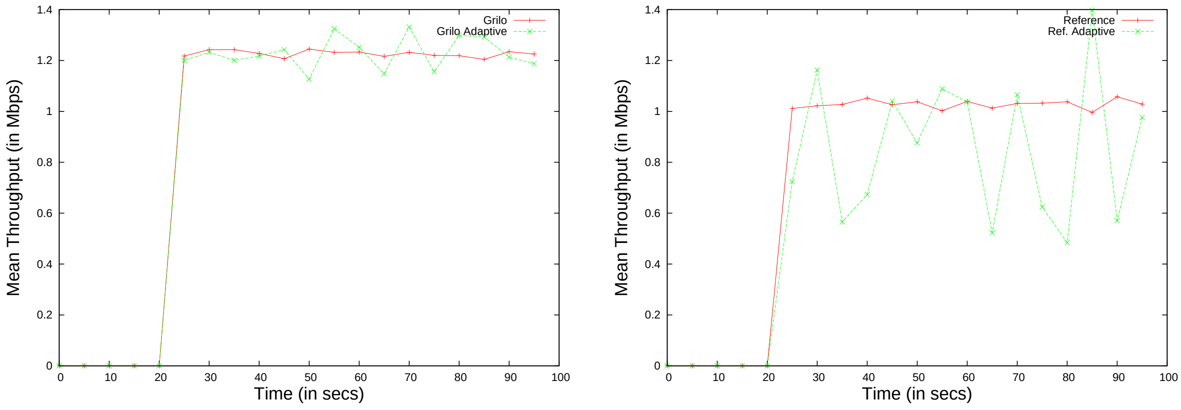
(a) Grilo scheduler (b) Reference scheduler

Fig. 6. Performance comparison in the Rayleigh fading model with velocity of 0.1 km/h.



(a) Grilo scheduler (b) Reference scheduler

Fig. 7. Performance comparison in the Rayleigh fading model with velocity of 3 km/h.



(a) Grilo scheduler

(b) Reference scheduler

Fig. 8. Performance comparison in the Rayleigh fading model with velocity of 100 km/h.

TABLE VI
THE AVERAGE DURATION (MSEC) IN BAD AND GOOD STATES

Velocity	scheduler	Bad	Good
0.1 km/h	Grilo	104.0	142.9
	Reference	167.3	67.1
3 km/h	Grilo	138.7	67.5
	Reference	264.0	48.3
100 km/h	Grilo	135.4	31.6
	Reference	195.3	23.3

improvement by our algorithm is summarized in Table V. The maximum improvement is found in the range of 15% and 18% analytically by counting the number of TXOPs given in good and bad states like (8). In cases of 0.1 km/h and 3 km/h, the average durations in bad and good states do not differ that much, so our proposed algorithm can catch up with the lending TXOPs. However, when the velocity is 100 km/h, the average duration in good state is too short to accommodate all the lagging TXOPs. Therefore, with the velocity increase, the performance of our algorithm becomes poor, and even worse than that of the legacy algorithm at the velocity of 100 km/h. The average duration in each state is listed in Table VI.

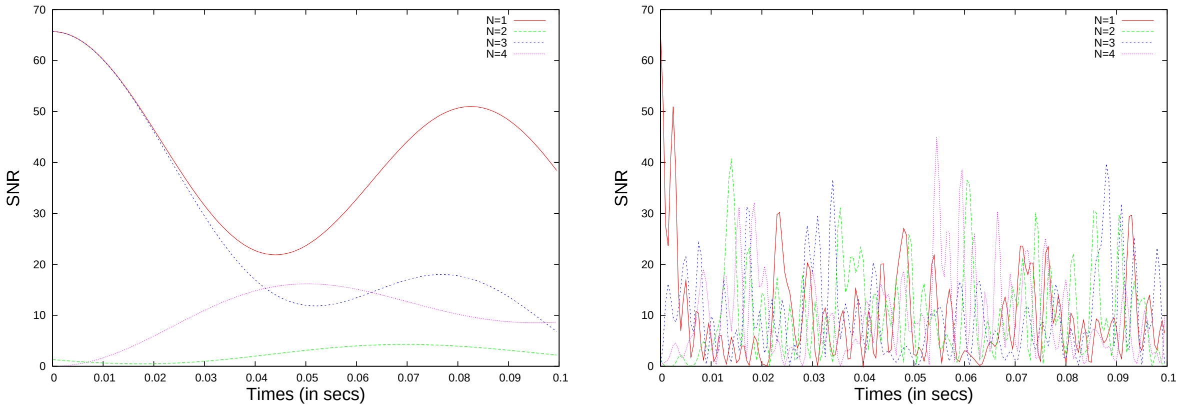
The most significant factor in our adaptive algorithm is the channel variation period. Fig. 9 depicts an SNR sample for four BE flows. The periods of SNR variation at 0.1 km/h, 3 km/h, and 100 km/h are about 2.5 sec, 80 msec, and 2.5 msec, respectively. In our simulation, we observed that the average periods of TXOP allocation are 6.4 msec and 40 msec for the Grilo and reference schedulers, respectively. When the period is too short compared to the period of SNR variation, our adaptive algorithm may not work effectively, because the SNR will be similar at the next TXOP opportunity which would be, with a better SNR, an opportunity to compensate a lagging TXOP. As a result, the adaptive reference scheduler gets the highest improvement at 0.1 km/h, and the adaptive Grilo scheduler at 3 km/h. Especially, at 0.1 km/h when the channel varies very slowly, the improvement in the adaptive Grilo scheduler is only 1.12%. On the contrary, the Grilo scheduler is more robust than the reference scheduler in fast-varying channel conditions, since the TXOP allocation period of the Grilo scheduler is shorter than that of the reference scheduler. In conclusion, our algorithm achieves better throughput than the legacy one at a low velocity. Since the IEEE 802.11 WLAN systems are typically deployed for static or nomadic environments (i.e., low mobility), our adaptive TXOP algorithm can operate well.

VI. CONCLUSION AND FUTURE WORK

This paper proposed the idea of adaptive TXOP allocation which exploits the channel condition to increase the efficacy of scheduler while ensuring long-term temporal fairness. Mathematical analysis and simulation results verified that our scheme performs better when compared to reference and Grilo standard implementations. This paper simply adapts TXOP within the pre-defined interval $[TXOP_{\max}, TXOP_{\min}]$ according to traffic and channel conditions. Future work on this proposal can also adapt SI according to channel conditions to improve the efficiency further.

REFERENCES

[1] IEEE Std 802.11e, "IEEE STANDARD FOR Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements. Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 8: Medium Acess Control (MAC) Quality of Service Enhancements", Nov. 2005.
[2] S. Mangold, S. Choi, G. Hiertz, O. Klein, and B. Walke "Analysis of IEEE 802.11e for QoS Support in Wireless LANs", *IEEE Wireless Commun.*, vol. 10, no. 6, pp. 40-50, Dec. 2003.
[3] S. Mangold, S. Choi, and P. May, "IEEE 802.11e Wireless LAN for Quality of Service", in *Proc. European Wireless*, Florence, Italy, Feb. 2002.
[4] N. Ramos, D. Panigrahi, and S. Dey "Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions", *IEEE Network*, vol. 19, no. 4, pp. 14-20, Jul. 2005.
[5] P. Ansel, Q. Ni, and T. Turletti, "An Efficient Scheduling Scheme for IEEE 802.11e", in *Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Cambridge, UK, Mar. 2004.
[6] V. Bharghavan, S. Lu, and T. Nandagopal, "Fair Queueing in Wireless Networks: Issues and Approaches", *IEEE Pers. Commun.*, vol. 6, no. 1, pp. 44-53, Feb. 1999.
[7] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 2053-2064, Oct. 2001.
[8] S. Kandala et al., "Normative Text for TGe Consensus Proposal", *IEEE 802.11-02/612r0*, Nov. 2002.
[9] A. Grilo, M. Macedo, and M. Nunes, "A Scheduling Algorithm For QoS Support In IEEE 802.11e Networks", *IEEE Wireless Commun.*, vol. 10, no. 3, pp. 36-43, Jun. 2003.



(a) 3 km/h

(b) 100 km/h

Fig. 9. SNR variation of four BE flows.

[10] Y. Chen, Q.-A. Zeng, and D. P. Agrawal "Performance Analysis of IEEE 802.11e Enhanced Distributed Coordination Function", in *Proc. 11th IEEE International Conference on Networks (ICON'03),* Sydney, Australia, Sep. 2003.

[11] N. Ramos, D. Panigrahi, and S. Dey "Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks", in *Proc. Intl. Wksp.Broadband Wireless Multimedia*, San Jose, CA, USA, Oct. 2004.

[12] D. Skyrianoglou, N. Passas, and A. K. Salkintzis "ARROW: An Efficient Traffic Scheduling Algorithm for IEEE 802.11e HCCA", *IEEE Trans. Wireless Commun.*, pp. 3558-3567, vol 5, no. 12, Dec. 2006.

[13] E. Kim and Y.-J. Suh "ATXOP: An Adaptive TXOP Based on the Data Rate to Guaantee Fairness for IEEE 802.11e WLAN", in *Proc. IEEE Vehicular Technology Conference*, Los Angeles, CA, USA, Sep. 2004.

[14] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea "A Software Architecture for Simulating IEEE 802.11e HCCA", in *Proc. 3rd Internet Performance, Simulation, Monitoring and Measurement - IPS-MoMe 2005*, Warsaw, Poland, Mar. 2005.

[15] A. Kamerman and L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band", *Bell Labs Tech. J*, pp. 118-133, Summer 1997.

[16] *ns* documentation, http://www.isi.edu/nsnam/ns/nsdocumentation.html.

[17] Soni P. M.,and Chockalingam A. "Performance Analysis of UDP with Energy Efficient Link Layer on Markov Fading Channels", *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 769-780, Oct. 2002.

[18] William C. Jakes, "*Microwave Mobile Communications*," John Wiley & Sons, Feb. 1975.

Appendix A
Proof of Lemma 1

We prove this by induction. From the observation of our algorithm we can see that compensation service (time) is updated in either of the following cases: (1) STA gets active where c_i is initialized to 0 or (2) STA is selected to receive additional TXOP.

1) Basic Step: Since our algorithm assumes that all the STAs start at the same time and hence lemma holds true at time = 0 (see case (1)).

2) Induction Step: Now c_i will be updated only when STA i is selected to receive additional TXOP from some other STA indicated as k. Let's assume that before STA i is selected, the lemma holds true. Hence, after selection, its c_i value will be changed according to the following:

$$c_i \leftarrow c_i + TXOP_{\text{lend}}, \tag{15}$$

where $TXOP_{\text{lend}}$ is the TXOP lent by STA k to STA i. Since c_i should have a minimum value among all other STAs, we have

$$c_i \leq c_j, \ \forall j. \tag{16}$$

By hypothesis, we have the following equation holding true before STA i get selected,

$$\mid c_i - c_j \mid \leq TXOP_{\max}. \tag{17}$$

Now, when STA i gets selected from (15) and (16), we obtain

$$c_i + TXOP_{\text{lend}} - c_j \leq \ TXOP_{\text{lend}} \leq \ TXOP_{\max} \quad \forall j. \tag{18}$$

The other inequality will follow in the similar fashion if we had chosen STA j, reversed with respect to i and j, with (15) and (16).

Appendix B
Proof of Theorem 1

After t_1, STA i will start getting additional service, its lag will decrease. Since the total compensation received by STA i during the interval $[t_1, t_2]$ is $c_i(t_2) - c_i(t_1)$, we have

$$lag_i(t_2) = lag_i(t_1) - (c_i(t_2) - c_i(t_1)). \tag{19}$$

Let $C(t_1, t_2)$ be the total additional TXOP received by all STAs in this interval, and let $L(t_1, t_2)$ denote the set of all lagging STAs in the same interval. Since we have at least one error-free lagging STA for this interval, all the additional TXOP will go to lagging STAs till they all get $TXOP_{\max}$. Our assumption that leading STAs always have TXOP to lend will be lifted after arriving at this result. Now, considering the worst case that all the STAs in $L(t_1, t_2)$ are error-free, which means they can accept additional TXOP at anytime during the interval $[t_1, t_2]$, we have

$$C_i(t_1, t_2) \leq \sum_{j \in L(t_1, t_2)} (c_j(t_2) - c_j(t_1)). \tag{20}$$

Using Lemma 1, for any error free STAs during the interval, we have

$$c_j(t_2) - c_j(t_1) \leq c_i(t_2) - c_i(t_1) + 2 \cdot TXOP_{\max}. \tag{21}$$

Then we obtain

$$\begin{aligned} C_i(t_1,t_2) &\leq \sum_{j \in L(t_1,t_2)} (c_j(t_2) - c_j(t_1)) \\ &\leq \sum_{j \in L(t_1,t_2)} (c_i(t_2) - c_i(t_1) + 2 \cdot TXOP_{\max}) \\ &\leq (c_i(t_2) - c_i(t_1) + 2 \cdot TXOP_{\max}) \cdot N \end{aligned} \tag{22}$$

where N is the number of actual QoS STAs in the interval (t_1, t_2) which will be at least $|L(t_1, t_2)| + 1$ (minimum number of leading stations).

The worst case in the network would be if there is only one leading STA to give back compensation service. At any point, it will receive $TXOP_{\min}$ and will compensate rest for the lagging STA by $(TXOP_{\text{ideal}} - TXOP_{\min})$. Therefore for the whole interval, we have

$$C_i(t_1,t_2) \geq \Phi \cdot (TXOP_{\max} - TXOP_{\min}), \tag{23}$$

where Φ is the time interval $(t_2 - t_1)$. Using (22) and (23), and putting $lag_i(t_2)$ as zero, which will be the time that STA i will be satisfied, we can solve for Φ as

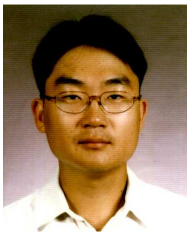
$$\Phi \leq \frac{N \cdot (lag_i(t_1) + 2 \cdot TXOP_{\max})}{TXOP_{\max} - TXOP_{\min}}. \tag{24}$$

To relax the assumption made earlier, we assume that on the average a leading STA can lend TXOP after β time interval. Finally we obtain

$$\Phi \leq \frac{1}{\beta} \frac{N \cdot (lag_i(t_1) + 2 \cdot TXOP_{\max})}{TXOP_{\max} - TXOP_{\min}}. \tag{25}$$

Abhinav Arora received Bachelor of Technology in Computer Science from the Indian Institute of Technology (IIT), Guwahati, India. He began his career with Samsung, India. In 2004, he received Samsung Global Scholarship to pursue Master in Electrical Engineering at Seoul National University. At present, he is a project engineer with Samsung Electronics, South Korea and is currently working on developing and commercializing vertical handover technologies. He also has a chapter published in the book, " Advances in Security and Payment Methods for Mobile Commerce".

Sung-Guk Yoon Sung-Guk Yoon is currently a Ph.D. candidate in the School of Electrical Engineering and Computer Science, Seoul National University. He received his B.S. degree from Seoul National University, in 2006. His research interests include fourth-generation wireless networks, cross-layer optimization, resource management, and power line communication.



Young-June Choi received B.S., M.S., and Ph.D. degrees from the Department of Electrical Engineering & Computer Science, Seoul National University, in 2000, 2002, and 2006, respectively. From Sept. 2006 through July 2007, he was a postdoctoral researcher at the University of Michigan, Ann Arbor, MI, USA. From 2007 through 2009, he was with NEC Laboratories America, Princeton, NJ, USA, as research staff member. He is currently an assistant professor at Ajou University, Suwon, Korea. He was awarded a Gold Prize at the Samsung Humantech Thesis Contest in 2006 and holds 10 US/Korean patents. His research interests include fourth generation wireless networks, radio resource management, and cognitive radio networks. He is a member of ACM and IEEE.

Saewoong Bahk received B.S. and M.S. degrees in Electrical Engineering from Seoul National University in 1984 and 1986, respectively, and the Ph.D. degree from the University of Pennsylvania in 1991. From 1991 through 1994 he was with AT&T Bell Laboratories as a member of technical staff where he worked for AT&T network management. In 1994, he joined the school of electrical engineering at Seoul National University and currently serves as a professor. He has been serving as TPC members for various conferences including ICC, GLOBECOM, INFOCOM, PIMRC, WCNC, etc. He is on the editorial board of Journal of Communications and Networks (JCN). His areas of interests include performance analysis of communication networks and network security. He is an IEEE senior member and a member of Who's Who Professional in Science and Engineering.