An Analytic Study of Tuning Systems Parameters in IEEE 802.11e Enhanced Distributed Channel Access

Ye Ge
Dept of Electrical Engineering
Ohio State University
Columbus, OH 43210
Email: ge.14@osu.edu

Jennifer C. Hou
Dt of Computer Science
Univ. of Illinois at Urbana-Champaign
Urbana, USA 61801
Email: jhou@cs.uiuc.edu

Sunghyun Choi
School of Electrical Engineering
Seoul National University
Seoul, Korea
Email: schoi@snu.ac.kr

Abstract

In this paper, we derive, based on the analytical model developed by Cali *et al.*, a multi-class model to study how to adaptively tune all the parameters in IEEE 802.11e EDCA and support service differentiation in WLANs. Through analytical modeling, we demonstrate that by assigning appropriate different transmission probabilities (or contention window sizes) to stations of different classes, it is feasible to provide (proportional) service differentiation and achieve pre-specified targeted throughput ratios among different classes, while at the same time, maximizing the total system capacity. We also extend the derived theoretical model to analyze the role of AIFS and TXOP values on service differentiation perceived by different traffic classes. We show that, to achieve QoS guarantees (i.e. throughput differentiation) and high channel utilization, it may not be desirable to allow tuning of multiple parameters (e.g., both the contention window sizes and the AIFS values). Instead, the design dimension should be kept small by allowing turning of only one set of parameters, while keeping the other two sets of parameters for all the access categories fixed at the same value (i.e., setting the AIFS values of all access categories to 2, which is equivalent to $AIFS = DIFS$).

We also elaborate on several implementation issues on incorporating theoretical results into IEEE 802.11e. These include (i) how to reduce the computational complexity and practically calculate results on-line, (ii) how to convert the optimal parameters derived in the model that characterizes the p-persistent version of IEEE 802.11e to those in IEEE 802.11e (which is based on the notion of the contention window to determine whether or not to transmit in a slot), and (iii) how to on-line measure parameters needed for calculating the best value of the contention window size. Both the analytical models and the proposed approaches for practically incorporating theoretical findings into IEEE 802.11e EDCA are validated through detailed *ns-2* simulations and empirical experimentation on a Linux-based MADWifi driver for wireless LAN devices with the Atheros chipset.

I. INTRODUCTION

IEEE 802.11 based wireless Local Area Networks (WLANs) have become popular at an unprecedented rate. Besides conventional Internet applications such as email, file transfer, and web access and browsing, WLANs are also expected to support QoS-centric applications such as audio/video streaming in smart home environments. The latter applications are delay sensitive and require certain level of throughput and delay guarantees. This creates a urgent need for supporting QoS in 802.11-based WLANs.

The current IEEE 802.11 standard [24] defines two access methods: (i) the Distributed Coordination Function (DCF), also known as the basic access method, is a carrier sense multiple access protocol with collision avoidance (CSMA/CA); (ii) the Point Coordination Function (PCF) is a polling-based access method and uses a point coordinator to arbitrate access among stations. In DCF, all the data traffic is transmitted on a first come first serve, best-effort basis. There is no notion of priorities and all the stations in the basic service set (BSS) contend for the wireless medium with the same priority. When the number of stations in a BSS increases, the probability of collisions increases, leading to frequent retransmission and a decrease in the overall throughput (and QoS) [8]. PCF, on the other hand, was designed to support time-bounded traffic, and defines two periods between transmission of two consecutive Delivery Traffic Indication Message (DTIM) beacon frames: Contention Free Period (CFP) and Contention Period (CP). Beacon frames are sent periodically by the access point (AP), and carry synchronization and network (BSS) information. In particular, DTIM beacon frames are used to indicate the start of a CFP. During a CP all the stations contend for the wireless medium using DCF, while during a CFP the AP schedules transmissions to and/or from individual stations using a polling mechanism. In spite of the intention to support time-bounded traffic, many inadequacies have been identified in the design [15]: (i) unpredictable beacon delays resulting in significantly shortened contention free periods (CFPs), and (ii) unknown transmission durations of polled stations making it very difficult for the AP to predict and control the polling schedule for the remainder of the CFP. In addition, there has no management interface defined to set up and control PCF operations and to communicate QoS requirements to the AP. As a result, it is difficult, if not impossible, to set up a PCF policy which is compatible to traffic policies in the Internet IntServ/DiffServ architectures.

To deal with the inadequacy of legacy IEEE 802.11 DCF and PCF access mechanisms in supporting QoS, several schemes have been proposed in literature that aim to enhance IEEE 802.11 DCF and provide certain service differentiation between different traffic classes. Schemes proposed in [1, 13, 27] tune the values of parameters in DCF (CW_{min}, CW_{max}, inter-space values, and maximum frame lengths) to achieve service differentiation. Schemes proposed in [6, 7, 23, 30, 31] adaptively adjust the transmission probability (or the contention window size) to both differentiate services among different traffic classes and to improve channel utilization. Vaidya *et al.* [26] and Banchs *et al.* [2] apply the concept of fair queueing to achieve differentiated channel bandwidth distribution among different traffic classes. Qiao *et al.* [22], Li and Battiti [17], and Xiao [29] extend the Markov chain model in [5,3,4] to the case of multiple priority classes. Based on the extended analytical model, Qiao *et al.* then proposes a priority based fair medium access control protocol P-MAC, Li and Battiti study how to scale the minimum contention window size and the length of the packet payload according to the priority of each traffic flow under the saturated scenario, while Xiao [29] incorporates three adjustable parameters (the initial contention window size, the retry limit, and the backoff window-increasing factor) into the model.

On the industry side, IEEE 802.11 has chartered a Working Group E to investigate possible enhancement to IEEE 802.11 MAC. In particular, the IEEE 802.11e draft standard defines a new coordination function called the *Hybrid Coordination Function (HCF)*. HCF includes both a contention-based channel access method, called the *Enhanced Distributed Channel Access (EDCA)* mechanism, for contention based data transmission, and a controlled channel access, referred to as the *HCF Controlled Channel Access (HCCA)* mechanism, for contention free data transmission. At a very high level, EDCA envisions an architecture of multiple access categories and assigns to different access categories different values of the minimum idle delay before contention (AIFS), the minimum and maximum contention windows (CW_{min} and CW_{max}), and the transmission opportunity limit (TXOP). Its performance with respect to service differentiation has been studied via simulation in [12, 20, 18, 19, 21]. (We will give a more detailed summary in Section II.)

Most of the aforementioned research/industry efforts on providing QoS in legacy IEEE 802.11 DCF [1, 13, 27, 23, 31, 6, 7, 30, 26] and on evaluating IEEE 802.11e [12, 20, 18, 19, 21] are based on simulation. The only exception is perhaps those reported in [22, 29, 17]. In particular, the analytical model derived in [29] can be used to study the performance of IEEE 802.11e in terms of the saturation throughput, the saturation delay and the frame dropping probability. However, the important tunable parameter in the IEEE 802.11e draft standard, AIFS, is not figured into the model. (As a matter of fact, a detailed analysis on how AIFS values affect service differentiation among different traffic classes has been lacking.) The issue of on-line fine tuning the above parameters to meet the QoS requirements of different access categories in the presence of network dynamics is also not discussed in [22, 29].

In this paper, we bridge the gap and derive, based on the analytical model developed in [8], a multi-class model to study how to adaptively tune parameters in IEEE 802.11e EDCA and support service differentiation in WLANs. We only consider EDCA, because EDCA bears some similarities to DCF and there are several analytical models that characterize data transmission activities governed by DCF and for which we will leverage. (We leave the task of analyzing data transmission activities governed by HCCA and tuning their parameters as part of our future work.) Through analytical modeling, we demonstrate that by assigning appropriate different transmission probabilities (or different contention window sizes in the case of contention window-based IEEE 802.11e) to stations of different classes, it is feasible to provide service differentiation and achieve pre-specified targeted throughput ratios among different classes, while at the same time maximizing the total system capacity. We also extend the derived theoretical model to incorporate the effect of using different AIFS and TXOP values on service differentiation perceived by different traffic classes. We show that, to achieve QoS guarantees (i.e. throughput differentiation) and high channel utilization, it may not be desirable to allow tuning of multiple parameters (e.g., both the contention window sizes and the AIFS values). Instead, the design dimension should be decreased by allowing turning of only one set of parameters, while keeping the other two sets of parameters for all the access categories fixed at the same value (i.e., setting the AIFS values of all access categories to 2, which is equivalent to $AIFS = DIFS$).

We also elaborate on several implementation issues on incorporating theoretical results into IEEE 802.11e. These include (i) how to reduce the computational complexity and practically calculate results on-line, (ii) how to convert the optimal parameters derived in the model that characterizes the p-persistent version of IEEE 802.11e to those in IEEE 802.11e (which is based on the notion of the contention window), (iii) how to on-line measure parameters needed for calculating the best value of the contention window size, and and (iv) how to extend the analytic model to incorporate the concept of Transmission Opportunity (TXOP). Both the analytical models and the proposed approaches for practically incorporating theoretical findings into IEEE 802.11e EDCA are validated through detailed *ns-2* simulations and empirical experimentation on a Linux-based MADWifi driver for wireless LAN devices with the Atheros chipset.

The rest of the paper is organized as follows. In Section II, we give a summary of IEEE 802.11e EDCA operations that pertain to our work. In Section III, we present our analytical model for providing service differentiation among multiple traffic classes and maximizing the channel utilization for IEEE 802.11 e EDCA. In Section IV, we extend our analytical model to analyze the role of AIFS and TXOP values on service differentiation perceived by different traffic classes. Following that, we discuss in Section V several implementation issues for practically incorporating theoretical results into IEEE 802.11e EDCA, and present a performance study in Section VI, evaluating IEEE 802.11e EDCA (with its parameters on-line tuned by the analytical model). Finally, we conclude the paper in Section VII, with a list of research avenues for future work.

II. Preliminaries

In this section, we summarize the operations of IEEE 802.11e EDCA that pertain to our work. As EDCA contention access is an extension of the legacy CSMA/CA DCF mechanism to support traffic of different priorities, we first summarize the operations of IEEE 802.11 DCF.

A. IEEE 802.11 Distributed Coordination Function

The basic CSMA/CA mechanism in DCF operates as follows. Each station maintains a contention window (CW) and a back-off timer. When a station has a frame to transmit, it senses the medium first. If the medium is sensed idle for more than a time interval of *Distributed InterFrame Space (DIFS)*, it starts to transmit the frame at the beginning of the immediately following slot. Otherwise, the station defers its frame transmission according to the *binary exponential backoff* algorithm: The station waits until the medium has become idle for an interval of DIFS and then sets its backoff timer. The backoff timer gives the additional time it should wait before the next transmission attempt, and is set as the product of *aStotTime* and a pseudorandom integer that is drawn from a uniform distribution over the interval $[0, CW]$. The contention window CW is set to its minimum value CW_{min} ($CW_{min} = 31$) for the first time a frame is backed off. Every time the frame incurs a collision, CW is increased to $(CW + 1) \times 2 - 1$, with the restriction that CW can not exceed its maximum possible value CW_{max} ($CW_{max} = 1023$). The backoff timer is decreased by one time slot for every consecutive idle slot after the medium has been sensed idle for an interval of DIFS, and is suspended whenever the medium becomes busy. The backoff process resumes when the wireless medium has been detected to be idle for an interval of DIFS. When the backoff timer reaches zero, the station transmits the frame immediately.

If the destination station receives the frame correctly, it sends a positive acknowledgment (ACK) frame after a time interval of *Short InterFrame Space (SIFS)*. Note that SIFS is shorter than DIFS. When the source station does not receive an ACK within an *ACKTimeout* interval, it assumes the frame has experienced a collision, updates the contention window CW according to the binary exponential backoff algorithm described above, and sets its backoff timer according to a newly selected random value within $[0, CW]$. A maximum of 7 retransmissions (4 retransmissions) for short frames (long frames) are allowed before the frame is dropped. The basic access procedure is depicted in Fig. 1 (a). To further decrease the overhead caused by frame collision and hidden terminal effects, Request-To-Send (RTS) and Clear-To-Send (CTS) frames may also be exchanged before the data transmission.

It is worth noting that collision avoidance is achieved by a virtual carrier sense mechanism. Whether the channel is idle or busy is not solely determined by the physical carrier sense result, but also by the value of the NAV timer maintained by the MAC. A duration value is included in each frame that is transmitted by a station and indicates how long the transmission will last, including any subsequent acknowledgments and fragments. Each station in the vicinity of the transmitting station receives the frame and uses the duration value to update its NAV. Therefore the NAV value indicates how long another station has access to the wireless medium no matter what is the real activity of that station.

B. IEEE 802.11E Enhanced Distributed Channel Access

As mentioned in Section I, IEEE 802.11e defines a new HCF coordination function that includes both the contention-based channel access method, EDCA, and the controlled channel access method, HCCA. As EDCA is essentially an extension of the legacy DCF mechanism for which several analytical models exist and can be leveraged, we focus on EDCA in this paper.

In EDCA, several parameters that control how and when a node gains access to the medium (e.g., the contention window, the inter-frame space, and the transmission opportunity) differ among different priority levels, so as to favor/disfavor data transmission from high-priority/low-priority flows. A total of eight user priority levels are available, and mapped to four access categories (ACs). Each AC corresponds to one of the four transmit queues that implement the EDCA contention algorithm with different parameters. These parameters are the minimum idle delay before contention (AIFS), the minimum and maximum contention windows (CW_{min} and CW_{max}), and the transmission opportunity limit (TXOP). Specifically, the EDCA functions under different ACs wait for different values of *Arbitration Interframe Space (AIFS)*, rather than using the same value of DIFS after channel has become idle. Fig. 1 (b) depicts the relation between the various interframe space (IFS) parameters.

Similarly, EDCA associates different ACs with different values of CW_{min} and CW_{max}, and allows traffic of different priorities to back off for different time intervals, so as to increase/decrease their probability of medium access. Finally, a station that wins an EDCA contention is granted a Transmission Opportunity (TXOP)—the right to use the medium for a period of time. The duration of this TXOP is specified per access category, and is contained in the TXOP limit field of the access category (AC) parameter record in the EDCA parameter set. A station can use a TXOP to transmit multiple frames within an access category. If the DATA-ACK exchange sequence has been completed, and there is still time remaining in the TXOP, the station may transmit another frame in the same access category, provided that the frame to be transmitted and its necessary acknowledgment can fit into the time remaining in the TXOP. All the parameters can be dynamically updated by the QoS access point (QAP) for each access category through the EDCA parameter set, and are sent from the QAP as part of the beacon frames, and probe/re-association response frames. This adjustment allows stations in the WLAN to adapt to changing conditions, and gives the QAP the ability to manage the overall QoS performance.

When data frames arrive at the 802.11e MAC layer, they are classified into an appropriate AC and enqueued into the corresponding transmit queue. When frames are available in multiple transmit queues for transmission, contention for the medium occurs both internally and externally, based on the same coordination function, so that internal scheduling resembles external scheduling. Internal collisions are resolved by allowing the frame with a higher priority to be transmitted, while the flow with a lower priority invokes a queue-specific backoff procedure as if it had incurred collision.

III. Analytical, Multi-Class Model

The first step to analyzing how to tune all the parameters in IEEE 802.11e EDCA and supporting service differentiation is to devise an analytical model that characterizes the data transmission activities under EDCA. Our analytical model is built upon, and extends, that proposed by Cali *et al.* in [8, 9, 10]. Essentially Cali's work considers channel access by a *single* traffic class in a *p-persistent* version of IEEE 802.11 DCF, with the objective of improving the system channel utilization. They do not provide service differentiation among multiple traffic classes.

We assume all stations can hear each other, and hence there are no hidden terminal and exposed terminal problems. As the major application scenario we consider is an managed wireless LAN in hotspots (such as airports, restaurants and hotels) or home networks for wireless audio/video distribution, the assumption is valid.

We first summarize the basic model. Note that the basic model is slightly different from that in Cali's work, due to the fact that Cali em et al. assume that each station has to wait for an interval of DIFS after a frame collision in their model, while we assume EIFS.[1] Then we elaborate on the proposed multi-class model. The results derived in this section and Section IV can be used as a guideline to fine-tune parameters in real networks to achieve desired service differentiation.

A. Basic Model

For analytical tractability, Cali *et al.* $[8, 9, 10]$ consider a *p-persistent* version of IEEE 802.11 DCF, which differs from the standard protocol only in the selection of the backoff interval. Instead of using the binary exponential backoff timer values, the *p-persistent* version determines its backoff interval by sampling from a geometric distribution with parameter p. Due to the memoryless property of this geometric-distributed backoff algorithm, it is more tractable to analyze the *p-persistent* IEEE 802.11 protocol.

The analytic model is derived under the assumption that all the stations always have packets ready for transmission (which is termed the $asymptotic$ condition in $[8, 9, 10]$). Under the geometrically-distributed backoff assumption, the process that characterizes the occupancy behavior of the channel (idle slots, collisions, and successful transmission) till the end of each successful transmission is regenerative, with the sequence of time instants corresponding to the completion of successful transmission being the regenerative points. Cali *et al.* exploit this regenerative property and define the jth virtual transmission time as the time interval between the *j*th and *(j+1)*th successful transmissions. In such a time period, idle periods and collisions precede a successful transmission, where an idle period is a time interval in which the channel is idle due to the fact that all the back-logged stations are in the back-off mode, and a collision is the interval in which two or more stations attempt for transmission and their packets collide with one another.

Let $E(m)$ denote the average message duration, and $E(T_v)$ the average virtual transmission time. It then follows that the channel utilization ρ can be expressed as

$$\rho = \frac{E(m)}{E(T_v)}. \tag{1}$$

Let I_i and $T_{c,i}$ denote the lengths of the ith idle period and the ith collision in a virtual transmission time respectively, N_{col} the number of collisions in a virtual transmission time, and S the time required to complete a successful transmission. Note that S is the time interval between the start of a successful frame transmission and the receipt of the corresponding ACK plus a DIFS, i.e.,

$$S = m + SIFS + ACK + DIFS,$$

[1]As will be elaborated on later, it is non trivial to determine whether a specific listening station will use a DIFS or an EIFS after a frame collision.

where m is the time it takes to transmit a message. Then, we have

$$\begin{aligned} E(T_v) &= E\left(\sum_{i=1}^{N_{col}} (DIFS + I_i + T_{c,i} + SIFS + ACK)\right) + E\left(I_{N_{col}+1}\right) + E\left(S\right) \\ &= E\left(N_{col}\right) \cdot \left(E\left(T_c\right) + DIFS + SIFS + ACK\right) + \left(E\left(N_{col}\right) + 1\right) \cdot E\left(I\right) \\ &\quad + E\left(m\right) + SIFS + ACK + DIFS, \end{aligned} \tag{2}$$

where the $SIFS$ and ACK in the first term on the right hand side of Eq.(2) is due to the extra waiting period in the extended inter-frame space (EIFS) after detection of a incorrectly-received frame (i.e., frame collision). Note that in Cali's model, it is assumed that each station waits for an interval of DIFS after a frame collision, while we assume the use of EIFS here. In fact, it is non trivial to determine whether a specific listening station should use a DIFS or an EIFS after a frame collision. If the listening station is able to synchronize with one of the colliding frames, and thus initiate a receiving process, an EIFS will be used. Otherwise (e.g., in the case that the received power level is comparable for the two colliding frame preambles), the listening station will only perceive a busy channel. In this case, without initializing the reception of any frame, the station will use a DIFS. All the simulation (as well as analytical) models we are aware of do not take into account this technical issue, but simply use either DIFS or EIFS exclusively for all the collisions.

B. Proposed Analytical Model

For tractability of analysis, we follow Cali's methodology and consider a *p-persistent* version of IEEE 802.11e EDCA. To support multiple traffic priorities, we assume each traffic class use a different probability, p, to access the channel. For the time being, we assume that all the traffic classes use DIFS as the interframe space value. (We will incorporate the effects of using different AIFS and TXOP values on service differentiation in Section IV-A.)

To facilitate the analysis, we make the following assumptions and notations:

A1) There are P classes of stations, each of which contains

A2) A class-i station transmits its frame in a slot (after the medium becomes idle for DIFS) with probability p_i in the *p-persistent* version of IEEE 802.11e EDCA ($1 \leq i \leq P$).

A3) The size, m_i, of a packet sent by a class-i station is uniformly distributed between (x_0, x_1), i.e.,

$$P(m_i \leq x) = \begin{cases} 0 & x < x_0, \\ \frac{(x-x_0)}{(x_1-x_0)} & x_0 \leq x \leq x_1, \\ 1 & x > x_1. \end{cases}$$

A4) All the stations always have a packet ready for transmission (i.e., the *asymptotic condition* holds).

Recall that the channel utilization in the case of a single class is expressed in Eq. (1). By defining $P_{pkt}(i) = P$ (the packet successfully transmitted in a virtual transmission time is of class-i), the utilization attained by class-i stations can be expressed as

$$\rho_i = \frac{E(m_i) \cdot P_{pkt}(i)}{E(T_v)}, \tag{3}$$

the utilization ratio between class-i and class-j as

$$r_{ij} = \frac{\rho_i}{\rho_j} = \frac{E(m_i)P_{pkt}(i)}{E(m_j)P_{pkt}(j)}, \tag{4}$$

and the utilization ratio between a class-i station and a class-j station as

$$\widehat{r}_{ij} = \frac{\rho_i/N_i}{\rho_j/N_j} = \frac{N_j}{N_i} \cdot r_{ij}. \tag{5}$$

To derive ρ and ρ_i, we need to derive $E(T_v)$ and $P_{pkt}(i)$. As mentioned above, in the case of a single class, $E(T_v)$ is given in Eq. (2). It is easy to see that Eq. (2) still holds in the case of multiple priority classes, except that $E(N_{col})$, $E(T_c)$, and $E(I)$ have to be derived. In what follows, we derive, following the same line of reasoning in [9], $E(N_{col})$, $E(T_c)$, and $E(I)$ in the case of multiple priority classes. Then, we derive $P_{pkt}(i)$.

Derivation of $E(N_{col})$: Let N_{ts} denote the number of transmitting stations in the slot immediately after an idle interval of length DIFS, and let $P_{collision}$ and $P_{success}$ denote, respectively, the probability that a collision occurs and that a transmission is successful, both conditioned on that at least one station transmits. Then,

$$\begin{aligned} P_{collision} &= P\left(N_{ts} \geq 2 \mid N_{ts} \geq 1\right) = \frac{1 - P\left(N_{ts} = 0\right) - P\left(N_{ts} = 1\right)}{1 - P\left(N_{ts} = 0\right)} \\ &= \frac{1 - \prod_{i=1}^{P}(1-p_i)^{N_i} - \sum_{i=1}^{P} N_i p_i (1-p_i)^{N_i - 1} \prod_{j \neq i}(1-p_j)^{N_j}}{1 - \prod_{i=1}^{P}(1-p_i)^{N_i}}, \end{aligned} \tag{6}$$

and

$$P_{success} = P\left(N_{ts} = 1 | N_{ts} \geq 1\right) = \frac{\sum_{i=1}^{P} N_i p_i (1-p_i)^{N_i - 1} \prod_{j \neq i}(1-p_j)^{N_j}}{1 - \prod_{i=1}^{P}(1-p_i)^{N_i}}. \tag{7}$$

The probability distribution of N_{col} can then be expressed as

$$P\left(N_{col} = j\right) = P_{Collision}^{j} \cdot P_{Success}, \tag{8}$$

and $E(N_{col})$ as

$$\begin{aligned} E\left(N_{col}\right) &= \sum_{j=0}^{\infty} j \cdot P\left(N_{col} = j\right) \\ &= \frac{1 - \prod_{i=1}^{P}(1-p_i)^{N_i}}{\sum_{i=1}^{P} N_i p_i (1-p_i)^{N_i - 1} \prod_{j \neq i}(1-p_j)^{N_j}} - 1. \end{aligned} \tag{9}$$

Derivation of $E(T_c)$: Recall that since IEEE 802.11 does not implement any collision detection mechanism, once a collision occurs, it lasts until all the colliding packets have completed their transmissions. The length of a collision is hence equal to the maximum length of the colliding packets. Specifically, let L_j denote the length of the packet sent by the jth colliding station, N_{cp} the total number of colliding stations, and N_{cp_i} the number of colliding class-i stations, then

$$T_c = \max\left(L_1, L_2, \ldots, L_{N_{cp}}\right). \tag{10}$$

$E(T_c)$ can be expressed as

$$\begin{aligned} E\left(T_c\right) &= \sum_{\substack{j_1=0 \\ (N_{cp}>1)}}^{N_{i_1}} \sum_{j_2=0}^{N_{i_2}} \cdots \sum_{j_P=0}^{N_{i_P}} [E(T_c \mid N_{cp_1}=j_1, N_{cp_2}=j_2, \ldots, N_{cp_P}=j_P) \\ &\quad \times P(N_{cp_1}=j_1, N_{cp_2}=j_2, \ldots, N_{cp_P}=j_P \mid N_{cp}>1)]. \end{aligned} \tag{11}$$

Under the assumption of **(A3)**, the conditional expectation value of T_c, $E(T_c \mid N_{cp_1}=j_1, N_{cp_2}=j_2, \ldots, N_{cp_P}=j_P)$, can be derived as

$$\begin{aligned} E(T_c \mid N_{cp_1}=j_1, N_{cp_2}=j_2, \ldots, N_{cp_P}=j_P) &= \int_{x_0}^{x_1} x d\left(P\left(T_c \le x\right)\right) \\ &= \int_{x_0}^{x_1} x d\left(\left(\frac{x-x_o}{x_1-x_0}\right)^{\sum_{i=1}^{P} j_i}\right) \\ &= x_1 - (x_1-x_0) / \left(\sum_{i=1}^{P} j_i + 1\right), \end{aligned} \tag{12}$$

and the term, $P(N_{cp_1}=j_1, N_{cp_2}=j_2, \ldots, N_{cp_P}=j_P \mid N_{cp}>1)$, can be calculated as

$$\begin{aligned} &P(N_{cp_1}=j_1, N_{cp_2}=j_2, \ldots, N_{cp_P}=j_P \mid N_{cp}>1) \\ &= \frac{\prod_{i=1}^{P} \begin{pmatrix} N_i \\ j_i \end{pmatrix} p_i^{j_i}(1-p_i)^{N_i-j_i}}{1-P(N_{ts}=0)-P(N_{ts}=1)} \\ &= \frac{\prod_{i=1}^{P} \begin{pmatrix} N_i \\ j_i \end{pmatrix} p_i^{j_i}(1-p_i)^{N_i-j_i}}{1-\prod_{i=1}^{P}(1-p_i)^{N_i} - \sum_{i=1}^{P} N_i p_i (1-p_i)^{N_i-1} \prod_{j\neq i}(1-p_j)^{N_j}}. \end{aligned} \tag{13}$$

By substituting Eqs. (12) and (13) into Eq. (11), $E(T_c)$ can be calculated.

Derivation of $E(I)$: Since a class i station may transmit in a slot with probability p_i, we have

$$\begin{aligned} E(I) &= t_{slot} \cdot \sum_{j=1}^{\infty} j \cdot \left(P\left(N_{ts}>0\right) \cdot \left(P\left(N_{ts}=0\right)\right)^j\right) \\ &= t_{slot} \cdot \frac{\prod_{i=1}^{P}(1-p_i)^{N_i}}{1-\prod_{i=1}^{P}(1-p_i)^{N_i}}. \end{aligned} \tag{14}$$

Derivation of $P_{pkt}(i)$: The probability that the packet successfully transmitted in a virtual transmission time is of class i can be derived as

$$P_{pkt}(i) = \frac{N_i p_i (1-p_i)^{N_i-1} \prod_{j\neq i}^{P} (1-p_j)^{N_j}}{\sum_{k=1}^{P} N_k p_k (1-p_k)^{N_k-1} \prod_{j\neq k}^{P} (1-p_j)^{N_j}}. \quad (15)$$

Finally we have

$$\begin{aligned} \frac{P_{pkt}(i)}{P_{pkt}(j)} &= \frac{N_i p_i (1-p_i)^{N_i-1} \prod_{k\neq i}^{P} (1-p_k)^{N_k}}{N_j p_j (1-p_j)^{N_j-1} \prod_{k\neq j}^{P} (1-p_k)^{N_k}} \\ &= \frac{N_i \cdot p_i \cdot (1-p_j)}{N_j \cdot p_j \cdot (1-p_i)}. \end{aligned} \quad (16)$$

By plugging Eqs. (9), (11), and (14) into Eq. (2), one obtains the expression of ρ in the multiple priority class case.

Determination of p_i values that maximize the channel utilization Without loss of generality, we express all the flow throughput requirements in terms of the relative ratio to a class 1 flow (i.e., $\widehat{r}_{i1}$). For clarity of presentation, we also assume that the data frame size of all traffic classes are of the same distribution, that is, $\forall i, \forall j,\ E(m_i) = E(m_j)$, then we can express p_i as a function of p_1 using Eq. (4) and Eq. (16), i.e.,

$$p_i = \frac{\widehat{r}_{i1} \cdot p_1}{\widehat{r}_{i1} \cdot p_1 + (1-p_1)}. \quad (17)$$

Now the protocol capacity can be optimized by finding the optimal value of p_1 that maximizes ρ (Eq. (1)) subject to the constraint on the relation between p_i an p_1 (Eq. (17)). In Section V-B, we will show how to convert the optimal value of p_i in the p-persistent version to the optimal value of the contention window in the contention-window-based EDCA.

IV. Incorporating the Effect of AIFS and TXOP into the Analytical Model

A. *Incorporating AIFS in the Analytical Model*

In Section III, we assume after a busy period, all the stations have to wait for the channel to become idle for a time interval of DIFS before they start to decrease their backoff timers. Recall that in the IEEE 802.11e draft, different AIFS values can be assigned to different traffic classes, which represents another dimension of design freedom. Table I gives the default values of CW_{min}, CW_{max} and *AIFSN* for different traffic classes. Note that for access category i, its $AIFS$ value $AIFS[i]$ is determined by its $AIFSN$ value $AIFSN[i]$, that is, $AIFS[i] = SIFS + AIFSN[i] \times aSlotTime$. Although it is intuitive to assign larger AIFS values to low-priority traffic for service differentiation, it is not clear to what extent extending the design space along this dimension facilitates service differentiation. Moreover, the default values specified in the draft standard (Table I) are not backed up by theoretic analysis. In this section, we extend our analysis model in Section III and study the impact of different AIFS values on the channel throughput attained by flows of different traffic classes.

For ease of analysis, we consider only two traffic classes. As it is likely that the AIFS values are only used to differentiate low priority background traffic from high priority traffic, an analysis with two traffic classes shed lights on how AIFS values affect QoS differentiation. Moreover, the model can be straightforwardly extended to the case of multiple traffic classes. To facilitate the analysis, we make the following assumptions and notations:

A1') There are two classes of stations, each of which contains N_i stations ($1 \le i \le 2$).

A2') A class-i station attempts to transmit a frame in an idle slot with probability p_i in the *p-persistent* version of IEEE 802.11e EDCA ($1 \le i \le 2$).

A3') All packets are of the same size L.

A4') All the stations always have a packet ready for transmission (i.e., the *asymptotic condition* holds).

A5') Instead of waiting DIFS at the end of each busy channel period, a station of class-i waits for $AIFS[i]$ time, where $AIFS[i] = SIFS + aSlotTime \times AIFSN[i]$, before it attempts to access the channel with probability p_i at the beginning of each subsequent idle slot ($1 \le i \le 2$). For ease of notation, we denote $AIFSN[1]$ and $AIFSN[2]$ as A_1 and A_2 respectively. Without loss of generality, we assume $A_1 < A_2$.

Figure 2 depicts an idle period between two channel busy periods. Let I denote the number of idle slots after SIFS time but before any station starts to transmit. The probability mass function of I can be expressed as

$$P[I=j] = \begin{cases} 0, & 0 \le j < A_1, \\ \left[(1-p_1)^{N_1}\right]^{j-A_1} \cdot \left[1-(1-p_1)^{N_1}\right], & A_1 \le j < A_2, \\ \prod_{k=1}^{2} \left[(1-p_k)^{N_k}\right]^{j-A_k} \cdot \left[1-\prod_{k=1}^{2}(1-p_k)^{N_k}\right], & A_2 \le j. \end{cases} \tag{18}$$

The conditional probability that the busy slot that follows the j idle slots incurs a collision (no collision) can be derived as

$$P(\text{collision}|I=j) = 1 - P(\text{success}|I=j)\,., \tag{19}$$

and

$$P(\text{success}|I=j) = \begin{cases} 0, & 0 \le j < A_1, \\ \dfrac{N_1 p_1 (1-p_1)^{N_1-1}}{1-(1-p_1)^{N_1}}, & A_1 \le j < A_2, \\ \dfrac{N_1 p_1 (1-p_1)^{N_1-1}(1-p_2)^{N_2} + N_2 p_2 (1-p_2)^{N_2-1}(1-p_1)^{N_1}}{1-(1-p_1)^{N_1}\cdot(1-p_2)^{N_2}}, & A_2 \le j. \end{cases} \tag{20}$$

The probability that a class-1 or class-2 data frame is successfully transmitted under the condition $I = j$ can be derived as

$$P\left(class_1 \text{ success}|I=j\right)=\begin{cases}0, & 0\le j<A_1,\\ \dfrac{N_1p_1\left(1-p_1\right)^{N_1-1}}{1-\left(1-p_1\right)^{N_1}}, & A_1\le j<A_2,\\ \dfrac{N_1p_1\left(1-p_1\right)^{N_1-1}\left(1-p_2\right)^{N_2}}{1-\left(1-p_1\right)^{N_1}\cdot\left(1-p_2\right)^{N_2}}, & A_2\le j,\end{cases} \tag{21}$$

and

$$P\left(class_2 \text{ success}|I=j\right)=\begin{cases}0, & 0\le j<A_1,\\ 0, & A_1\le j<A_2,\\ \dfrac{N_2p_2\left(1-p_2\right)^{N_2-1}\left(1-p_1\right)^{N_1}}{1-\left(1-p_1\right)^{N_1}\cdot\left(1-p_2\right)^{N_2}}, & A_2\le j.\end{cases} \tag{22}$$

Let the probability that a successfully transmitted data frame is a class-1 or class-2 frame be denoted as $P\left[class_1 \text{ success}\right]$ and $P\left[class_2 \text{ success}\right]$ respectively. They can be derived as follows:

$$\begin{aligned}P\left(class_1 \text{ success}\right) &= \sum_{0}^{\infty}P\left(I=j\right)\cdot P\left(class_1 \text{ success}|I=j\right)\\ &= \sum_{j=A_1}^{A_2-1}\left(\left(1-p_1\right)^{N_1}\right)^{j-A_1}\cdot N_1p_1\left(1-p_1\right)^{N_1-1}\\ &\quad+\sum_{j=A_2}^{\infty}\left(\left(1-p_1\right)^{N_1}\right)^{j-A_1}\cdot\left(\left(1-p_2\right)^{N_2}\right)^{j-A_2}\cdot N_1p_1\left(1-p_1\right)^{N_1-1}\left(1-p_2\right)^{N_2}\\ &= N_1p_1\left(1-p_1\right)^{N_1-1}\cdot\left(1-p_2\right)^{N_2}\\ &\quad\cdot\left(\frac{1-\left(1-p_1\right)^{N_1\left(A_2-A_1\right)}}{1-\left(1-p_1\right)^{N_1}}+\frac{\left(1-p_1\right)^{N_1\left(A_2-A_1\right)}\left(1-p_2\right)^{N_2}}{1-\left(1-p_1\right)^{N_1}\cdot\left(1-p_2\right)^{N_2}}\right),\end{aligned} \tag{23}$$

and

$$\begin{aligned}&P\left(class_2 \text{ success}\right)\\ &= \sum_{j=0}^{\infty}P\left(I=j\right)\cdot P\left(class_2 \text{ success}|I=j\right)\\ &= \sum_{j=A_2}^{\infty}\left(\left(1-p_1\right)^{N_1}\right)^{j-A_1}\cdot\left(\left(1-p_2\right)^{N_2}\right)^{j-A_2}\cdot N_2p_2\left(1-p_2\right)^{N_2-1}\left(1-p_1\right)^{N_1}\\ &= N_2p_2\left(1-p_2\right)^{N_2-1}\cdot\left(1-p_1\right)^{N_1}\cdot\frac{\left(1-p_1\right)^{N_1\left(A_2-A_1\right)}}{1-\left(1-p_1\right)^{N_1}\cdot\left(1-p_2\right)^{N_2}}.\end{aligned} \tag{24}$$

The ratio, $\widehat{r}$, of the average flow throughput of the two traffic classes can be derived as

$$\widehat{r} = \frac{R_1 N_2}{R_2 N_1} = \frac{P\left(class_1 \text{ success}\right) N_2}{P\left(class_2 \text{ success}\right) N_1}$$

$$= \frac{N_1 p_1 \left(1-p_1\right)^{N_1-1} \cdot \left[\frac{1-\left(1-p_1\right)^{N_1(A_2-A_1)}}{1-\left(1-p_1\right)^{N_1}} + \frac{\left(1-p_1\right)^{N_1(A_2-A_1)} \cdot \left(1-p_2\right)^{N_2}}{1-\left(1-p_1\right)^{N_1} \cdot \left(1-p_2\right)^{N_2}}\right] N_2}{N_2 p_2 \left(1-p_2\right)^{N_2-1} \cdot \left(1-p_1\right)^{N_1} \cdot \frac{\left(1-p_1\right)^{N_1(A_2-A_1)}}{1-\left(1-p_1\right)^{N_1} \cdot \left(1-p_2\right)^{N_2}} N_1}$$

$$= \hat{r}_{21} \cdot \left[1 + \frac{\left[1-\left(1-p_1\right)^{N_1} \cdot \left(1-p_2\right)^{N_2}\right] \cdot \left[1-\left(1-p_1\right)^{N_1(A_2-A_1)}\right]}{\left(1-p_2\right)^{N_2} \cdot \left[1-\left(1-p_1\right)^{N_1}\right] \cdot \left[\left(1-p_1\right)^{N_1(A_2-A_1)}\right]}\right], \tag{25}$$

where R_1 and R_2 are the aggregated throughput of class-1 and class-2 traffic respectively, and $\hat{r}_{21} = \frac{p_1\left(1-p_2\right)}{p_2\left(1-p_1\right)}$ is the ratio of the average flow throughput of the two classes in the case that flows of two different classes use the same $AIFS$ value, but different transmission probabilities p_1 and p_2 respectively.

An important implication can be made on Eq. (25) — the ratio of average per flow throughput between different traffic classes is a function of both the transmission probability p_i (or equivalently the contention window size) and the AIFS values $A_i - A_{i-1}$. As the number of priority classes increases, it becomes increasingly difficult to adjust tunable parameters p_i and $A_i - A_{i-1}$ ($1 \leq i \leq P$), in order to simultaneously meet all $\widehat{r}$ constraints. (This will be corroborated by the simulation study in Section VI-A.0.d.) It would actually be desirable to decrease the dimension of design freedom by either (i) setting the AIFS values of all the access categories to the same value (i.e., 2) to enforce $AIFS = DIFS$ or (ii) setting the values of p_i to pre-determined values and determining the appropriate values of A_i to meet the $\widehat{r}$ constraints (Eq. (25)). The former approach is what has been taken in Section III, where it has been shown that throughput differentiation requirements can be met by adjusting p_i while keeping all values of AIFS of all access categories the same. In the latter approach, all the analysis performed in Section III remains unchanged, except that (a) the DIFS term in Eq. (2) is replaced by $SIFS$; and $E(I)$ is calculated using Eqs. (18)–(24). One important observation is that A_1 neither contributes to throughput differentiation nor improves the channel utilization. Therefore it should be set a small value (i.e. 2, which is equivalent to $AIFS_1 = DIFS$).

B. Incorporating Transmission Opportunity (TXOP) in the Analytical Model

In addition to the minimum idle delay before contention (AIFS) and the minimum and maximum contention windows (CW_{min} and CW_{max}), another important parameter that is associated with each access category in 802.11e is the Transmission Opportunity (TXOP). In the legacy DCF protocol, a station can only transmit one *MAC Protocol Data Unit (MPDU)* for each successful contention. In EDCA, a TXOP is defined by a starting time and a maximum duration. After a station successfully contends for the channel, it is allowed to transmit multiple MPDUs as long as the transmission time does not exceed the TXOP limit. As depicted in Fig. 3, within a TXOP, the TXOP holder station continuously transmits the next MPDU, a $SIFS$ time interval after it receives the ACK frame corresponding to the previous MPDU.

To incorporate the effect of TXOP into the derived analytical model, we note that in EDCA with TXOP, only the first MPDU within each TXOP burst may collide with MPDUs from some other stations. That is, as far as the analytical model is concerned, we can envision each TXOP burst as an "extended" frame transmission. Therefore, $E(T_c)$ in Eq. (2) can be calculated with the use of the distribution of the $MPDU$ size, and S in Eq. (2) can be calculated as the TXOP limit minus the residual time at the end of a TXOP (which cannot accommodate one more MPDU and its corresponding ACK and is noted as the *waste* time in Fig. 3).

$$S = DIFS + K \cdot (m_{MPDU} + SIFS + ACK + SIFS) - SIFS, \tag{26}$$

where $K = \lfloor TXOP\_\text{limit}/MPDU \rfloor$ is the number of MPDUs that can be "accommodated" in a TXOP, m_{MPDU} is the transmission duration of a MPDU, and

$$E(T_v) = E\left(\sum_{i=1}^{N_{col}} (DIFS + I_i + T_{c,i} + SIFS + ACK)\right) + E\left(I_{N_{col}+1}\right) + E\left(S\right). \tag{27}$$

Note that under the asymptotic scenario, whether or not a station successfully contends for channel access is independent of whether or not TXOP is used, and hence the derivation of the probabilities of collision and successful transmission is still valid.

Following a similar derivation as in Section IV-A, one can analyze the effect of TXOP limits among different traffic classes on service differentiation. Foreseeable, a similar conclusion may be drawn, i.e., with multiple dimensions of control knobs (CW_{min}, CW_{max}, AIFS, and TXOP), it may become intractable to simultaneously meet all the proportional throughput constraints among different classes. How to fine-tune TXOP policies/parameters under the derived analytical model with the other parameters fixed is a subject of our future study.

V. Incorporating Theoretical Results into IEEE 802.11e EDCA

There are several implementation issues that we must address in order to incorporate the theoretical results derived in Section III into the operations of IEEE 802.11e EDCA. First, we need to devise an approximate solution to calculate the optimal values of p_i with reasonably small computational complexity and error discrepancy (due to approximation). Second, we need to associate the derived optimal values of p_i in the p-persistent version with the contention window based backoff scheme in EDCA. That is, we have to find a mapping between the contention window, cw, and the persistent probability, p. Third, as the optimal values of the system parameters (i.e., p_i or equivalently CW_i) change with the traffic density, we need a simple yet effective mechanism to on-line estimate the number of active stations in a QoS basic service set (QBSS) so as to optimize the system performance in the presence of network dynamics.

A. An Approximate Solution to Obtaining the Optimal Value of p_i

To facilitate the derivation hereafter, we define the following variables

$$A \triangleq \prod_{i=1}^{P}(1-p_i)^{N_i}, \qquad B \triangleq \sum_{i=1}^{P}\left(\frac{N_i \cdot p_i}{1-p_i}\right), \tag{28}$$

$$C \triangleq E(T_c) + DIFS + SIFS + ACK, \tag{29}$$

$$D \triangleq \sum_{i=1}^{P} N_i \widehat{r}_{i1}, \qquad F \triangleq \sum_{i=1}^{P} N_i \widehat{r}_{i1}^2, \qquad T \triangleq t_{slot}. \tag{30}$$

$E\left(N_{col}\right)$ and $E\left(I\right)$ can then be expressed in terms of A and B as

$$E(N_{col}) = \frac{1-A}{A \cdot B} - 1, \tag{31}$$

and

$$E(I) = T \cdot \frac{A}{1-A}. \tag{32}$$

To optimize the channel throughput, we only need to minimize $E\left(T_v\right)$ (Eq. (2)). After omitting the constant items in Eq. (2), the problem of finding the optimal value of p_i that maximizes the channel utilization reduces to one that minimizes

$$\frac{(1-A) \cdot C}{A \cdot B} + \frac{T}{B}. \tag{33}$$

As $p_i = \dfrac{\widehat{r}_{i1} \cdot p_1}{\widehat{r}_{i1} \cdot p_1 + (1-p_1)}$ (Eq. (17)), B can be rewritten as

$$B = \left(\sum_{i=1}^{P} N_i \widehat{r}_{i1}\right) \frac{p_1}{1-p_1} = D \cdot \frac{p_1}{1-p_1}. \tag{34}$$

Thus the problem is equivalent to minimizing

$$\frac{\left(\prod_{i=1}^{P}\left(\frac{\widehat{r}_{i1} \cdot p_1 + (1-p_1)}{(1-p_1)}\right)^{N_i} - 1\right) \cdot C}{\left(\sum_{i=1}^{P} N_i \widehat{r}_{i1}\right) \frac{p_1}{1-p_1}} + \frac{T}{\left(\sum_{i=1}^{P} N_i \widehat{r}_{i1}\right) \frac{p_1}{1-p_1}}. \tag{35}$$

By defining $x = \dfrac{p_1}{1-p_1}$, the problem can be recast as one that finds the optimal value of x $(x \geq 0)$ that minimizes

$$\frac{\left(\prod_{i=1}^{P}(1+\widehat{r}_{i1}x)^{N_i} - 1\right) \cdot C}{x} + \frac{T}{x}. \tag{36}$$

To further simplify Eq. (36), we note that $N_i \widehat{r}_{i1} x \ll 1$. This is based on the observation that under normal, non-contention conditions, even if only class i nodes are active, the probability that at least one station starts to transmit at the beginning of an idle slot is far less than 1 (otherwise contention occurs), and hence $N_i \widehat{r}_{i1} x = N_i \frac{p_i}{1-p_i} \ll 1$. With $N_i \widehat{r}_{i1} x \ll 1$, we can make the following approximation:

$$\begin{aligned} \prod_{i=1}^{P} (1 + \widehat{r}_{i1} x)^{N_i} &\approx 1 + \sum_{i=1}^{P} \widehat{r}_{i1} N_i x + \frac{\left(\sum_{i=1}^{P} \widehat{r}_{i1} N_i\right)^2 - \left(\sum_{i=1}^{P} \widehat{r}_{i1}^2 N_i\right)}{2} \cdot x^2 \\ &= 1 + D \cdot x + \frac{D^2 - F}{2} \cdot x^2, \end{aligned} \tag{37}$$

and hence Eq. (36) can be rewritten as

$$D \cdot C + \left(\frac{D^2 - F}{2}\right) \cdot C \cdot x + \frac{T}{x}. \tag{38}$$

As $D^2 - F > 0$, Eq. (38) is minimized when

$$x = \sqrt{\frac{2T}{(D^2 - F) \cdot C}}. \tag{39}$$

When $x \ll 1$, we have $p_1 = \frac{x}{1+x} \approx x = \sqrt{\frac{2T}{(D^2-F) \cdot C}}$. That is, we use $\sqrt{\frac{2T}{(D^2-F) \cdot C}}$ to approximate the value of p_1 that optimizes the channel utilization.

B. Mapping the *p-persistent* Probability to the Contention Window Size

As discussed in Section II, EDCA in the current 802.11e protocol employs a contention window based backoff mechanism. If we assume the backoff counter value $b(t)$ is randomly chosen in $[0, CW]$, and the contention window CW is fixed (throughout the duration in which a station attempts to transmit a frame), the probability that a station transmits in a slot can be derived (with the use of the same technique in [5]) as

$$p = \frac{2}{cw + 2}. \tag{40}$$

Eq. (40) enables us to apply the analytical results in Section III to the contention-window-based EDCA protocol, by setting CW_{min} and CW_{max} to

$$CW_i^* = \lfloor \frac{2}{p_i^*} - 2.0 \rfloor, \tag{41}$$

where p_i^* is the optimal transmission probability calculated using our analytical model in Section III. That is, with CW^* as the contention window size, the probability that a station transmits in a slot is equal to the optimal transmission probability derived in our analytical model.

C. On-line Measurement of Parameters Needed for the Analytical Model

In the multiple-class, p-persistent analytical model, the number of stations competing for the channel is assumed to be fixed and known *a priori*. In practice, the number of back-logged stations varies, which necessitates estimation of the number of back-logged stations that compete for the wireless medium, so that the optimal transmission probability p_i^* can be determined accordingly. In this subsection, we devise a simple and yet effective measurement mechanism to on-line estimate the system parameters (i.e., the number, N_i, of back-logged stations in each access category) that are required to calculate the optimal values of p_i (and CW_i).

In the case that there exists one priority class, where all stations use the same transmission probability, the average idle period duration between two consecutive transmissions, $E\left(T_{idle}\right)$, can be expressed as (Eq. (14))

$$E\left(T_{idle}\right) = \frac{(1-p)^N}{1-(1-p)^N} \cdot t_{slot}, \tag{42}$$

where p is the transmission probability and N is the number of active stations. When the p-persistent probability is known, one can infer the number of active stations by on-line measuring the idle period.

In the case of multiple priority classes, although the average idle period duration between two consecutive transmissions can still be expressed as a function of the number of stations in each class and the transmission probability of each class, i.e., Eq. (14), it is difficult to use this relation to estimate the number of active stations in each class and to ensure the uniqueness of the solution $\{N_i, 1 \le i \le P\}$.

To deal with the problem, we propose to keep track of the number of active, class-i stations from the channel access history overheard in the past H_i successful transmissions. The value of H_i determines the trade-off between the accuracy and the sensitivity of the online estimation algorithm. The larger the value of H_i, the smaller the probability of missing stations that experience large access delays due to multiple consecutive collisions. On the other hand, a large value of H_i also implies a slower response to station state changes. We set H_i to be the value such that the probability that any given, class-i station successfully transmits at least one data frame is larger than a predefined threshold α (e.g., $\alpha = 0.9$). As the probability that a given station successfully transmits in a virtual transmission time can be calculated as

$$\begin{aligned} P_s(i) &= \frac{p_i(1-p_i)^{N_i-1}\prod_{j\neq i}^{P}(1-p_j)^{N_j}}{\sum_{i=1}^{P} N_i p_i(1-p_i)^{N_i-1}\prod_{j\neq i}^{P}(1-p_j)^{N_j}} \\ &= \frac{p_i}{1-p_i}\cdot\left(\sum_{j=1}^{P}\frac{N_j \cdot p_j}{1-p_j}\right)^{-1}, \end{aligned} \tag{43}$$

we can set the value of H_i to be the largest integer k such that

$$\sum_{j=1}^{k}\left(P_s\left(i\right)\right)^j\left(1-P_s\left(i\right)\right)^{k-j} > \alpha. \tag{44}$$

Eq. (44) ensures that in the estimation period, the probability that any given active, class-i station successfully transmits at least one data frame is larger than α.

D. Complete Procedures for Realizing Service Differentiation in IEEE 802.11e EDCA

By incorporating the above (i) on-line algorithm for estimating the number of active, class-i stations and (ii) mapping between the optimal p-persistent probability and the contention window size, EDCA will be able to support (proportional) service differentiation. The complete procedures that the AP and each of the mobile stations take are described in Fig. 4. Note that in the procedures, rather than having both $CW_{i,min}$ and $CW_{i,max}$ set to CW_i^* (as suggested in Section V-B), only $CW_{i,min}$ is set to the optimal value calculated according to the analytical model. That is, both the AP and mobile stations still carry out the exponential binary backoff algorithm for medium access control, except that $CW_{i,min} = CW_i^*$. This enhances the robustness of the protocol under some abnormal scenarios, such as the temporary interference and deviation of the estimated number of active stations from its actual value. Under these cases, frame collision still occurs and the exponential binary backoff algorithm will take effect to mitigate collision. On the other hand, under normal cases, collision does not occur frequently when the parameters are updated with quasi-optimal values, and hence the performance discrepancy between these two choices should not be notable.

VI. Performance Evaluation

To validate the correctness of our analytic model and to evaluate the performance of IEEE 802.11e EDCA with the analytical results incorporated, we have implemented the *p-persistent* version of 802.11e EDCA in *ns-2*. We have also implemented an experimental prototype of the enhanced EDCA mechanism (that includes the service differentiation algorithm described in Section V-D) on a Linux-based MADWifi driver for wireless LAN devices with the Atheros chipset. In what follows, we report our simulation and empirical results.

A. Simulation Study

In the simulation study, the network topology includes N_i class-i mobile stations, each of which has a CBR traffic source that generates packets either (i) at a rate high enough to emulate the asymptotic condition or (ii) in compliance with the on-off model to emulate bursty traffic. All stations send CBR packets of sizes 500 bytes to the base station. We do not consider TCP traffic, because we are primarily interested in the performance of IEEE 802.11 EDCA and do not intend to model the interaction of TCP with EDCA. (The reason for not modeling the interaction is in part due to the difficulty and complexity involved in characterizing the intrinsic characteristics of TCP in addition to MAC protocols themselves). In reality it is also unlikely QoS sensitive traffic will be transported through TCP.

The values of system parameters used in the simulation conform to the 802.11b standard, and are listed in Table II. The optimal class-i transmission probability, p_1^*, is calculated using numerical methods (Section III-B) or using the approximate solution (Section V-A), with the objective of maximizing the channel utilization (subject to the throughput ratio constraint). The optimal transmission probabilities, p_i^* ($i \geq 2$), are calculated using Eq.(17).

We calculate the throughput in terms of the received payload of MAC data frames, and carefully consider all the overhead introduced in the physical and MAC headers in both the theoretic analysis and the simulation. We assume that all the stations can hear each other, and hence do not consider the hidden terminal/exposed terminal problem. We present only simulation results in the 2-priority class case. Unless specified otherwise, the targeted ratio, $\hat{r}^{21}$, of the throughput attained by a class 1 station and that by a class 2 station is set to 2.0. Simulations results under other network configurations can be found in [14].

a) Validation of the Analytic Model: In this set of simulation, we validate the analytic model in Section III with respect to its capability of providing throughput differentiation and achieving the maximum channel utilization. For the purpose of comparison, in addition to using the optimal p_1 and p_2 values, we also use 18 sets of transmission probabilities to calculate the channel utilization, where p_1 is selected from $\{p_1^* \times 0.1 \times i | 1 \leq i \leq 9\}$ and $\{p_1^* \times (1.0 + 0.5 * i) \, | 1 \leq i \leq 10\}$ and p_2 is calculated using Eq.(17).

Figures 5 depict the total system throughput, the throughput attained by each class, and the per flow throughput ratio in the case of $N_1 = 14$ and $N_2 = 7$. Several observations are in order: First, the simulation results are in extremely good agreement with those obtained in the analytic model. Second, when the transmission probabilities deviate from the optimal value, the total system throughput decreases accordingly, as predicted in the analytical model. Also, the throughput ratio between the two classes is very close to the specified value, indicating that QoS provisioning through appropriate setting of backoff values is feasible. This is also in line with our analysis.

b) Validation of the Approximate Solution: To validate whether or not the approximate solution proposed in Section V-A (Eq. (39)) renders acceptable results, we conduct simulation to compare the throughput results with the optimal and approximate values of p_1 respectively, by varying the values of $\widehat{r}_{12}$, the sizes of data frames, and the number of stations in each class. Again, due to the page limit, we present only one set of simulation results in Table III where the packet size is set to 500Bytes.

As given in Table III (a), the system throughput obtained with the approximate values of p comes surprisingly close to that with the optimal value of p. To better understand why such the approximate solution renders such good results, we list in Table III (b) the value of C (Eq. (29)) which corresponds to the fixed portion of $E(T_v)$. In fact, C is equal to the value of $E(T_v)$ assuming perfect scheduling algorithm is employed. As both the optimal value and the approximate value of p_1 result in very low frame collision rates, in the vicinity of the optimal network operation state, the frame collision rate is quite small and the overhead incurred in frame collision and binary backoff only counts for a small portion of $E(t_v)$. Even if the approximate value of p_1 cannot achieve the minimal overhead duration (i.e., $E(N_{col}) \cdot C + (E(N_{col}) + 1) \cdot E(I)$), the aggregated channel throughput is still close to its maximal achievable value.

c) Performance of IEEE 802.11e EDCA with Analytical Results Incorporated: To evaluate the performance of IEEE 802.11e EDCA with the analytical results incorporated (Section V-D), we have carried out simulation under various scenarios.

Single Class, Greedy Traffic: In this set of simulations, we evaluate the performance of IEEE 802.11e EDCA with the on-line parameter estimation mechanism incorporated (Fig. 4) in the case of a single traffic class.

There are a total of 10 mobile nodes and one access point. Each mobile node generates data packets with a rate high enough to saturate the channel (i.e., the asymptotic scenario). The average data packet size is 500 bytes. We activate one station every five seconds, and 30 seconds after all stations are activated, we start to deactivate one station every five seconds. We keep track of the number of active nodes (both actual and estimated), the calculated transmission probability, p_1, and the system throughput every 2 seconds and depict them in Fig. 6. The simulation results show that the number of active nodes estimation algorithm works quite well and the system throughput is kept high due to the fact that the transmission probability is adaptively calculated to mitigate collision.

Two Classes, Greedy Traffic: In this set of simulations, we evaluate the performance of IEEE 802.11e EDCA with the analytical results incorporated in the case of two traffic classes. The ratio of per flow rate $\hat{r}_{12}$ is set to 2.0. A total of 20 mobile nodes exist in the WLAN, with 10 nodes in each class. Each node generates data packets with a rate high enough to saturate the channel (i.e., the asymptotic scenario). We activate one station in each class every 10 seconds, and 50 seconds after all stations are activated, we start to deactivate one station in each class every 10 seconds. As shown in Fig. 7, for most of the time, the ratio of per-flow attainable throughput between the two classes is very close to $\hat{r}_{12}$, i.e., the system capacity is distributed among different traffic classes in compliance with the throughput requirement. Moreover, the channel utilization is kept high regardless of the change in the number of active nodes.

Two Classes, On-Off Traffic: In this set of simulations, we study the performance of the analytically enhanced version of IEEE 802.11e EDCA in the case of bursty traffic. The network configuration is similar to that in the previous set of simulations, except that each mobile node generates on-off traffic. The duration of the on and off periods are both exponentially distributed with mean time 5000ms. The data packet generation rate in the on period is set to 500K bps. The data packet size is 500 bytes. As shown in Fig. 8, the ratio of per-flow attainable throughput between the two classes is kept, for most of the time, at the desired value (i.e., 2.0), although the per flow throughput ratio is not strictly achieved when the channel offered load is low (in the first one third and the last one third simulation period). The failure to keeping the desired throughput differentiation during any time interval in the case of on-off traffic is, in part, due to the fact that the analytical result is derived under the fluid model assumption, which does not hold in the case of bursty traffic. The same phenomenon has also been observed and discussed in [16].

d) Effect of AIFS on Service Differentiation: As mentioned in Section IV-A, the ratio of average per flow throughput between different traffic classes is a function of both the transmission probability p_i (or equivalently the contention window size CW_i) and the AIFS values $A_i - A_{i-1}$ (Eq. (25)), and it becomes difficult to simultaneously tune both sets of parameters in order to meet all the throughput ratio constraints, while maximizing the channel utilization. To validate the above statement, we carry out simulation with different AIFSN values.

In this set of simulations, two groups of mobile stations exist in a BSS, each of which contains m and n

mobile stations. Each mobile station generates CBR traffic at a high rate so that the asymptotic condition holds. The AIFSN value is fixed at 2 for one group, and varies from 2 to 8 from one simulation run to another to for another group. For each combination of m and n, the value of p_1 and p_2 are set to that calculated in the analytic model in Section III, with the targeted per flow throughput ratio, $\hat{r}_{21}$, set to 2.0 and $AIFS[i] = DIFS$. Figure 9 depicts the ratios of average per flow throughput between the two classes obtained via simulation and the analytical model. As shown in Figure 9, in the worst case, the ratio of average per flow throughput is larger than the targeted ratio by 3 times. This implies that without appropriate admission control, low priority flows may be starved by high priority flows.

B. Empirical Study

We have leveraged the Linux-based MADWifi (Multiband Atheros Driver for WiFi) driver for wireless LAN devices with the Atheros chipset, and implemented much of the functionality of the enhanced EDCA mechanism. The major reason we chose this chipset is that it fulfills most of the criteria necessary to implement the proposed change. A majority of other drivers, including those developed for Intel and Prism chipsets, require a specific firmware. As the firmware implements much of the device functionality, such as enforcing radio regulations, allowing the device to act as an access point, and handling IEEE 802.11 management [25], the use of firmware typically restricts any modifications to operating parameters.

The Atheros hardware, on the other hand, does not require loading of firmware, but instead relies on a *Hardware Access Layer (HAL)* module that is provided in a binary-only form. The HAL module operates between the hardware and driver to manage much of the chip-specific operations and to enforce the required FCC regulations. The HAL is similar to firmware in that it ensures that users do not set invalid operating parameters, but implements less functionality than other firmware and actually provides an interface that allows changes of various device parameters, including the minimum and maximum contention windows. The only restriction that HAL enforces on the contention windows is that their values must be set to $2^x - 1$, where $1 \leq x \leq 11$. Therefore, the contention window value calculated from Eq. (41) in Section V-B must be approximated. Another advantage of the Atheros chipset is that, because the chipset is basic, most of the MAC functionality is handled in the driver, as opposed to the firmware. Therefore, the IEEE 802.11 MAC protocol, including the state machine and protocol support, can be easily modified to support the enhanced EDCA mechanism.

e) How to support floating point operations: Apart from several low-level implementation details (which the interested reader is referred to [11]), there are two major implementation issues that arises in the course of prototype implementation. First, floating point operations (such as *sqrt*) are required in the enhanced EDCA mechanism, but the kernel drivers do not contain floating point operation routines. To deal with this problem, we have divided the EDCA implementation into kernel and user-space components such that all floating-point operations are performed in the user-space. The practice of splitting the implementation between kernel and user-space is quite common in Linux.

Given that the prototype must be divided between the kernel and user-space, the two components must be able to communicate with each other. As described in Section V-D, each station has to estimate the number of active stations in each class, and send the estimated result to the user-space. The user-space component will then calculate the new contention window for each class, and instrument the HAL in the kernel-space to set the parameters accordingly. Linux provides various methods for interprocess communication between kernel and user-space components, such as system calls, ioctl calls, or netlink sockets. As system calls and ioctl calls do not allow the kernel to initiate communication with the user-space. for the user-space component to remain synchronized with the kernel-space component, it must continually poll the kernel, which has been tested (in our experiment) to be inefficient. Instead, we leverage the netlink socket facility, as it provides a full-duplex, bi-directional link between the kernel and user-space components, thereby allowing the kernel to initiate communication with the user-space component whenever necessary.

f) How to include additional information required by EDCA with consideration of backward compatibility: The second implementation issue is that, to support on-line computation of optimal contention window sizes, each station must know the number of active stations in each class in the system. Although new fields can be introduced into the IEEE 802.11 MAC header of data and management frames, we have decided not to do so, to ensure as few code changes as possible and backward compatibility with stations that do not employ the enhanced EDCA mechanism.

Instead, we will place the needed information in the body of a beacon frame. As defined in [28], the body of a beacon frame consists of fixed fields, which are mandatory and fixed-length, and information elements, which are variable-length and may be mandatory or optional. Information elements are defined to have a common general format consisting of a 1 octet Element ID field, a 1 octet length field, and a variable-length element-specific information field, whose length is specified in the length field. We decide that the Information element is ideal for placing the additional information because it can support a variable number of service classes, and a majority of the element ids are not being used. Also, it is legitimate to include optional information elements in a beacon frame body, and if the MAC protocol does not support an information element, it is simply ignored.

g) Empirical Results: The network topology used for the empirical study consists of two mobile stations and one AP that were within four feet of each other. Each station runs Fedora Core 2 with the Linux 2.6.9 kernel. Each station had a CBR traffic source that generates 500-byte UDP packets and send the packets to the AP at a rate high enough to keep its system buffer full. The stations starts transmitting packets immediately after they associate with the AP. Table IV summarizes the relevant parameters used by the Atheros driver.

For each experiment, the total system throughput, the throughput attained by each station, and the throughput ratio of the two stations are shown. In the course of collecting statistics, we ignore the first few seconds in each experiment because each station may not always have a packet to send while the traffic source attempts to fill the station's system buffer to capacity (i.e., the asymptotic condition may not hold). Unless otherwise stated, each set of results is the average of 20 runs of the experiments, where each run lasts 100 seconds and each station updates its traffic classes every 0.5 second. Although a wide variety of scenarios have been tested, due to the space limit, we report below two sets of representative results.

Performance in the presence of two traffic classes with constant traffic sources: In this set of experiments, both the class-1 and class-2 stations are active during the entire duration of the experiment. Fig. 10 shows the throughput results when $\widehat{r}_2 = 4$. The total system utilization was kept high and steady during the duration of the experiment, and the throughput ratio between the two traffic classes was fairly close to the specified value.

Performance in the presence of two traffic classes with on-off traffic sources: In this set of experiments, only the class-2 station is active during the entire experiment. The class-1 station sends packets in an on-off manner, with the duration of its on and off periods being set to ~ 20 sec.

Fig. 11 shows the throughput results when $\widehat{r}_{12} = 4$. Note that when the class-1 station is inactive, the bandwidth is allocated to the class-2 station. The throughput ratio is kept reasonably close to 4, and the total channel utilization remains fairly high during the entire experiment, regardless of the changes in the number of active stations. Note, however, that there is a slight decrease in the total channel throughput when the class-1 station is inactive. This is because the class-1 station is assigned a CW value of 3 when both stations are active, and the class-2 station is assigned a CW value of 7. As a result, during the inactive periods, even though the class-2 station has no other station to contend with, it cannot achieve as high a throughput because of its longer backoff time.

h) Possible sources of error: Although the above results show that the enhanced EDCA mechanism performs reasonably well under various scenarios, the throughput ratio between the two classes could have been closer to the specified QoS. The error is, in part, attributed to the fact that the HAL module of the Atheros driver places restrictions on the value of CW_{min}: the value calculated by Eq. (41) must be rounded to the closest $2^x - 1$ value, where $1 \leq x \leq 11$. Another possible source of error is that these experiments were not performed in a closed environment. As a result, nearby stations and APs also contended for channel access. In our experiments, each station received, on average, approximately 40-50 beacon packets per second from nearby APs.

VII. Future Work and Conclusion

In this paper, we have derived, based on the analytical model developed in [8], a multi-class model to study how to adaptively tune parameters in IEEE 802.11e EDCA and support service differentiation in WLANs. Through analytical modeling, we demonstrate that by assigning appropriate different transmission probabilities (or different contention window sizes) to stations of different classes, it is feasible to provide (proportional) service differentiation and achieve pre-specified targeted throughput ratios among different classes, while at the same time maximizing the total system capacity. We then extend the derived theoretical model to incorporate the role of AIFS and TXOP values on service differentiation perceived by different traffic class. We show that it may not be desirable to allow tuning of multiple parameters (e.g., both the contention window sizes and the AIFS values). Instead, the design dimension should be decreased by allowing turning of only one set of parameters, while keeping the other two sets of parameters for all the access categories fixed at the same value (i.e., setting the AIFS values of all access categories to 2, which is equivalent to $AIFS = DIFS$).

To practically incorporate the analytical model for on-line parameter tuning, we derive an approximate solution for the model to reduce the computational complexity. We also propose a simple and yet effective method to on-line estimate the number of active stations in a QoS basic service set so as to deal with network dynamics and optimize the system parameters accordingly. Through *ns-2* simulation and experimentation on a systems prototype, we show that with the analytical results and the on-line parameter tuning mechanism incorporated, IEEE 802.11e EDCA can indeed achieve (proportional) QoS.

We have identified several research avenues for future work. Although we believe the dimension of design freedom should be kept small in order to ensure (deterministic) QoS guarantees, we are currently exploring parameter tuning in the other dimensions and investigating how to tune the values of AIFS and TXOP for service differentiation, based on the models derived in Section IV. Also, we recognize that QoS support can not be solely realized in the MAC layer, but rather involves close interaction across multiple layers spanning from the application layer (e.g., the coding scheme selected) to the physical layer (e.g., the channel transmission rate selected). As such, we will attempt to model the interaction between multiple layers in terms of QoS provisioning, and devise accordingly possible signaling mechanisms between multiple layers. Our ultimate objective is to demonstrate that the application layer can be provided with better QoS by the underlying QoS-aware MAC protocols.

ACKNOWLEDGMENT

The authors would like to thank Dennis Chi for implementing the enhanced EDCA mechanism in the MADWifi driver and carrying out the experiments. The authors would also like to express their gratitude toward the anonymous reviewers whose constructive comments have greatly help improving the quality of the manuscript.

REFERENCES

[1] I. Aad and C. Castelluccia. Differentiation mechanisms for IEEE 802.11. In *Proc. IEEE INFOCOM'01*, April 2001.

[2] A. Banchs and X. Perez. Distributed weighted fair queueing in 802.11 wireless LAN. In *Proc. of IEEE International Conference on Communications*, volume 5, pages 3121–3127, April 2002.

[3] Giuseppe Bianchi. IEEE 802.11–saturation throughput analysis. *IEEE Comm. Letters*, 2(12):318–320, 1998.

[4] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J-SAC*, 18(3), Marth 2000.

[5] Giuseppe Bianchi, Luigi Fratta, and Matteo Oliveri. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In *Proceedings of IEEE PIMRC'96*, volume 2, pages 392–396, October 1996.

[6] Luciano Bononi, Marco Conti, and Lorenzo Donatiello. Design and performance evaluation of a distributed contention control (DCC) mechanism for IEEE 802.11 wireless local area networks. *Journal of Parallel and Distributed Computing*, 60(4):407–430, April 2000.

[7] Luciano Bononi, Marco Conti, and E. Gregori. Run-time optimization of IEEE 802.11 wireless lans performance. *IEEE Transactions on Parallel and Distributed Systems*, 15(1):66–80, January 2004.

[8] F. Cali, M. Conti, and E. Gregori. IEEE 802.11 Wireless LAN: Capacity analysis and protocol enhancement. In *Proc. IEEE INFOCOM'98*, March 1998.

[9] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Trans. on Networking*, 8(6):785–799, December 2000.

[10] F. Cali, M. Conti, and E. Gregori. IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism. *IEEE JSAC*, 18(9):1774–1786, September 2000.

[11] Dennis Chi. Design and implementation of the generic wireless device driver layer to support qos provisioning. Master's thesis, University of Illinois at Urbana Champaign, August 2005.

[12] Sunghyun Choi, Javier del Prado, Sai Shankar N, and Stefan Mangold. IEEE 802.11e contention-based channel access (EDCF) performance evaluation. In *Proceedings of IEEE ICC'03*, 2003.

[13] J. Deng and R.S. Chang. A priority scheme for IEEE 802.11 DCF access method. *IEICE Trans, Commun.*, E82-B(1):96–102, 1999.

[14] Ye Ge. *QoS Provisioning for IEEE 802.11 MAC Protocols*. PhD thesis, The Ohio State University, 2004.

[15] IEEE 802.11 wireless local area networks. http://grouper.ieee.org/groups/802/11/.

[16] Zhimei Jiang, Li-Fung Chang, and N. K. Shankaranarayanan:. Providing multiple service classes for bursty data traffic in cellular networks. In *Proceedings of IEEE INFOCOM'00*, volume 3, pages 1087–1096, 2000.

[17] B. Li and R. Battiti. Performance analysis of an enhanced IEEE 802.11 distributed coordination function supporting service differentiation. *Quality for all*, pages 152–161, 2003. QofIS 2003, LNCS 2811.

[18] Anders Lindgren, Andreas Almquist, and Olov Schelen. Evaluation of Quality of Service schemes for IEEE 802.11 wireless LANs. In *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, pages 348–351, November 2001.

[19] Anders Lindgren, Andreas Almquist, and Olov Schelen. Quality of Service schemes for IEEE 802.11 wireless LANs - an evaluation. *In Special Issue of the Journal on Special Topics in Mobile Networking and Applications (MONET) on Performance Evaluation of Qos Architectures in Mobile Networks*, 8(3):223–235, June 2003.

[20] Stefan Mangold, Sunghyun Choi, Peter May, and Guido Hiertz. IEEE 802.11e - fair resource sharing between overlapping basic service sets. In *Proc. of IEEE PIMRC'02*, 2002.

[21] Dennis Pong and Tim Moors. Call admission control for IEEE 802.11 contention access mechanism. In *Proceedings of IEEE GLOBECOM'03*, pages 173–178, December 2003.

[22] Daji Qiao and Kang Shin. Achieving efficient channel utilization and weighted fairness for data communications in ieee 802.11 WLAN under the DCF. In *Proceedings of the Tenth International Workshop on Quality of Service*, 2002.

[23] Lamia Romdhani, Qiang Ni, and Thierry Turletti. Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks. In *Proc. of IEEE WCNC'03*, 2003.

[24] The Institue of Electrical and Electronics Engineerings, Inc. IEEE standard 802.11: Wireless LAN media access control (MAC) and physical layer (PHY) specifications, 1999.

[25] J. Tourrilhes. Linxu wireless lan howto. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireles s.pdf, January 2005.

[26] N. H. Vaidya, P. Bahl, and S. Gupa. Distributed fair scheduling in a wireless LAN. In *Proceeding of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 167–178, August 2000.

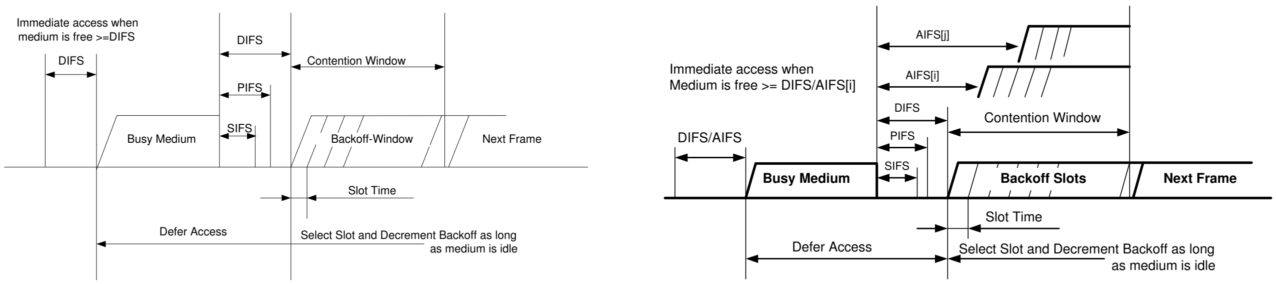
[27] A. Veres, A. T. Campbell, M. Barry, and L-H. Sun. Supporting service differentiation in wireless packet networks using distributed control. *IEEE Journal of Selected Areas in Communications, Special Issue on Mobility and Resource Management in Next-Generation Wireless Systems*, 19(10):2094–2104, October 2001.

[28] IEEE 802.11 WG. Reference number ISO/IEC 8802-11:1999(E) IEEE std 802.11, 1999 edition. internation standard [for] Information Technology-Telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements- part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) specifications, 1999.

[29] Yang Xiao. An analysis for differentiated service in ieee 802.11 and IEEE 802.11e wireless LANs. In *Proceedings of IEEE ICDCS'04*, pages 32–39, Match 2004.

[30] Yang Xiao, Haizhou Li, and Sunghyun Choi. Protection and guaranteee for voice and video traffic in IEEE 802.11e wireless LANs. In *Proceedings of IEEE INFOCOM'04*, 2004.

[31] Hao Zhu, Allen D. Mathias, and Guohong Cao. EDCF-DM: A novel enhanced distributed coordination function for wireless ad hoc networks. In *Proceedings of IEEE ICC'04*, June 2004.



(a) Basic access in DCF

(b) Basic access in EDCA

Fig. 1

ACCESS METHODS IN IEEE 802.11 DCF AND IEEE 802.11E EDCA.

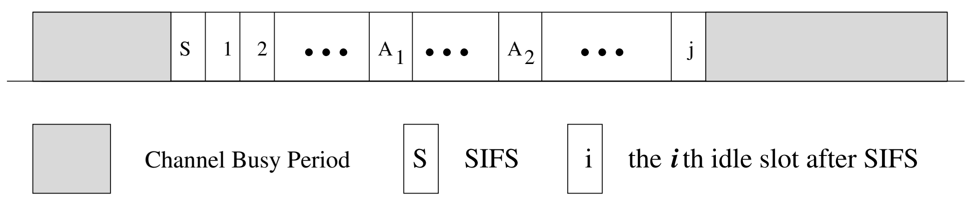


Fig. 2

THE IDLE PERIOD BETWEEN TWO BUSY PERIODS.

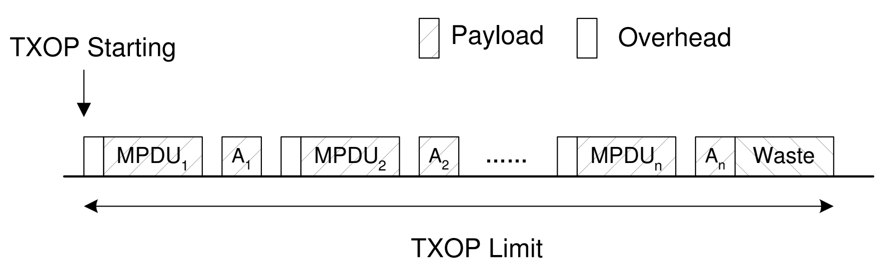


Fig. 3

HOW THE TRANSMISSION OPPORTUNITY (TXOP) GOVERNS CHANNEL ACCESS.

AC	aCWmin	aCWmax	AIFSN
AC_BK	31	1023	7
AC_BE	31	1023	3
AC_VI	15	31	2
AC_VO	7	15	2

TABLE I

DEFAULT EDCA PARAMETER SET.

Given service differentiation requirement $\{r_{ij}\}$

I. Procedure running on the access point (AP):

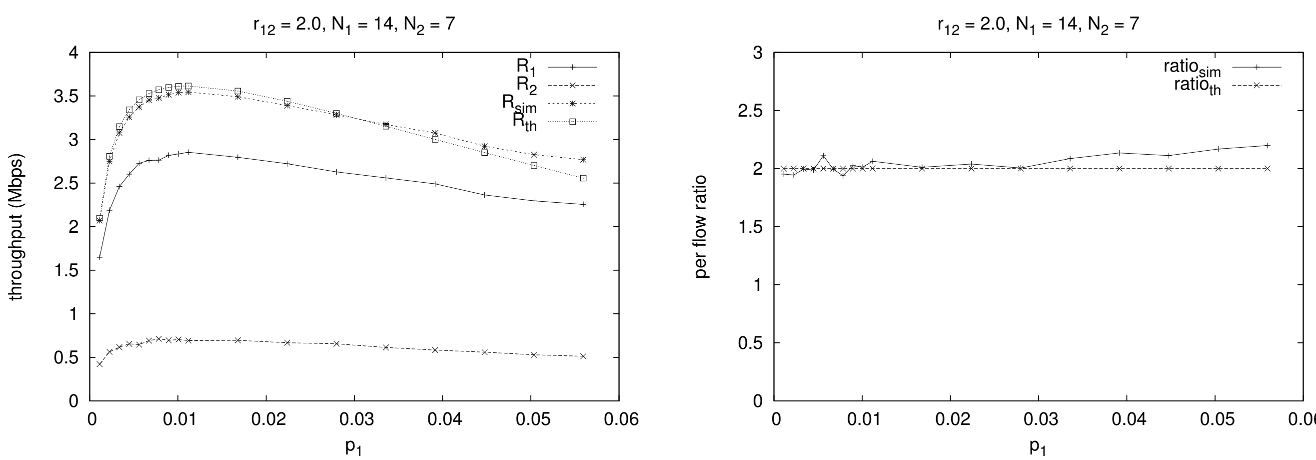
Step 0. Initialize the number of active stations in each class, N_i to the total number of stations associated with the AP in this QBSS.
Step 1. Calculate p_i^{opt} according to the analytical model.
Step 2. Calculate the contention window size cw_i^{opt} according to Eq. (41).
Step 3. In the beacon frame, set $CW_{i,min}$ to the newly calculated CW_i^* value, and the $AIFSN$ values for all the classes to 2. Other parameters can be set to the default parameters in the IEEE 802.11e Draft standard. For example, CW_{max} can be set to $(CW_{min} + 1) \cdot 2^5 - 1$. Broadcast the beacon frame.
Step 4. Update the estimation period H_i according to Eq. (44), and calculate H, where $H = \max_j H_j$.
Step 5. Record up to H sender ids and class ids of successfully received or transmitted data frames.
Step 6. Estimate the number of active stations N_i in each class by counting the number of unique, active class-i stations in the transmission history recorded in Step 6. Go to Step 1.

II. Procedure running on each mobile station:

Step 1. Listen to the beacon messages and update the MAC parameters (i.e., $CW_{i,min}$, $CW_{i,max}$, etc.) carried in the beacon messages.
Step 2. Transmit frames according to the exponential binary backoff algorithm as specified in EDCA.

Fig. 4

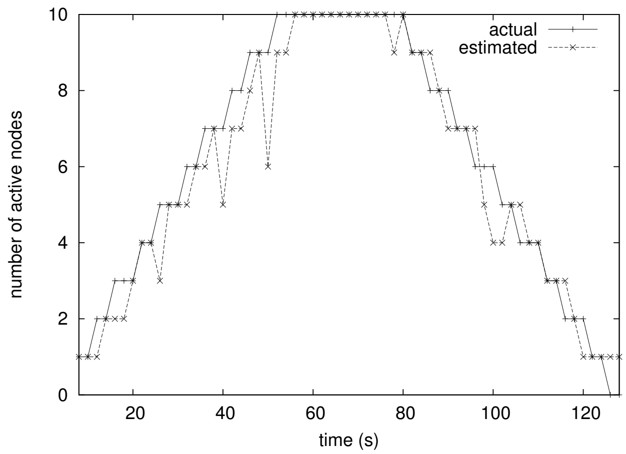
DETAILED PROCEDURES TO REALIZE (PROPORTIONAL) SERVICE DIFFERENTIATION IN IEEE 802.11E EDCA.



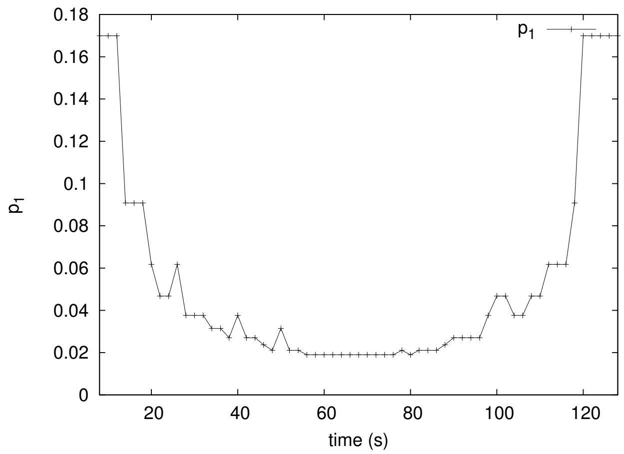
(a) Throughput (b) Throughput ratio

Fig. 5

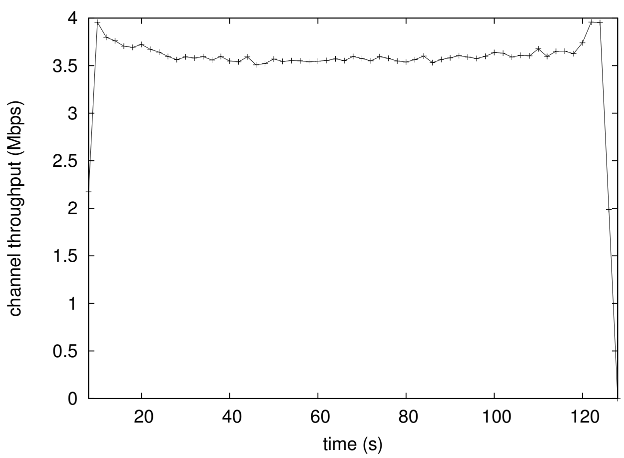
THE THROUGHPUT AND THROUGHPUT RATIO VERSUS THE TRANSMISSION PROBABILITY, p_1, OF CLASS-1 STATIONS. $N_1 = 14$, $N_2 = 7$, AND r_{12}=2.0.



(a) Number of active nodes



(b) Transmission probability



(c) System throughput

Fig. 6

SIMULATION RESULTS IN A WIRELESS LAN WITH 10 MOBILE NODES AND GREEDY TRAFFIC GENERATORS.

t_{slot}	20 μs
SIFS	10 μs
DIFS	50 μs
Data Rate	11 Mbps
PLCPDataRate	1 Mbps
PreambleLengthBits	144 bits
PLCPHeaderLength	48 bits
ACKLengthBytes	14 bytes
MacHeaderLengthBytes	28 bytes

TABLE II

PARAMETERS USED IN THE SIMULATION.

$\widehat{r}_{12}$	n	p_1	R	$p_{1approx}$	R_{approx}
2	1	0.171008	3.74086	0.206284	3.72878
2	2	0.0724368	3.61077	0.0809113	3.60662
2	5	0.0268989	3.54636	0.029173	3.54412
2	10	0.0131568	3.5265	0.014151	3.5247
2	20	0.00651062	3.51684	0.00697368	3.51523
2	30	0.00432398	3.51365	0.00462714	3.5121
2	40	0.00323937	3.51207	0.0034622	3.51055
2	50	0.00258883	3.51112	0.00276587	3.50962
4	1	0.225843	3.81065	0.29173	3.78687
4	2	0.0897676	3.63667	0.101567	3.6315
4	5	0.0326288	3.55545	0.0355082	3.55302
4	10	0.0158719	3.53088	0.0170942	3.529
4	20	0.00783225	3.51899	0.00839532	3.51734
4	30	0.00519728	3.51508	0.00556434	3.51351
4	40	0.00389065	3.51313	0.00416121	3.5116
4	50	0.00310671	3.51197	0.00332322	3.51045

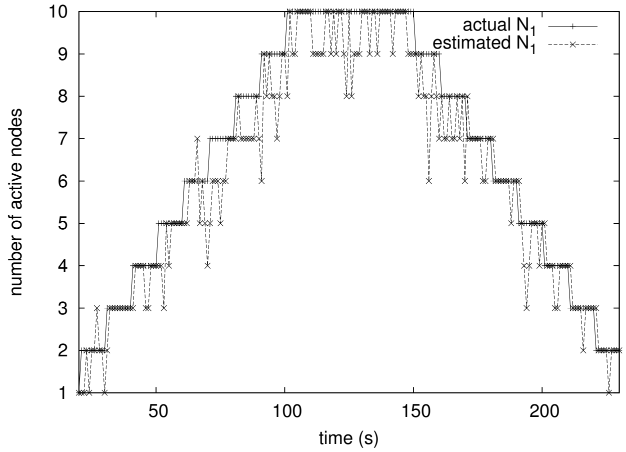
(a) System throughput

$\widehat{r}_{12}$	n	$E(Tv)$	$E(Tv)_{approx}$	C
2	1	0.00106927	0.00107274	0.00094
2	2	0.0011078	0.00110907	0.00094
2	5	0.00112792	0.00112863	0.00094
2	10	0.00113427	0.00113485	0.00094
2	20	0.00113739	0.00113791	0.00094
2	30	0.00113842	0.00113892	0.00094
2	40	0.00113893	0.00113942	0.00094
2	50	0.00113924	0.00113973	0.00094
4	1	0.00104969	0.00105628	0.00094
4	2	0.00109991	0.00110147	0.00094
4	5	0.00112503	0.0011258	0.00094
4	10	0.00113286	0.00113346	0.00094
4	20	0.00113669	0.00113722	0.00094
4	30	0.00113795	0.00113846	0.00094
4	40	0.00113858	0.00113908	0.00094
4	50	0.00113896	0.00113945	0.00094

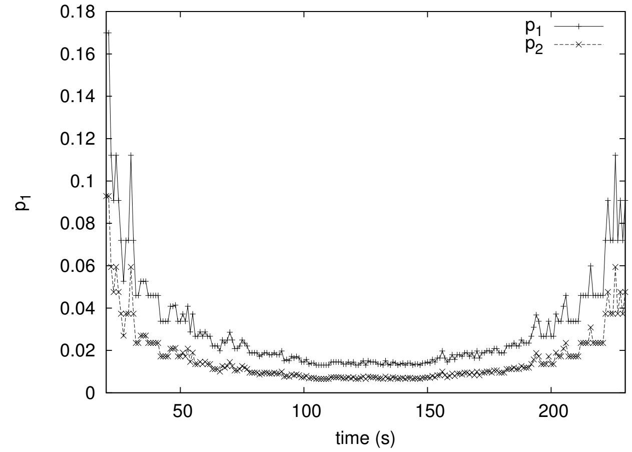
(b) Virtual transmission period

TABLE III

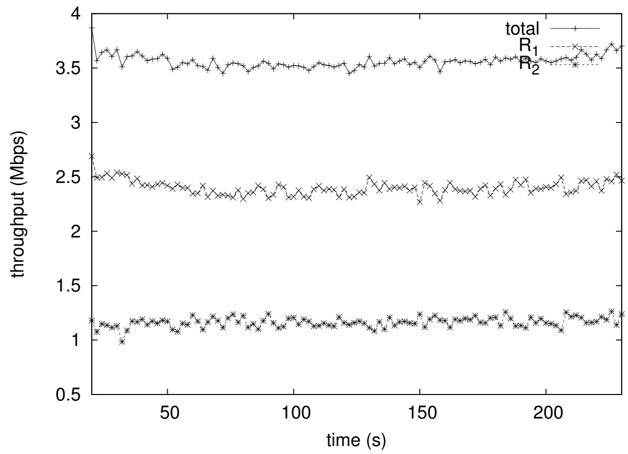
SYSTEM THROUGHPUT AND VIRTUAL TRANSMISSION PERIOD WITH THE USE OF THE OPTIMAL AND THE APPROXIMATE VALUES OF p_1 RESPECTIVELY. PACKET SIZE = 500BYTES.



(a) Number of active nodes



(b) Transmission probability



(c) System throughput

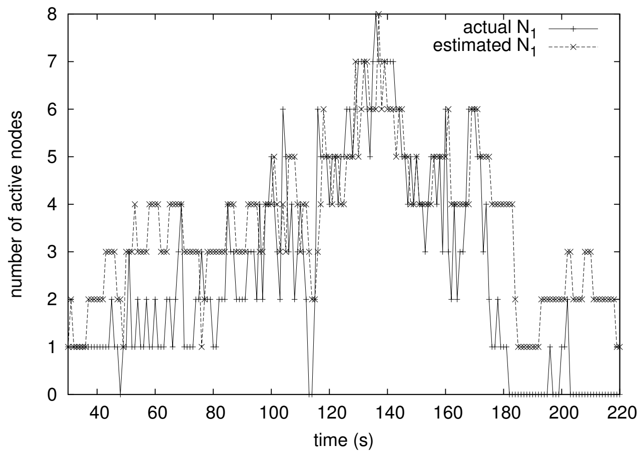
Fig. 7

SIMULATION RESULTS IN A WIRELESS LAN WITH 10 MOBILE NODES IN EACH CLASS ($N_1 = 10, N_2 = 10$) AND GREEDY TRAFFIC GENERATORS.

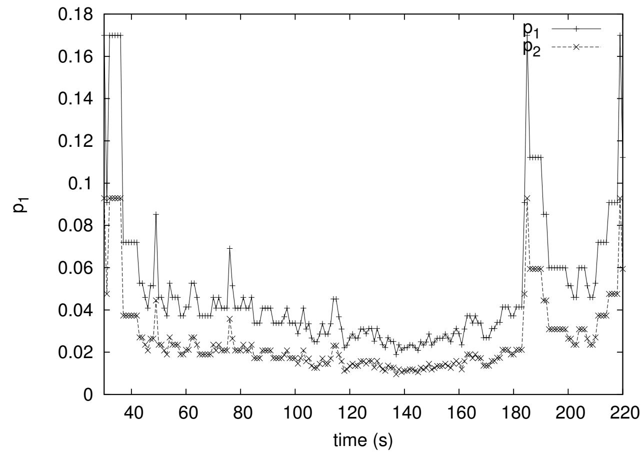
t_{slot}	20 μs
$SIFS$	10 μs
$DIFS$	50 μs
PLCP Data Rate	1 Mbps
Preamble Length	18 bytes
PLCP Header Length	6 bytes
Data Rate	11 Mbps
MAC Header Length	28 bytes
ACK Length	14 bytes

TABLE IV

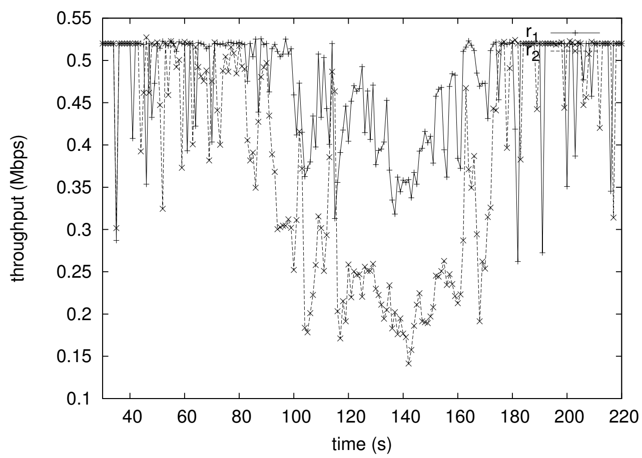
RELEVANT PARAMETERS USED BY THE ATHEROS DRIVER.



(a) Number of active nodes



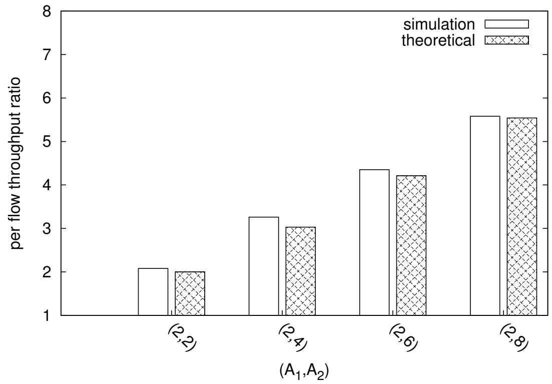
(b) Transmission probability



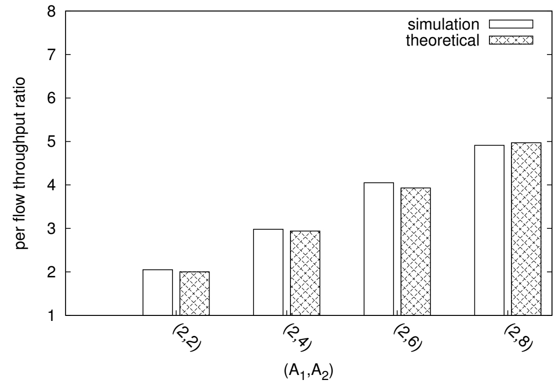
(c) Per-flow attainable throughput

Fig. 8

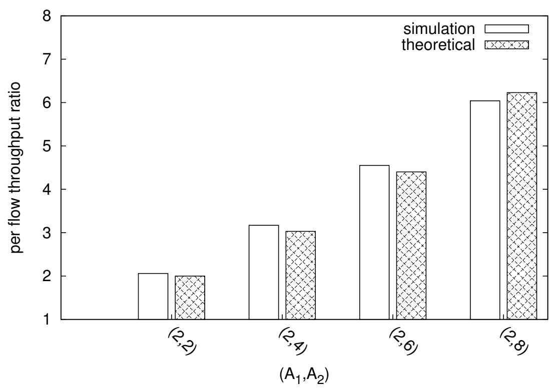
SIMULATION RESULTS IN A WIRELESS LAN WITH 10 MOBILE NODES IN EACH CLASS AND ON-OFF TRAFFIC GENERATORS.



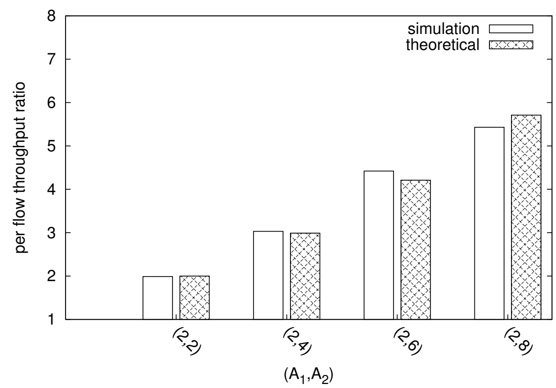
(a) $m = 1, n = 5$



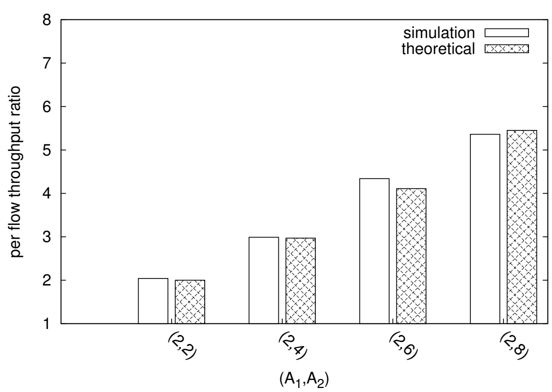
(b) $m = 1, n = 15$



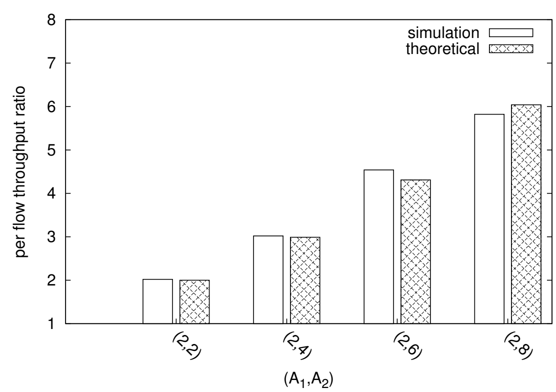
(c) $m = 5, n = 5$



(d) $m = 5, n = 10$



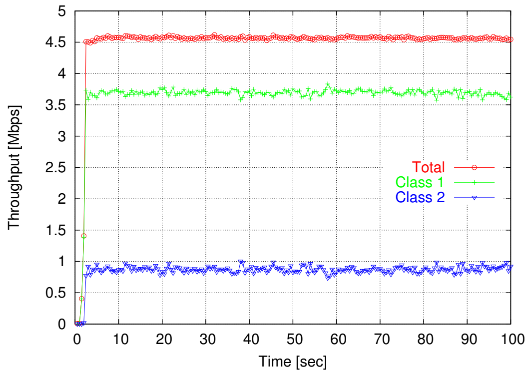
(e) $m = 5, n = 15$



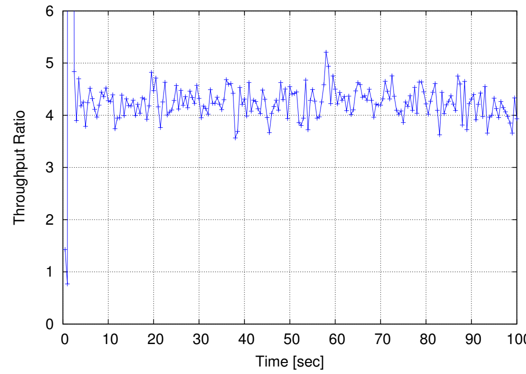
(f) $m = 10, n = 10$

Fig. 9

RATIO OF AVERAGE PER FLOW THROUGHPUT FOR DIFFERENT COMBINATIONS OF m AND n. THE x-AXIS GIVES (A_1, A_2).



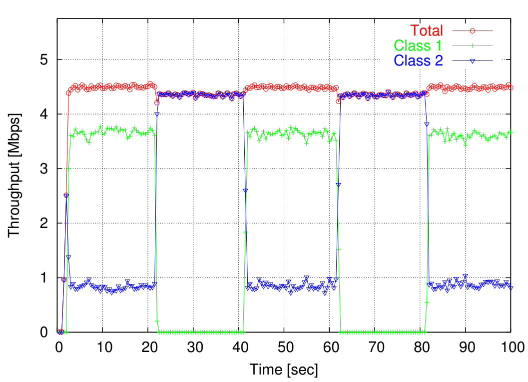
(a) Throughput



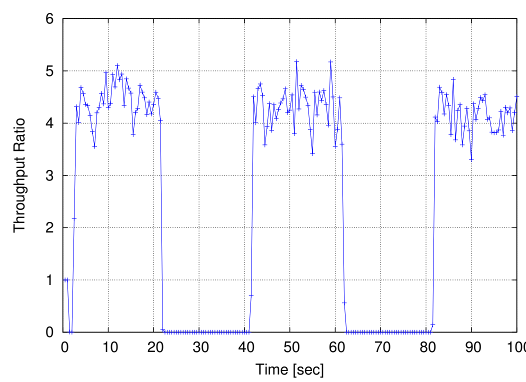
(b) Throughput ratio

Fig. 10

THROUGHPUT ATTAINED BY TWO TRAFFIC CLASSES WITH CONSTANT TRAFFIC SOURCES. $\widehat{r}_{12} = 4$



(a) Throughput



(b) Throughput ratio

Fig. 11

THROUGHPUT ATTAINED BY TWO TRAFFIC CLASSES WITH ON-OFF TRAFFIC SOURCE. $\widehat{r}_{12} = 4$