Provisioning Quality Controlled Medium Access in UltraWideBand (UWB) WPANs

Chunyu Hu†, Hwangnam Kim‡, Jennifer C. Hou‡, Dennis Chi‡
† Department of Electrical and Computer Engineering
‡ Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801 USA
E-mail:{*chunyuhu, hkim27, jhou, dychi*}*@uiuc.edu*

Sai Shankar N*
Staff Engineer
Qualcomm Standards Engineering Dept.
5775 Morehouse Drive
San Diego, CA, 92121 USA
E-mail: *nsai@qualcomm.com*

***Abstract*— Quality of service (QoS) provisioning is one of the most important criteria in newly emerging UWB-operated WPANs, as they are expected to support a wide variety of applications from time-constrained, multimedia streaming to throughput-hungry, content transfer applications. As such, the Enhanced Distributed Coordinated Access (EDCA) mechanism has been adopted by MultiBand OFDM Alliance in its UWB MAC proposal. In this paper, we conduct a rigorous, comprehensive, theoretical analysis and show that with the currently recommended parameter setting, EDCA cannot provide adequate QoS. In particular, without responding to the system dynamics (e.g., taking into account of the number of active class-i stations), EDCA cannot allocate bandwidth in a deterministic proportional manner and the system bandwidth is under-utilized.**

After identifying the deficiency of EDCA, we propose, in compliance with the EDCA-incorporated UWB MAC protocol proposed in [18] [23] [24], a framework, along with a set of theoretically grounded methods for controlling medium access with deterministic QoS for UWB networks. We show that in this framework, 1) real-time traffic is guaranteed of deterministic bandwidth via a *contention-based* reservation access method; 2) best-effort traffic is provided with deterministic proportional QoS; and moreover, 3) the bandwidth utilization is maximized. We have also validated and evaluated the QoS provisioning capability and practicality of the proposed MAC framework both via simulation and empirically by leveraging the MADWifi (Multiband Atheros Driver for WiFi) Linux driver for Wireless LAN devices with the Atheros chipset.

I. Introduction

With the maturity of several enabling techniques (ranging from communication theory to semiconductor technologies), ultra wide band (UWB) networks have become a promising candidate for Wireless Personal Area Networks (WPANs), capable of transmitting packets at a high rate (480+ Mbps) for short-range communication (up to 20 meters). The MultiBand OFDM Alliance (MBOA), established in 2003 and merged with the WiMedia Alliance in 2004, represents a leading force with 170+ participating members. Efforts are put forth to standardize both PHY and MAC specifications.

*: The work was done when the author was with Philips Research USA.

Technical Report No. UIUCDCS-R-2005-2600 (Engr. No. UILU-ENG-2005-1795), July 2005

To support peer-to-peer mobile applications, it is necessary for the network to support mobility and allow devices to move in/out of a piconet without significant performance degradation. Therefore, a distributed MAC architecture with loose coordination is preferred to a centralized design. This, combined with several other design concerns, rules out the IEEE 802.15.3 MAC specification [1], which is TDMA-based and requires a central Piconet Coordinator. A new MAC protocol has to be defined to meet the WPAN requirements as well as to utilize the high data rates as afforded by UWB.

One of the vital requirements for WPANs is quality of service (QoS) and efficient bandwidth utilization. As a WPAN is envisioned to support a vast number of applications varying in type, nature and/or user preference, QoS provisioning has to be a *built-in* function in the PHY/MAC specification. There have been many research efforts, largely focusing on designing and evaluating the PHY layer performance [2] and [12] and PHY-related issues in the MAC layer [19], [16], [24]. In particular, Merz *et al.* [19] addressed the problem of dynamically selecting appropriate power and link rates. Lu *et al.* [16] addressed the time acquisition issue. A MAC protocol specification for UWB currently being developed by MBOA [18] [23] [24] has proposed to leverage an *Enhanced Distributed Coordinated Access* (*EDCA*) mechanism for QoS provisioning. The main intent of this paper is to investigate, through analytic modeling, simulation, and experimentation, how deterministic QoS provisioning can be realized in such an EDCA-incorporated UWB protocol.

EDCA is a fully distributed service differentiation mechanism originally defined in IEEE 802.11e. Note that IEEE 802.11e is proposed as a supplement to IEEE 802.11 MAC Distributed Coordination Function (DCF), and aims to support QoS for IEEE 802.11. For a detailed description of IEEE 802.11 and 802.11e MAC, refer to [11] and [3]. In EDCA, medium access is contention-based and prioritized by two configurable parameters: the contention window size (*CW*) and the arbitration inter frame space (*AIFS*). The contention window size determines the number of backoff slots (which is uniformly distributed in $[0, CW - 1]$) a station has to count

down before a transmission attempt can be made. The AIFS value determines the number of slots that has to be sensed idle before the backoff procedure is initialized/resumed. See Fig. 1 for an illustration. Flows of different priorities are assigned different parameter values to increase/decrease their chance of gaining medium access. Although this is intuitively correct, it is important to understand quantitatively how, and to what extent, the two parameters favor/disfavor data transmission from high-priority/low-priority flows.

Several studies on evaluating the performance of EDCA have been made *via simulation* in the context of IEEE 802.11e in [7], [14], [15], [17] and [20]. Several theoretical models that shed insights on how service differentiation can be achieved have been reported in [8], [10], [13], [21], [22], [26], and [27], again in the context of IEEE 802.11e. Most of the models, if not all, are based on Bianchi's model [4] or Calí's model [5] that were proposed to study the performance of IEEE 802.11 DCF under the asymptotic condition (i.e., all the stations always have packets ready for transmission). Among all the models, those reported in [13] and [26] analyze the effect of varying the contention window size on the performance of service differentiation, and those reported in [8], [10], [21], [22], and [27] also study the effect of varying AIFS values on the performance. In most (if not all) of the work that studys the AIFS effect, such as [10], [21], [22], and [27] a different contention window range $[1, CW]$ (rather than $[0, CW-1]$) is assumed. As has been pointed out in [9], this subtle difference results in considerable degradation in the system throughput. In summary, a model that conforms to the standard and fully incorporates both parameters is yet to observe. A joint study based on that on how, and to what extend, the two parameters affect the performance is expected.

In this paper, we propose a comprehensive, accurate, and yet simple model to characterize data activities in EDCA and jointly study the effect of varying both the contention window size and AIFS values. Instead of using a p-persistent model (that is commonly assumed in literature), we use a discrete-time Markov chain to describe the transition of the channel state. In particular, the model has revealed a common pitfall used in the p-persistent model (i.e., the probability of accessing *any* slot is p) and its adverse effect on the model accuracy. With the results derived in the proposed model (and corroborated in the simulation study with the use of UWB-specified parameters), we have made several important observations that will help in the design of the EDCA-incorporated UWB: 1) Traffic classes with small values of AIFS dominate channel access, depriving traffic classes with larger AIFS values of their channel access. 2) With the currently proposed parameter setting, the differentiation mechanism fails to allocate bandwidth among stations of different classes at deterministic QoS. That is, given the currently proposed parameter setting, the EDCA-incorporated UWB MAC protocol does not fully support both real-time data streams and best-effort applications at their configured QoS. 3) The available bandwidth is under-utilized given the proposed parameter setting.

Guided by the derived model, we then devise, in compliance with the EDCA-incorporated UWB MAC protocol (and in particular, the superframe structure) proposed in [18] [23] [24], a framework, along with a set of theoretically grounded methods for controlling medium access with deterministic QoS for UWB networks. In particular, we derive the optimal contention window sizes that give deterministic proportional QoS for different traffic classes. In this enhanced UWB MAC protocol, 1) real-time traffic is guaranteed of deterministic bandwidth via a *contention-based* reservation access method; 2) best-effort traffic is provided with deterministic proportional QoS; and moreover, 3) the bandwidth utilization is maximized. The performance of the enhanced UWB MAC protocol is evaluated via both analytical derivation, simulation studies, and empirical experiments (leveraging the MADWifi (Multiband Atheros Driver for WiFi) Linux driver for Wireless LAN devices with the Atheros chipset).

The rest of the paper is organized as follows. After introducing the network model and stating the assumptions, we present in Section II the analytical model that characterizes the EDCA service differentiation mechanism. In Section II, we carry out simulation to validate the model via simulation. The validation process also reveals the deficiency of the current parameter settings used in the UWB MAC protocol. In Section IV, we devise a framework for controlling medium access with deterministic QoS for UWB networks. We evaluate the performance of the enhanced UWB MAC protocol both via simulation and empirically in Sections V–VI, and conclude the paper in Section VII.

II. An Analytical Model for EDCA

A. Network Model and Assumptions

We envision a general single-cell UWB-operated wireless network without any central coordinator. All stations can hear each other, i.e., there exists no hidden terminal. We assume that no capture technique is used and that every station is backlogged and always has packet(s) to send (i.e., the asymptotic condition commonly used to analyze the saturation performance holds).

There are M priority classes, with the number of stations in each class being N_j, $j = 1, \cdots, M$. (We will address how to on-line determine N_j in the dynamic case later.) Each class is configured with a set of QoS parameters for distributed access contention: the inter-frame space $AIFS_j$ ($= SIFS + AIFSN_j \times aSlotTime$ and the contention window size CW_j. Without loss of generality, we assume $AIFSN_1 \leq AIFSN_2 \leq \cdots \leq AIFSN_M$. In particular, let m denote the priority index such that $AIFSN_1 = \cdots = AIFSN_m \neq AIFSN_{m+1} \leq AIFSN_{m+2} \ldots$.

All stations share and access the channel with the use of the enhanced distributed coordinated access (EDCA) mechanism. That is, a random backoff interval value is uniformly chosen in $[0, \widehat{CW_i} - 1]$ and used to initialize the backoff timer, where $\widehat{CW_i}$ is the current contention window for traffic class i. The

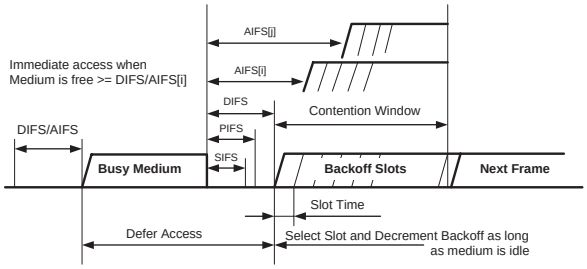


Fig. 1. The relation between different inter-frame spaces.

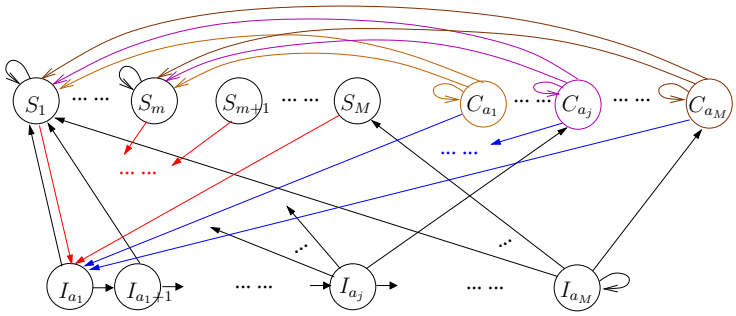


Fig. 2. The discrete Markov chain that describes the channel state transition

backoff timer is decreased as long as the channel is sensed idle, stopped when data transmission (initiated by other stations) is in progress, and reactivated when the channel is sensed idle again for more than AIFS[i], where i denotes the traffic class. The time immediately following an idle period of length *short inter-frame space* (SIFS) is slotted, with each slot equal to the time needed for any station to detect the transmission of a packet from any other station. When the backoff timer expires, the station attempts for frame transmission at the beginning of the next slot time. Finally, if the data frame is successfully received, the receiver transmits an acknowledgment frame after a SIFS, [1] If an acknowledgment is not received, the data frame is presumed to be lost, and a retransmission is scheduled. The value of $\widehat{CW_i}$ is set to $CW_{min,i}$ in the first transmission attempt, and is doubled at each retransmission up to a pre-determined value $CW_{max,i}$.

Let $a_j \triangleq AIFSN_j + 1$. For ease of explanation, we divide the slots subsequent to a busy period into consecutive *contention zones*. Specifically, as illustrated in Fig. 3, *contention zone* j ($j = 1, \cdots, M-1$) starts at the a_j-th slot and ends at (including) the ($a_{j+1}-1$)-th slot. The *M-th contention zone* includes the a_M-th slot and all the slots beyond. An important observation that will be used throughout the derivation is — under EDCA, *only stations of the first j classes are eligible to transmit in the j-th contention zone.* Also, as the a_1-th slot is the first slot immediately following a busy period and (as will be shown later) plays an important role in the performance, we term it as the *post-busy slot*.

B. Channel State Space

For ease of explanation, we treat a collision period or a successful transmission as a virtual (busy) slot. The channel state, denoted by $s(t)$, is sampled at the end of each busy/idle slot. There are three types of possible channel states:

1) State I_k ($k = a_1, a_1+1, \cdots, a_M-1$): the idle channel state in the k-th slots after the busy slot. (Recall that the first slot is the one immediately after a busy slot plus a SIFS.) In other words, when the channel is in I_k, it means there are $k-1$ consecutive idle slots subsequent to a busy slot and the k-th slot is still idle. In addition, I_{a_M} is the channel state in which there are $\geq a_M$ consecutive idle slots.
2) State S_j: a successful transmission made by a station of class j.
3) State C_{a_j}: either a collision subsequent to an idle slot that is in the j-th contention zone or a collision subsequent to such a collision. By the definition of the contention zone, collision C_{a_j} only involves stations of the first j classes.

The channel state space is thus defined as $\mathbb{S} = \{ I_k,\ S_j,\ C_{a_j} : j = 1, \cdots, M \text{ and } a_1 \leq k \leq a_M \}$.

[1]The necessity of returning an acknowledgment is due to that typically WLAN devices are equipped with half-duplex radios and therefore are unable to simultaneously transmit and receive.

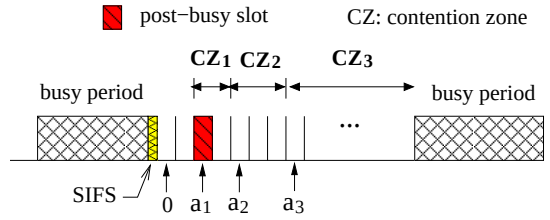


Fig. 3. An illustration of the notions of contention zones and the post-busy slot. In this example, there are three classes with different AIFS values. $a_j = AIFSN_j + 1$.

C. Transitions of Channel States

We use a discrete-time Markov chain (Fig. 2) to describe the transition of channel states. Possible transitions are:

1) $I_k \rightarrow \{ I_{k+1}, S_1, \cdots, S_j, C_{a_j} \}$, for $a_j \leq k \leq a_{j+1} - 1$, $j = 1, \cdots, M-1$: An idle slot state can transition to another idle slot state, a collision state, or a successful transmission state. Specifically, an idle slot state in the j-th contention zone can transition to a successful transmission (made by one of the stations of the first j classes), or to a collision C_{a_j} (caused by two or more stations of the first j classes attempting for transmission), or to the next idle slot in the time order.
2) $I_{a_M} \rightarrow \{ I_{a_M}, S_1, \cdots, S_M, C_{a_M} \}$: The only exception in the transition from an idle slot state occurs when the

idle slot is in the M-th contention zone. By the definition of the M-th contention zone, an idle slot state in the M-th contention zone can transition to itself. Hence, as shown in Figure 2, instead of having an outgoing arrow into other idle slot states, state I_{a_M} contains a self-loop pointing to itself.

3) $S_j \rightarrow \{\ S_j,\ I_{a_1}\ \}$, for $j = 1, \cdots, m$: As a station will transmit immediately once its backoff timer counts down to zero, when the backoff timer freezes upon detection of the busy medium, the timer value is *always* positive and has at least 1 slot time. Therefore, stations that did not participate in transmission in the busy period will *not* transmit in the post-busy slot (i.e., the a_1-th slot immediately after a busy period). This peculiar access behavior to the *post-busy* slot is often ignored in previous work, which assumes uniform and independent access to *any slot* (*a.k.a.* p-persistent). Because of this behavior, subsequent to a successful transmission, either another successful transmission made by the *same* station follows, or the channel becomes idle. This implies, for $j \leq m$, state S_j can either transition to itself or the idle slot state I_{a_1}.
4) $S_j \rightarrow \{\ I_{a_1}\ \}$, for $j = m+1, \cdots, M$: As stations of classes $j > m$ are assigned larger AIFS values, they are not eligible to access the *post-busy* slot. Hence, states S_j, $j > m$, will transit exclusively to the idle slot state I_{a_1}.
5) $C_{a_j} \rightarrow \{\ C_{a_j},\ I_{a_1},\ S_1, \cdots,\ S_m\ \}$, for $j = 1, \cdots, M$: Transitions from collision states can be similarly explained as in cases 3–4 and hence are not elaborated on here.

D. Derivation of State Transition Probabilities

Now we are in a position to derive the transition probabilities of all the possible transitions. We assume that stations of class j access a slot other than an *post-busy* slot *independently* and *uniformly* with probability τ_j. τ_j is termed as the *attempt probability*. For ease of exposition, we first consider the case that the contention window size CW_j is fixed. Then we extend the model to accommodate that case that the contention window size changes in compliance with the the binary exponential backoff procedure.

Before we proceed, we define the following terms for notational convenience:

$$A_j \triangleq \prod_{h=1}^{j}(1-\tau_h)^{N_h}, \quad B_j \triangleq \prod_{h=1}^{j}\left(1-\frac{\tau_h}{CW_h}\right)^{N_h}. \tag{1}$$

a) Transitions from the idle slot state to other states: Recall that in the j-th contention zone, only the first j classes are eligible to contend for channel access. Hence, we derive, for each contention zone, transition probabilities from an idle channel state. For notational convenience, we define $a_{M+1} = a_M + 1$.

For $k \in [a_j, a_{j+1})$, $j = 1, 2, \cdots, M$, and $u \leq j$,

$$P[I_k \rightarrow I_{k+1}] = A_j, \tag{2}$$

$$P[I_k \rightarrow S_u] = N_u\tau_u(1-\tau_u)^{N_u-1}\prod_{h=1,h\neq u}^{j}(1-\tau_h)^{N_h} = \frac{N_u\tau_u}{1-\tau_u}A_j \tag{3}$$

$$P[I_k \rightarrow C_{a_j}] = 1 - P[I_k \rightarrow I_{k+1}] - \sum_{u=1}^{j}P[I_k \rightarrow S_u] \tag{4}$$

b) Transitions from a successful transmission state to other states: After a station of the first m classes finishes a successful transmission, it may gain the channel access again if it chooses 0 as the next backoff timer value. This occurs with probability $\frac{1}{CW_j}$, since the backoff timer value is selected uniformly in $[0, CW_j - 1]$. On the other hand, if the station is of of class $m+1$ to M, it is not eligible to access the *post-busy* slot. Moreover, all the other stations have frozen their backoff timer with the remaining timer value at least 1 slot, and hence will not attempt to transmit either. In this case, the *post-busy* slot is idle with probability 1. We have

$$P[S_j \rightarrow S_j] = \frac{1}{CW_j}\ \text{, for } j = 1, \cdots, m, \tag{5}$$

$$P[S_j \rightarrow I_{a_1}] = \begin{cases} 1 - \frac{1}{CW_j}\ \text{, for } j = 1, \cdots, m, \\ 1\ \text{, for } j = m+1, \cdots, M. \end{cases} \tag{6}$$

c) Transitions from a collision state to other states: Recall that by the definition of m, $AIFSN_1 = \cdots = AIFSN_m$, i.e., the first m contention zones is practically the same one. Consequently, we merge the collision states C_{a_k}, $k = 1, \cdots, m$, into one state, and denote it by C_{a_m}. The transition probabilities originating from C_{a_j}, for $j = m, \cdots, M$ and $u \leq m$ are given below, with their detailed derivation given in Lemma 1–2 in Appendix I.

For $j = m, \cdots, M$ and $u \leq m$,

$$P[C_{a_j} \rightarrow I_{a_1}] = \frac{1}{P[I_k \rightarrow S_u]}\left\{B_m - A_j\left[1 + \sum_{k=1}^{m}\frac{N_k\tau_k}{1-\tau_k}\times\left(1-\frac{1}{CW_k}\right) + \sum_{k=m+1}^{j}\frac{N_k\tau_k}{1-\tau_k}\right]\right\}, \tag{7}$$

$$P[C_{a_j} \rightarrow S_u] = \frac{1}{P[I_k \rightarrow S_u]}N_u\frac{\tau_u}{CW_u}\left(\frac{B_m}{1-\frac{\tau_u}{CW_u}} - \frac{A_j}{\tau_u}\right), \tag{8}$$

$$P[C_{a_j} \rightarrow C_{a_j}] = 1 - P[C_{a_j} \rightarrow I_{a_1}] - \sum_{u=1}^{m}P[C_{a_j} \rightarrow S_u]. \tag{9}$$

With all the derived transition probabilities, we can compute the stationary probabilities of channel states by solving the equilibrium equations for the Markov chain, $\mathbf{s} = \mathbf{s}\mathbb{P}$. Let the equilibrium channel state be denoted by $\widetilde{s}$.

E. Derivation of the Attempt Probability

Recall that each station attempts to transmit in a slot (other than the *post-busy* slot) independently and uniformly with

probability τ_j, where j is the priority class which the station belongs to.

Given a fixed contention window size, CW_j, for each class j, τ_j can be simply expressed as $\frac{2}{CW_j}$ – the inverse of the average waiting (backoff) time. Note that the backoff timer is frozen when data transmission (initiated by other stations) is in progress, and resumed when the channel is sensed idle again for more than AIFS[i]. The expression results from that the backoff time at the beginning of any *eligible* slot is approximately uniformly distributed in $[0, CW_j - 2]$, where an *eligible* slot refers to any non-*post-busy* slot.

An iterative algorithm to derive $\overline{CW_j}$ and τ_j: In the case that the contention window size CW_j changes in compliance with the binary exponential backoff procedure, we develop an iterative algorithm to derive the average contention window size $\overline{CW_j}$.

Consider the view of a tagged station of class j. Let $p_{coll}(j)$ denote the probability that when the tagged station transmits a frame in an eligible slot, the frame incurs a collision; and $p_{coll}^{(r)}(j)$ denotes this probability in the r-th iteration. The average contention window size, $\overline{CW_j}^{(r)}$, in the r-th iteration can be computed from its probability mass function:

$$\begin{aligned} q(j,\ell) &\triangleq P\left[CW_j^{(r)} = W(j,\ell)\right] \\ &= p_0 \left(p_{coll}^{(r)}(j)\right)^{\ell} \text{ for } \ell = 0, \ldots, L_j, \end{aligned} \tag{10}$$

where p_0 is the normalization factor, and can be obtained by noting $\sum_{\ell=0}^{L_j} q(j,\ell) = 1$. L_j is the retry limit for class j and $W(j,\ell) = \min\{2^{\ell} CW_{min}(j), CW_{max}(j)\}$, $\ell = 0, \cdots, L_j$. The attempt probability for the next iteration, $\tau_j^{(r+1)}$, can be computed from $\overline{CW_j}^{(r)}$ by $\tau_j^{(r+1)} = \frac{2}{\overline{CW_j}^{(r)}}$.

The probability $p_{coll}(j)$ is yet to be derived. As this probability varies in different contention zones, we will flirtly derive the conditional probability of collision given that the system is in the contention zone k ($j \le k \le M$), and then the probability that the system is in the contention zone k. The former probability can be expressed as $p_{coll}(j,k) = 1 - \frac{A_k}{1-\tau_j}$, i.e., the probability that at least one other station of the first k classes transmits in the same slot. The latter probability that the system is in the contention zone k is $\sum_{h=a_k}^{a_{k+1}-1} \mathbb{P}[\widetilde{s} = I_h]$. Now $p_{coll}(j)$ can be expressed as

$$p_{coll}(j) = \frac{1}{c_0} \sum_{k=j}^{M+1} \left(1 - \frac{A_k}{1-\tau_j}\right) \left(\sum_{h=a_k}^{a_{k+1}-1} \mathbb{P}[\widetilde{s} = I_h]\right), \tag{11}$$

where $c_0 = \sum_{h=a_k}^{a_M} \mathbb{P}[\widetilde{s} = I_h]$ and recall $a_{M+1} \triangleq a_M + 1$. Note that all the stationary probabilities used here should be derived from the perspective of of the tagged station, i.e., the number of stations in the class which the tagged station belongs to is reduced by 1 in all the relevant calculation.

The average attempt probabilities calculated in the iterative algorithm will replace all the τ_j terms in Eqs. (5)-(9). Since after a successful transmission, a station will reset its contention window size to the minimum value, CW_j in Eqs. (2)-(9) will be replaced by $CW_{min}(j)$ after a successful transmission.

F. Derivation of the System Throughput

We compute the system throughput by calculating the average amount of successful transmission (in bits) over the expected length of a slot (By a *slot*, we mean either a successful transmission, a collision, or an idle slot). Specifically, let t_s denote the length of an idle slot (which is a PHY parameter), and T_D and T_C, respectively, the average length of a successful transmission and the average length of a collision period. With the results derived in Sections II-C and II-E, the expected slot time can be expressed as

$$\overline{t} = t_s \sum_{k=a_1}^{a_M} \mathbb{P}[\widetilde{s} = I_k] + T_D \sum_{j=1}^{M} \mathbb{P}[\widetilde{s} = S_j] + T_C \sum_{j=1}^{M} \mathbb{P}[\widetilde{s} = C_{a_j}]. \tag{12}$$

In the basic distributed access mechanism, i.e.,, without the RTS-CTS floor acquisition mechanism, a successful transmission contains transmission of a DATA frame and a SIFS followed by an ACK. The second term results from that after each successful transmission, the backoff timer of a station is resumed only after an idle period of AIFS, and for ease of computation, we consider $AIFS_{min}$ ($\triangleq \min\{AIFSN_j,\ j = 1, \cdots, M\}$) as part of a successful transmission, i.e., $T_D = DATA + SIFS + ACK + AIFS_{min}$, where $DATA$ is the transmission time of a data frame.

A collision is detected by a sender station upon the timeout of the sender timer, and by other stations when they receive corrupted packets. After detecting a collision, a receiver node resumes its backoff timer after an idle period of EIFS, where EIFS is set to $EIFS = SIFS + ACK + AIFS_{min}$, so that both colliding and non-colliding stations resume their backoff timers or start to sense the channel at approximately the same time. This gives $T_C = DATA_{max} + SIFS + ACK + AIFS_{min}$, where $DATA_{max}$ is the largest DATA frame incurred in the collision. In the case that the RTS-CTS mechanism is used, T_D and T_C can be derived in a similar manner.

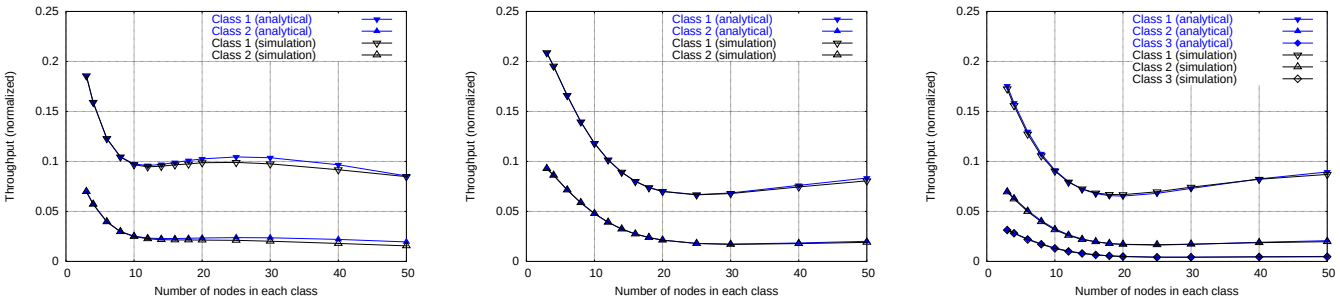
The throughput of stations of class η_j ($j = 1, \cdots, M$) can be expressed as

$$\eta_j = \frac{\overline{m} \times \mathbb{P}[\widetilde{s} = S_j]}{\overline{t}}, \tag{13}$$

where $\overline{m}$ is the average payload (in bits) carried in a DATA frame. Note that the underlying assumption in Eq. (13) is that the size of data frames of all classes has the same distribution. It is, however, straightforward to extend Eq. (13) to accommodate the case that the size of data frames of different classes has its individual distribution.

III. Deficiency of Current UWB Parameter Settings

We have performed a simulation study to both validate the analytic model derived in Section II and to evaluate the performance of EDCA (in conjunction with the current



(a) 2 classes: $CW_{1,2}$=[8,8], [16,16] (b) 2 classes: $CW_{1,2}$=[16,16], [32,32] (c) 3 classes: $CW_{1,2,3}$=[8,16], [16,32], [32,64]

Fig. 4. Analytical and simulation results for multiple classes (each with a different contention window size but the same AIFS value ($AIFSN = 2$)).

parameter setting as suggested in MBOA MAC [18], [23]). In particular, we will study both the individual and combined impacts of the *contention window size* and the *AIFS* value on the performance. An empirical study leveraging the MADWifi (Multiband Atheros Driver for WiFi) Linux driver for Wireless LAN devices with the Atheros chipset has also be performed (with parameters set to optimal values derived in Section IV), the result of which will reported in Section V.

The PHY and MAC parameters, as suggested in MBOA MAC [24] for UWB-operated WPANs, have been used in our simulation. They are listed in Table I. Note that the UWB PHY will support a rate set of {53.3, 80, 110, 160, 200, 320, 400, and 480 Mbps}, among which support for transmitting and receiving at data rates of {53.3, 110 and 200 Mbps} is mandatory. We choose the highest mandatory rate, 200 Mbps, as the data rate. Both analytical and simulation results for other data rates exhibit similar trends and thus are not reported. Each simulation run lasts for 200 simulation seconds. Due to the space limit, in what follows we present three representative sets of simulation results in Figs. 4–7.

TABLE I
PHY AND MAC PARAMETERS AS SUGGESTED IN MBOA UWB MAC.

Channel Rate	200 Mb/s
Basic Rate	55 Mb/s
Slot Time	8 μsec
SIFS	10 μsec
ACK Time	13.125 μsec
MPDU[1] + FCS[2]	41.25 μsec
Data Payload	1024 Bytes

[1]MPDU: Message Data Protocol Unit.
[2]FCS: Frame Check Sequence

Impact of the *contention window size (CW)*: In the first set of results, we use the same AIFS value for all classes and assign different values of CW for each class. We consider three cases: In the first two cases (Fig. 4(a) and (b)), there are two priority classes each assigned a CW value (a) $CW_1 = 8, CW_2 = 16$, and (b) $CW_1 = 16, CW_2 = 32$. In the third case (Fig. 4(c)), there are 3 classes, each assigned a CW range: $CW_1 \in [8, 16]$, $CW_2 \in [16, 32]$ and $CW_2 \in [32, 64]$. Several observations are in order.

First, the analytical results agree with the simulation results very well, being in the interval $[0.9774, 1.0794]$ of the simulation results.

Second, stations that use smaller values of CW (e.g., 8 vs. 16 and 32, 16 vs. 32) grasp a large portion of available bandwidth. Clearly, a scheme that varies the values of CW is effective in allocating bandwidth in a QoS-controlled manner among different classes. (We will elaborate on how to determine the optimal values of CW to achieve deterministic proportional services differentiation in Section IV.)

Third, in general the throughput attained by each class decreases as the number of stations increases. This is consistent with our intuition, since the larger the number of stations the more likely collisions will occur. However, as shown in Fig. 4 (a), (b) and (c), the throughput curves corresponding to class 1 exhibit a peculiar trend. Specifically, in Fig. 4 (a), instead of a monotone decrease (as the number of stations in each class, N increases), the throughput increases (though slowly) between $N = 10$ and $N = 30$, before it concedes to the decreasing trend. Similar trends are observed in Fig. 4 (b) and (c), although when N is larger (the curves are cut off before the decreasing trend shows).

The above phenomenon is a result of the peculiar access behavior of the *post-busy* slot. Recall in the *post-busy* slot, only a subset of stations are eligible to contend for channel access, with a probability inversely proportional to their respective CW. Therefore we observe two access patterns: one is in the *post-busy* slot, and the other is in all the other slots. The later affects the former yet the former is independent of the later. When the collision is sparse, the probability that the *post-busy* slot is accessed is also slim. However, when collisions occur more frequently, the *post-busy* slot is also more likely to be accessed. Moreover, the probability that access to the *post-busy* slot is successful exhibits a similar trend as that for non- post-busy slots, but only lags in *phase*. Indeed we observe that when the number, N, of stations grows large, almost all non-*post-busy* slots incur collisions, and yet it is possible for a successful transmission or an idle period to occur in the *post-busy* slot. The combination of the two trends induces the interesting, peculiar fluctuation in the throughput. Note that as stations of class 1 usually have the smallest contention windows, they

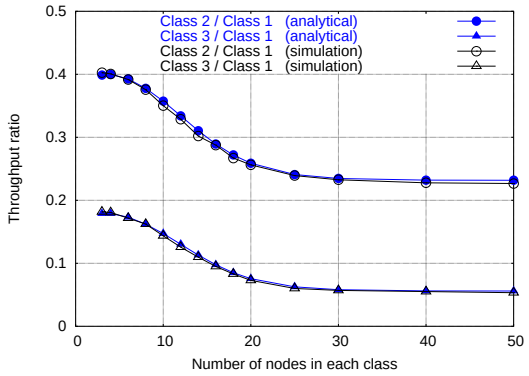


Fig. 5. Ratio of the throughput attained by a class $i+1$ station to that by a class 1 station, $i = 1, 2$. The configuration is the same as in Fig. 4 (c).

dominate in accessing the post-busy slot. This explains why the peculiar trend is only observed in the aggregated throughput attained by class-1 stations.

To verify whether or not EDCA can provide deterministic proportional QoS, we calculate, with the results shown in Fig. 4 (c), the ratio of the throughput attained by a class $i+1$ station to that by a class i station, $i = 1, 2$. As shown in Fig. 5, instead of being fixed at a stable level, the throughput ratio first decreases as N_i grows and then levels off as N_i continues to grow. This is again due to the peculiar behavior that results from that the access patterns to the *post-busy* slot and other non-*post-busy* slots are different. When N_i increases above certain level, the probability that access to the *post-busy* slot is successful decreases to the same level as that for non-*post-busy* slots, and the throughput ratio levels off. Another observation is that, although the throughput ratio indicates that stations of higher-priority classes do attain more throughput with the use of smaller CW ranges, the throughput ratio also depends on the number of stations in the system (which usually cannot be known *a priori*). Moreover, to provide deterministic proportional QoS, one has to determine the *optimal* CW ranges as a function of the number of stations of each class and the specified, proportional QoS.

Impact of the AIFS value: Now we study the impact of varying AIFS values on the performance of service differentiation. We consider two priority classes, both configured with the same congestion window size $CW = 16$ but different AIFS values. The high-priority class has $AIFSN_1 = 2$, and the other has $AIFSN_2 = 3, 5$, or 7. Fig. 6 gives both the simulation results and the analytical results. Several observations are in order.

First, the analytical results agree very well with the simulation results. Second (and more importantly), stations of the high-priority class (and with smaller AIFS values) almost grasp all the available bandwidth. In particular, when the number of stations in both classes reaches 12, 6 and 4 in three cases (a), (b) and (c), respectively, the throughput attained by class-2 stations is less than 1% of the bandwidth (200Mbps). This results from the fact that stations of class 1 can make access attempts in both contention zones 1 and 2, while stations of class 2 can only make attempts contention zone 2. This suggests that QoS provisioning by assigning different AIFS values to different access categories may lead to starvation of stations of low-priority access categories.

Combined impact of both the contention window and the AIFS value: Lastly we study the combined impact of both the contention window and the AIFS value on the performance of service differentiation. Again we use the configuration of the four access categories (AC) as defined in MBOA UWB MAC (and in IEEE 802.11e). Fig. 7 gives both simulation and analytical results for three different combinations of CW and AIFS values. All the observations made in the studies of varying either the contention window size or the AIFS value are observed (although in a mixed manner). In particular, in the presence of stations with the smallest AIFS (AIFSN=2), stations of all the other classes (AC3 and AC4) attain very little (close to zero) throughput.

IV. A Framework for QoS Provisioning in UWB MAC

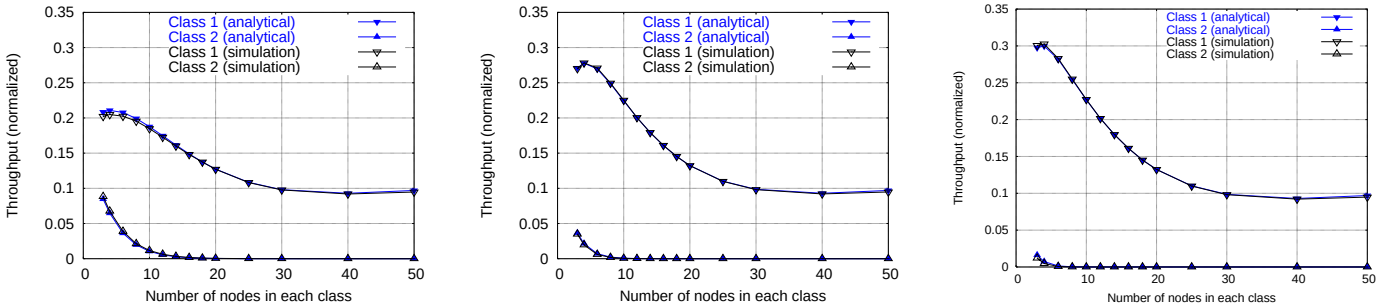
A. The Challenges

As mentioned in Section I, UWB MAC is expected to fulfill the following objectives: (i) support real-time flows (such as on-line music in the digital home environment with *fast access* and *small jitters*; and (ii) provide different levels of deterministic QoS to best-effort traffic of different access categories.

Guided by our analytical/simulation study, we have the following findings: first, objective (i) can be achieved by assigning small AIFS values to AC1. However, there are two drawbacks associated with this approach: (a) although stations in the AC1 category grasp most of the bandwidth, the nature of contention-based access cannot always guarantee small jitters; and (b) objective (i) is achieved really at the expense of starving stations of AC3–AC4 (Figs. 6 and 7). This suggests that if varying the AIFS values is indeed used as a control dimension to provide controlled access to real-time traffic, it should be used with caution and well-thought design.

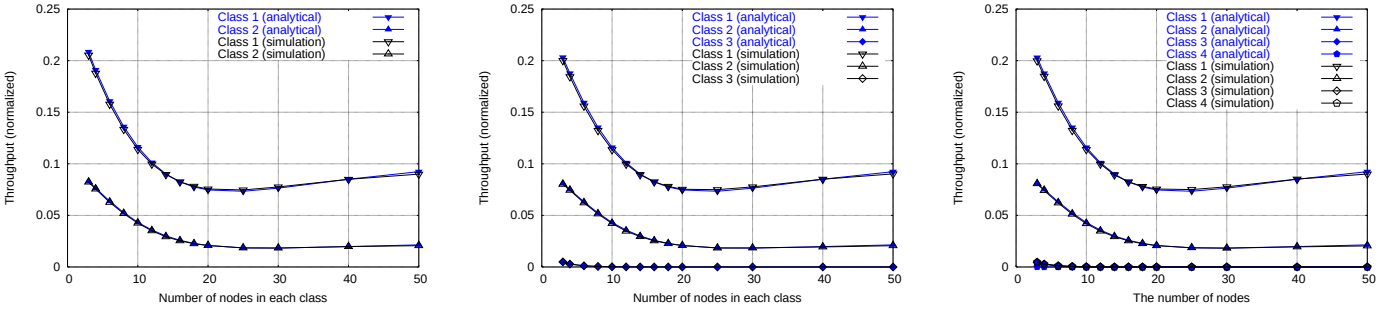
Second, although objective (ii) can be, to some extent, achieved by assigning different CW ranges to different access categories, the achieved throughput ratio has been shown to depend on the number of stations in the system (which usually cannot be known *a priori*) and cannot be fixed at a stable level under the currently recommended parameter setting. As a matter of fact, to provide *deterministic* proportional QoS, the *optimal* CW values as a function of the number of stations of each AC and the specified, proportional QoS is yet to be determined. Finally, both objectives (i) and (ii) should be realized without compromising bandwidth utilization.

To truly achieve objectives (i) and (ii) while maximizing the bandwidth utilization, we will leverage the *superframe structure together with a beacon mechanism* proposed in [18]



(a) 2 classes: $AIFSN_{1,2}$=2, 3 (b) 2 classes: $AIFSN_{1,2}$=2, 5 (c) 2 classes: $AIFSN_{1,2}$=2, 7

Fig. 6. Analytical and simulation results for multiple classes (each with a different AIFS value but the same contention window size ($CW = 16$)).



(a) 2 classes: AC1 and AC2 (b) 3 classes: AC1, AC2 and AC3 (c) 4 classes: AC1-4

Fig. 7. Analytical and simulation results under different combinations of CW and AIFS values (as defined for different categories in MBOA UWB MAC). The configuration of the four access categories, AC1-AC4, are respectively $CW_{1,2,3,4} = [8, 16], [16, 32], [32, 1024], [32, 1024]$, $AIFSN_{1,2,3,4} = 2, 2, 3, 7$.

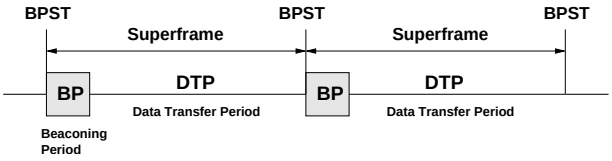


Fig. 8. The superframe structure in the MBOA MAC protocol.

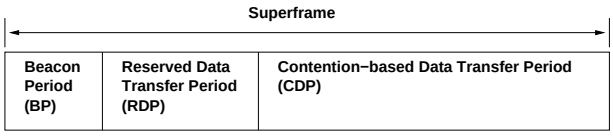


Fig. 9. Superframe structure.

[24]. The superframe structure was originally proposed to ease time-synchronization and certain distributed coordination. As illustrated by Fig. 8, the time is organized in superframes, each being divided into 1) a Beacon Period (BP) and 2) the Data Transfer Period (DTP). The beacon period is slotted and is used for all devices in the same UWB-operated network to register and identify themselves. Reservation can be made either in the beacon messages or in the data packets transmitted in the DTP.

B. Contention-based Reservation and Reservation-based Transmission for Real-time Traffic

To leverage the superframe structure with a beacon mechanism, we further divide the Data Transfer Period (DTP) into two parts: Reservation-based Data Transfer Period (RDP) and Contention-based Data Transfer Period (CDP), as illustrated in Fig. 9. RDP is used for stations with real-time traffic to transmit *after* the reservation has been made. CDP is used for stations with best effort traffic and/or real-time traffic in the reservation phase and is governed by EDCA.

A station with real-time traffic is required to use a *contention-based reservation access method*: it classifies the *first* packet of a real-time stream as traffic in the AC1 category and transmits it with the use of a small AIFS value. Sufficient information is also contained in this packet to make reservation in a RDP. After the reservation made is successfully confirmed (in a beacon message), the station with the real-time stream can then transmit in their reserved time slots in a RDP and hence incur no contention. In this manner, the amount of traffic that takes advantage of the aggressive bandwidth-grasping feature in the AC1 category is constrained only to the first packet of a real-time stream. Traffic in the other AC categories will not starve. Also, the reservation can be made within a short time (as traffic with a small AIFS value and small CW can access the medium quickly), with bandwidth being reserved in a periodic fashion in each of the subsequent RDPs.

The lengths of the superframe and the BP have been specified in [18] [24]. The length of the RDP may vary dynamically depending on the amount of real-time traffic present. The system can be configured to control the maximum RDP duration in a superframe, thus ensuring a certain portion of bandwidth for best-effort traffic. The bandwidth utilization and QoS provisioning in CDPs will be largely determined by how best-effort data traffic is governed to access slots in the CDPs, which is our focus in the reminder of the section.

C. Deterministic Proportional QoS Provisioning for Best-effort Traffic

As mentioned above, schemes that assign small AIFS values to high-priority access categories risk the possibility that stations of low-priority access categories (AC3–AC4) will starve. As such, we propose to consider only the dimension of varying contention window sizes in provisioning deterministic QoS to best-effort traffic.

For most applications in WPANs, the throughput attained by stations of different classes is perhaps the major measure of quality of service. As the *available* bandwidth in a wireless environment is variable and changes as the number of stations increases, instead of providing QoS in the form of absolute bandwidth, we aim to provide deterministic *proportional* QoS among best-effort traffic. We define the ratio, r_j, $j = 2, \cdots, M$, of per-station throughput attained by a station of class j to that attained by a station of class 1, i.e., $\frac{\eta_j}{N_j} = r_j \frac{\eta_1}{N_1}$, $j = 2, \cdots, M$.

As indicated in Section III, given the parameter settings currently recommended in [18], [23], the throughput ratio r_j dynamically changes as the number of stations varies (Fig. 5) and cannot be fixed at a stable level. Also, it is not clear whether or not the current parameter setting renders the maximal system throughput, although it has been proved in [5] that the current IEEE 802.11 parameter setting cannot achieve the maximal system throughput. In what follows, we study, by leveraging the analytical model derived in Section II, how the contention window sizes can be optimally set to provide deterministic proportional QoS and to maximize the system throughput.

d) Formulation of the throughput maximization and deterministic proportional QoS provisioning problem: We consider a UWB-operated network with M classes. Stations of all the classes are configured with the same AIFS value, but are assigned different contention window sizes CW_j, $1 \le j \le M$. Then the problem of combined throughput maximization and deterministic proportional QoS provisioning can be formally stated as

Problem 1: Given the throughput ratio r_j, $j = 2, \cdots, M$, determine the optimal contention window sizes CW_j, $j = 1, \cdots, M$ such that

$$\text{Maximize } \eta = \sum_{j=1}^{M} \eta_j \tag{14}$$

$$\text{s.t. } \frac{\eta_j}{N_j} = r_j \frac{\eta_1}{N_1}, \text{ for } j = 2, \cdots, M \tag{15}$$

Proposed solution: In the model given in Section II, the system throughput is derived as a function of the number of class-i stations (N_i), the contention window size (CW_i), the AIFS values (a_i). In principle, the optimization problem can be solved numerically. However, a simple, closed-form solution would be desirable so that stations can dynamically track the parameters in the solution, and on-line calculate the optimal solution.

Before delving into the derivation, we make the following observation. Figure 10 depicts the system throughput as a function of the contention window size (CW) in the case of one traffic class ($N = 5, 10, 20$ and 50). As shown in Fig. 10 (and as mentioned earlier), the analytical results derived under the proposed model agree very well with the simulation results, but those derived under the p-persistent model (which assume that all the stations independently access to *any* slot with a fixed probability) fails to do so. Nevertheless, both models give approximately the same optimal value of CW at which the system throughput is maximized. This is not a coincidence, because at the operational point where the maximal throughput is achieved (e.g., $CW = 20$ when $N = 5$), the channel is not overly congested and the peculiar effect of the access pattern to the *post-busy* slot has not yet became significant. Similar trends have also been observed in the case of multiple traffic classes. This observation suggests that as far as derivation of the optimal congestion window size is concerned, one can leverage the p-persistent model, subject to the proportional constraint Eq. (15). (We will further validate this conjecture in the simulation study in Section V.)

In the p-persistent model, stations of class j transmit in *a* slot independently and uniformly with probability τ_j. Given the contention window size CW_j, τ_j can be calculated as $\tau_j = \frac{2}{CW_j+1}$. Then the stationary probabilities of channel states, i.e., the *idle* state, the *successful class-j transmission* state, and the *collision* state, can be readily derived as

$$P_I = \prod_{j=1} (1-\tau_j)^{N_j} = A_M, \tag{16}$$

$$P_{S_j} = N_j \frac{\tau_j}{1-\tau_j} A_M, \tag{17}$$

$$P_C = 1 - P_I - \sum_{j=1}^{M} P_{S_j}. \tag{18}$$

For ease of exposition, we assume that the size of all data frames is a constant, and thus the duration of a successful transmission and a collision period are the same (i.e., $T_D = T_C$). (The assumption can be relaxed with modest modification.) Plugging the above stationary probabilities into Eqs. (12) and

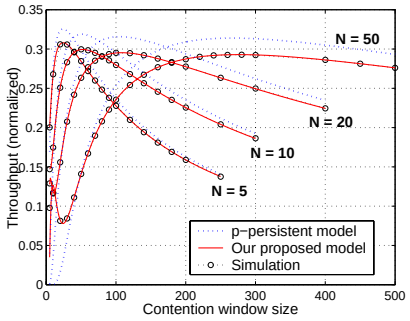
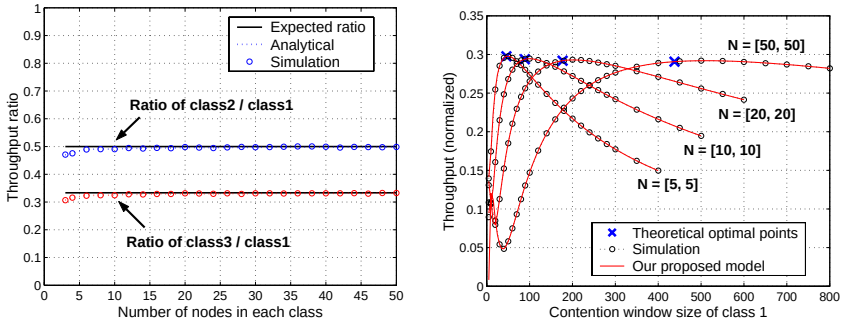


Fig. 10. The relationship between the saturation throughput and the contention window size. There is only one class in the system. N is the number of nodes.



(a) (b)

Fig. 11. The throughput ratio among different traffic classes ((a)) and the system throughput ((b)) in the case that there are 3 traffic classes and the proposed solution (given in Eq. (20)) is used to calculate the optimal contention window size. The QoS specified is $[r_2,\, r_3] = [0.5,\, 0.33]$.

(13), we have

$$\eta_j = \frac{\overline{m}}{T_D}\,\frac{N_j \frac{\tau_j}{1-\tau_j} A_M}{1 - A_M\left(1 - \frac{t_s}{T_D}\right)}. \tag{19}$$

Now we are in a position to derive the optimal value of CW_j, $1 \le j \le M$.

Theorem 1: Given the expression for system throughput, Eq. (19), the optimal solution to Problem 1 (defined in Eqs. (14) and (15)) is: for $j = 1, \cdots M$,

$$CW_j^* = \frac{\sqrt{2\beta T_D'}}{r_j} + 1, \tag{20}$$

where $\beta = \left(\sum_{j=1}^{M} N_j\, r_j\right)^2 + \sum_{j=1}^{M} N_j\, r_j^2$ and $T_D' = T_D/t_s$, that is, the duration of a successful transmission in the unit of slots. $r_1 \equiv 1$.

Proof: Refer to Appendix II.

Discussion: For a station to compute its optimal contention window size according to Eq. (20), it has to know the number of active stations in each class. As proposed in [24], each station is required to register and identify itself in every beacon period. It is thus convenient for each station to indicate in the beacon message whether or not it has packets of certain classes. By the end of each beacon period, a station can collect the information and estimate the number of active stations in each class.

Throughout the discussion, we have assumed that the basic access method without the RTS-CTS floor acquisition mechanism is used, and that all stations operate at a common data rate. However, the results can be readily extended to accommodate more general scenarios in which such assumptions are relaxed.

V. Analytical and Simulation Studies

In this section, we present our analytical and simulation studies. In the next section, we will report our experience of implementing and evaluating the enhanced EDCA mechanism on a Linux-based MADWifi driver for wireless LAN devices with the Atheros chipset. We first evaluate the analytical solution derived in Section IV-C with respect to its capability of achieving the proposed optimization goal. Then we evaluate the performance of the proposed framework for quality controlled medium access in a UWB-operated network.

A. Evaluation of the Solution Given in Section IV-C

We consider a UWB-operated network in which there are three classes with the requested QoS $[r_2,\, r_3] = [0.5,\, 0.33]$. The optimal solution to the congestion window size (Eq. (20)) is used to on-line calculate CW_j. Fig. 11 gives the throughput ratio among different traffic classes ((a)) and the system throughput ((b)). The number, N_i, of class-i stations varies from 3 to 50. As shown in Fig. 11 (a), the requested QoS and the throughput ratios obtained via the analytical model and the simulation agree very well, except the slight difference when N_i is small ($N_i = 3 - 8$).

To evaluate the capability of the proposed solution (Eq. (20)) in achieving maximal system throughput, we depict, for each combination of N_i's, the total system throughput as a function of the contention window size in Fig. 11 (b). For each value of CW_1 along the x-axis, CW_j is computed as $CW_j = \frac{CW_1 - 1}{r_j} + 1$ ($j = 2, 3$) (which is derived from Eq. (20)). Also, the optimal values of the contention window size computed from Eq. (20) are marked for comparison. As shown in Fig. 11 (b), for all the cases ($N = 5, 10, 20, 50$) the optimal values calculated from Eq. (20) and those identified in the simulation (and in the analytical model derived in Section II) do coincide.

B. Performance of the proposed UWB MAC framework

In this set of simulation, we consider an UWB-operated network with both real-time and best-effort traffic. The durations of a BP, RDP+CDP, MPDU+FCS are set, respectively, to 82.50 μsec, 825.0 μsec and 41.25 μsec. The other PHY/MAC parameters are specified in Table I. Each real-time stream is assumed to deliver 4-9 Mb/s MPEG-2 stream, and hence a

station with real-time traffic makes reservation for at most 9 Mb/s bandwidth when the real-time stream arrives. The tunable parameters for real-time traffic are set to $[CW_{min}, CW_{max}] = [8, 16]$, and $AIFSN = 2$. The AIFS value for best-effort traffic of all classes is set to $AIFSN = 3$ and the CW values of different classes are on-line calculated in compliance with Eq. (20). There are two traffic classes for best-effort traffic, with the number of class i stations being 10. (Similar trends have been observed when the number of class i stations is set to other values.) Each simulation run lasts for 200 simulation seconds.

Performance in the presence of two classes of best-effort traffic and one real-time stream: Fig. 12 depicts the total throughput and the throughput attained by each class in the presence of two classes of best-effort traffic and a real-time stream. The real-time stream is active in the period of [50s, 150s]. As shown in Fig. 12, the real-time stream is allocated most of the bandwidth in [50s, 150s], while the best-effort traffic uses the remaining bandwidth. The bandwidth allocated for the real-time stream is $\frac{41.25 \cdot 10^{-6}}{(82.50+825.0) \cdot 10^{-6}} \times 200 \cdot 10^6 \approx 9(Mb/s)$. No matter whether or not the real-time stream is active, the best-effort traffic shares the remaining bandwidth in a deterministic proportional manner (as specified by $r_{3/2}$).

Performance in the presence of two classes of best-effort traffic and two groups of real-time streams: Fig. 13 gives the total throughput and the throughput attained by each class in the presence of two real-time streams and two classes of best-effort traffic. The first group of real-time streams originates from 5 nodes, becomes active at 50s and terminates after 100s. The second group of real-time streams originates from another 5 nodes, becomes active at 70s and terminates after 60 seconds.

As shown in Fig. 13, with this communication pattern, we have five intervals in the entire simulation time. In [0, 50s], the best-effort traffic of two different classes shares the available bandwidth according to the specified QoS ($r_{3/2}$); in [50s, 70s], the first group of real-time streams join to acquire bandwidth, with the best-effort traffic of two different classes sharing the remaining bandwidth in CDPs, again according to the specified QoS; in [70s, 130s], the second group of real-time streams join, and together with the first group, acquire most of the bandwidth. The best-effort traffic of two different classes is allocated the remaining small bandwidth in a deterministic proportional manner; in [130s, 150s], the best-effort traffic is allocated more bandwidth since the second group of real-time traffic streams terminates; Finally, in [150s, 200s], the best-effort traffic is allocated the entire available bandwidth. Even in the presence of real-time traffic, best-effort traffic of two different classes is allocated bandwidth in a deterministic proportional manner (as specified by $r_{3/2}$).

VI. Empirical Study

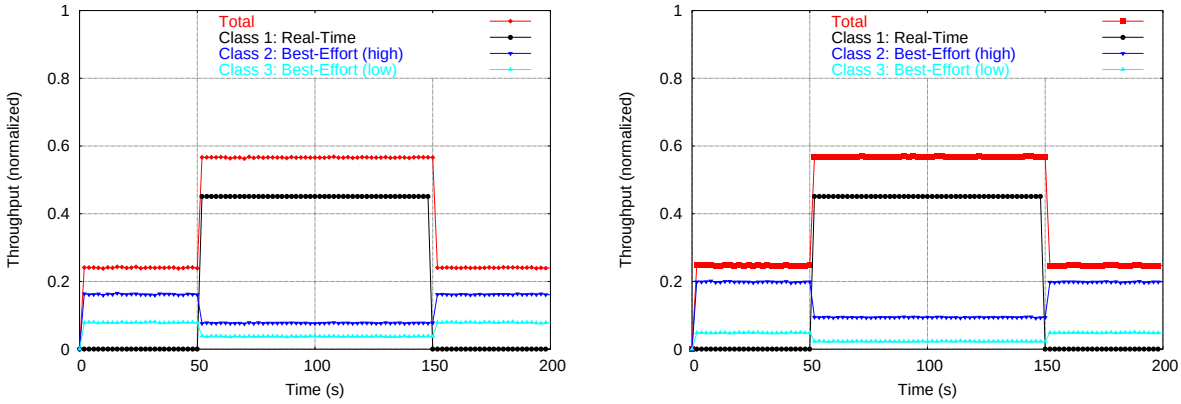
We have developed an experimental prototype to validate the analytical and simulation results, to demonstrate the practicality of the enhanced EDCA algorithm, and to understand implementation issues of integrating the algorithm into the current IEEE 802.11 protocol family. We first summarize how we implement an experimental prototype of the enhanced EDCA mechanism (that includes the service differentiation algorithm described in Section IV-C) on a Linux-based MADWifi driver for wireless LAN devices with the Atheros chipset. (Note that as no UWB chipsets and drivers are commercially available on the market, we can only implement our proposed framework on a WiFi device. Fortunately the chosen chipset does not require loading of IEEE 802.11-specific firmware, but instead relies on a Hardware Access Layer (HAL) module that allows changes of several device parameters through its well-defined interface. This allows us to readily implement the enhanced EDCA mechanism.) Then we present representative empirical results.

A. Developing an Experimental Prototype

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.

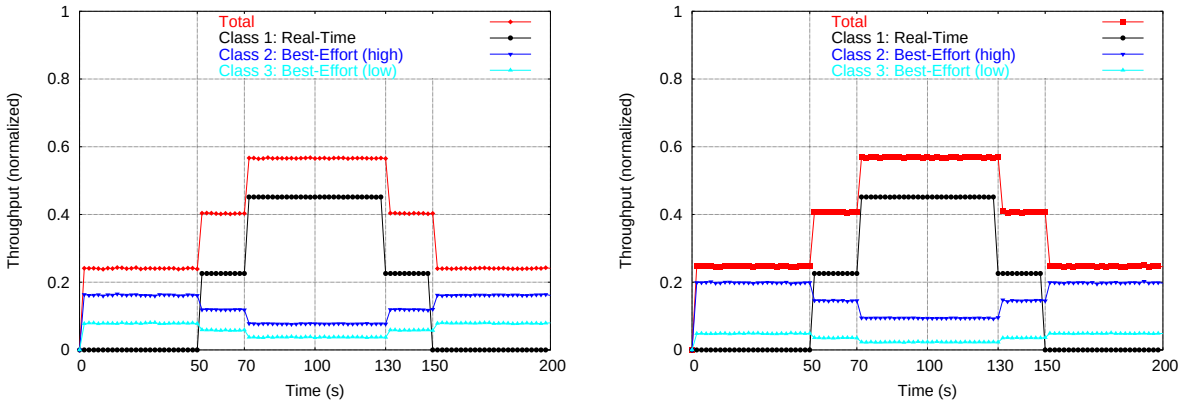
The Atheros hardware, on the other hand, does not require loading of firmware, but instead relies on a *Hardware Access Layer (HAL)* module that is provided in a binary-only form. The HAL module operates between the hardware and driver to manage much of the chip-specific operations and to enforce the 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. (20) in Section IV-C 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.

How to support floating point operations: Apart from several low-level implementation details (which the interested reader is referred to [6]), there are two major implementation issues that arises in the course of prototype implementation.



(a) QoS specification: $r_{3/2}$ = 0.5 (b) QoS specification: $r_{3/2}$ = 0.25

Fig. 12. The total throughput and the throughput attained by each class in the presence of two classes of best-effort traffic and a real-time stream.



(a) QoS specification: $r_{3/2}$ = 0.5 (b) QoS specification: $r_{3/2}$ = 0.25

Fig. 13. The total throughput and the throughput attained by each class in the presence of two real-time streams and two classes of best-effort traffic.

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. There are a number of possible solutions, including using a lookup table or adapting the floating point unit and emulation code in the kernel. They were found to be not viable. In the former case the lookup table would be extremely large in order to store all the possible values. In the latter case, even if the floating-point unit and emulation code are adapted correctly in the kernel, a separate math library would have to be written because the C runtime library cannot be used inside the kernel. Therefore, in order to realize the necessary functionality and to ensure that only essential, performance-critical code is implemented in the kernel, 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 IV-C, 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.

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.

TABLE II

RELEVANT PARAMETERS USED BY THE ATHEROS DRIVER.

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

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 [11], 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.

B. Empirical Results

The network topology used for the empirical study consists of two mobile stations and one AP (Access Point) 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 II 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. 14 shows the throughput results when $\widehat{r}_2 = 4$. The total system throughput 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.

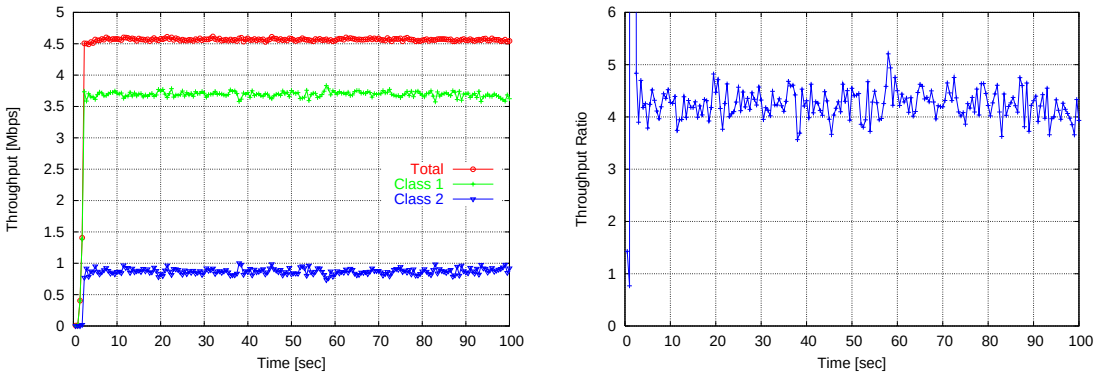
Performance in the presence of two traffic classes with on-off traffic sources: In this set of experiments, only the class-2 station is active during the entire experiment. The class-1 station sends packets in an on-off manner, with the duration of its on and off periods being set to ~ 20 sec.

Fig. 15 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 throughput 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 when the class-1 station is inactive. 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.

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. (20) 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.

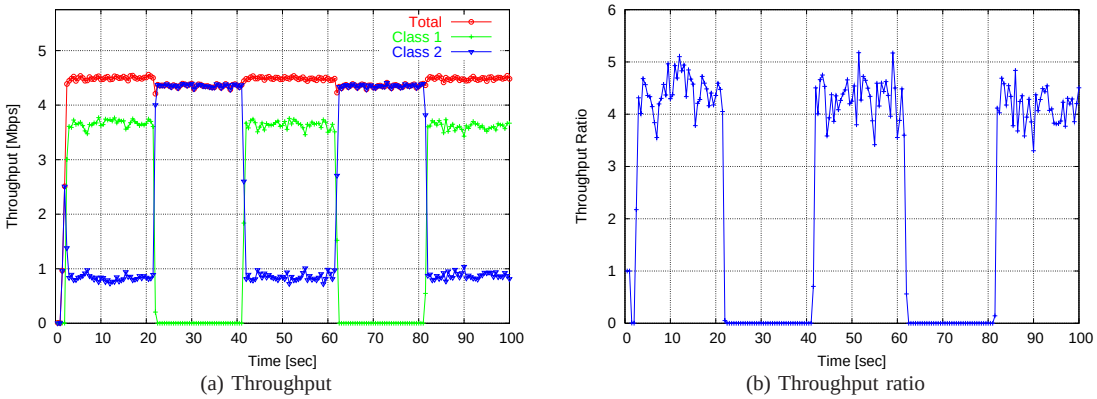
VII. CONCLUSION

The EDCA mechanism has been proposed by MultiBand OFDM Alliance [18] to support service differentiation in UWB-operated WPANs. EDCA achieves service differentiation essentially by configuring different traffic classes with different contention window sizes and AIFS values. In this paper, we have conducted a rigorous, comprehensive, and theoretical analysis of the EDCA mechanism, and have shown that with the currently recommended parameter setting, EDCA cannot provide deterministic proportional QoS. In particular, stations of a high-priority class (i.e., with a small AIFS value) will dominate the channel access, depriving stations of the other classes the chance to access the channel. Also, without responding to



(a) Throughput (b) Throughput ratio

Fig. 14. Throughput attained by two traffic classes with constant traffic sources. $\widehat{r}_{12} = 4$.



(a) Throughput (b) Throughput ratio

Fig. 15. Throughput attained by two traffic lasses with on-off traffic source. $\widehat{r}_{12} = 4$.

the system dynamics (e.g., taking into account of the number of active class-i stations), EDCA cannot allocate bandwidth in a deterministic proportional manner and the system bandwidth is under-utilized.

After identifying the deficiency of EDCA in service differentiation, we propose, in compliance with the EDCA-incorporated UWB MAC protocol proposed in [18] [23] [24], a framework, along with a set of theoretically grounded methods for controlling medium access with deterministic QoS for UWB networks. In this framework, 1) real-time traffic is guaranteed of deterministic bandwidth via a *contention-based* reservation access method; 2) best-effort traffic is provided with deterministic proportional QoS; and moreover, 3) the bandwidth utilization is maximized. The performance of the proposed framework has been validated and evaluated in analytic, simulation, and empirical studies.

As mentioned in Section IV-B, to prevent best-effort traffic from starvation, the system should be configured to control the maximum RDP duration in a superframe, thus ensuring a certain portion of bandwidth for best-effort traffic. This entails admission control of real-time traffic. As part of our future work, we will design an auxiliary admission control protocol that determines when and for how long real-time streams can reserve certain amount of bandwidth. Also, we have focused in this work service differentiation (for best-effort traffic) in the form of deterministic proportional bandwidth allocation. It would be interesting to extend the work to accommodate service differentiation in the form of statistical end-to-end delay guarantees.

REFERENCES

[1] IEEE 802.15.3 specification. http://www.ieee802.org/15/pub/TG3.html.

[2] MultiBand OFDM Alliance SIG, MultiBand OFDM physical layer proposal for IEEE 802.15 task group 3a, Sept. 2004.

[3] IEEE 802.11e/d13.0. Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: MAC Enhancements for Quality of Service (QoS), Jan. 2005.

[4] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE JSAC*, 18(3), Mar. 2000.

[5] 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), Dec. 2000.

[6] D. Chi. Design and implementation of the generic wireless device driver layer to support QoS provisioning. Master's thesis, University of Illinois at Urbana-Champaign, Aug. 2005.

[7] S. Choi, J. del Prado, S. Shankar, and S. Mangold. IEEE 802.11e contention-based channel access (EDCF) performance evaluation. In *Proc. of IEEE ICC*, May 2003.

[8] Y. Ge. QoS provisioning for IEEE 802.11 MAC protocols. Ph.D Thesis, University of Ohio State, 2004.

[9] C. Hu, H. Kim, and J. C. Hou. An analysis of the binary exponential backoff algorithm in distributed MAC protocols. In *Tech. Rep. No. UIUCDCS-R-2005-2599. http://lion.cs.uiuc.edu/∼chunyuhu*, July 2005.
[10] J. Hui and M. Devetsikiotis. Performance analysis of IEEE 802.11e EDCA by a unified model. In *Proc. of GLOBECOM*, Nov. 2004.
[11] IEEE Computer Society. IEEE standard 802.11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.
[12] N. Laurenti and P. Toniolo. Performance of the multi-band OFDM UWB system with time-varying channels. In *Proc. of WPMC*, Sept. 2004.
[13] B. Li and R. Battiti. Performance analysis of an enhanced IEEE 802.11 distributed coordination function supporting service differentiation. In *Quality for All, QofIS*, 2003.
[14] A. Lindgren, A. Almquist, and O. Schelen. Evaluation of quality of Service schemes for IEEE 802.11 wireless LANs. In *Proc. of IEEE LCN*, Nov. 2001.
[15] A. Lindgren, A. Almquist, and O. Schelen. Quality of service schemes for IEEE 802.11 wireless lans - an evaluation. *In Special Issue of the Journal on Special Topics in MONET on Performance Evaluation of QoS Architectures in Mobile Networks*, 8(3), June 2003.
[16] K. Lu, D. Wu, Y. Fang, and R. C. Qiu. On medium access control for high data rate ultra-wideband ad hoc networks. In *Proc. of IEEE WCNC*, Mar. 2005.
[17] S. Mangold, S. Choi, P. May, and G. Hiertz. IEEE 802.11e - fair resource sharing between overlapping basic service sets. In *Proc. of IEEE PIMRC*, Sept. 2002.
[18] MBOA. Distributed medium access control (MAC) for wireless networks. Draft specification 0.93, Feb. 2005.
[19] R. Merz, J.-Y. L. Boudec, J. Widmer, and B. Radunovic. A rate-adaptive MAC protocol for low-power ultra-wide band ad-hoc networks. In *Proc. of Ad-Hoc Now*, July 2004.
[20] D. Pong and T. Moors. Call admission control for IEEE 802.11 contention access mechanism. In *Proc. of IEEE GLOBECOM*, Dec. 2003.
[21] V. Ramaiyan and A. Kumar. Fixed point analysis of single cell IEEE 802.11e WLANs: Uniqueness, multistability and throughput differentiation. In *Proc. of ACM SIGMETRICS*, June 2005.
[22] J. W. Robinson and T. S. Randhawa. Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function. *IEEE JSAC*, 22(5), 2004.
[23] J. D. Sai Shankar N, V. Gaddam and K. Challapali. The new MBOA MAC specification: A distributed MAC protocol and OFDM based PHY for next generation WPANs.
[24] S. Shankar, J. del Prado Pavòn, V. G, and K. Challapali. Performance evaluation of the multiband OFDM alliance (MBOA) specification: A distributed MAC protocol and OFDM PHY layer for next generation ultra wide band (UWB) WPANs. In *submission*, 2005.
[25] J. Tourrilhes. Linux wireless LAN howto. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.pdf, January 2005.
[26] Y. Xiao. An analysis for differentiated service in IEEE 802.11 and IEEE 802.11e wireless LANs. In *Proc. of IEEE ICDCS*, Mar. 2004.
[27] J. Zhao, Z. Guo, Q. Zhang, and W. Zhu. Performance study of MAC for service differentiation in IEEE 802.11. In *Proc. of GLOBECOM*, 2002.

APPENDIX I
LEMMA 1 AND LEMMA 2

Lemma 1: Given the Markov chain described in Section II (Fig. 2), the transition probability from state C_{a_j}, $j = m, \cdots, M$ to state I_{a_1} can be approximately expressed as

$$P[C_{a_j} \rightarrow I_{a_1}] = \frac{1}{P[I_k \rightarrow S_u]} \left\{ B_m - A_j \left[1 + \sum_{k=1}^{m} \frac{N_k \tau_k}{1-\tau_k} \left(1 - \frac{1}{CW_k} \right) + \sum_{k=m+1}^{j} \frac{N_k \tau_k}{1-\tau_k} \right] \right\}. \quad (21)$$

Proof: $P(C_{a_j} \rightarrow I_{a_1})$ is the probability that the *post-busy* slot after a collision slot C_{a_j} is idle. For notational convenience, tag the collision slot and its *post-busy* slot by $slot_1$ and $slot_2$, respectively. Let E_j denote the event that $(n_1, n_2, \cdots, n_j)$ stations transmit in $slot_1$, where n_k ($n_k = 0, 1, \cdots, N_k$) is the number of stations in class k. We have

$$\begin{aligned} P[C_{a_j} \rightarrow I_{a_1}] &= P\left[slot_2 \text{ is idle, given } slot_1 \text{ is a collision of } C_{a_j} \right] \\ &= \underbrace{\sum_{n_1} \sum_{n_2} \cdots \sum_{n_j}}_{\sum_{k=1}^{j} n_k \geq 2} P\left[slot_2 \text{ is idle } | E_j \right] \; P\left[E_j | \; slot_1 \text{ is } C_{a_j} \right]. \end{aligned} \quad (22)$$

Given the event E_j, $slot_2$ is idle if and only if none of the stations of the first m classes transmit in $slot_2$. Therefore,

$$P\left[slot_2 \text{ is idle } | E_j \right] = \prod_{k=1}^{m} \left(1 - \frac{1}{CW_k} \right)^{n_k}. \quad (23)$$

To compute precisely the probability $P\left(E_j | slot_1 \text{ is } C_{a_j}\right)$ it is necessary to enumerate the channel states and expand the channel state space $\mathbb{S}$ into $(n_1, n_2, \cdots, n_M)$. Hence, we leverage the fact that the channel state immediately following a busy period is mostly likely to be the idle state, because only nodes that are involved in the busy period will contend in the *post-busy* slot ($slot_2$). Based on this argument (which was collaborated by the simulation results), we have

$$P\left[E_j | \; slot_1 \text{ is } C_{a_j} \right] = \frac{1}{P[I_k \rightarrow S_u]} \prod_{k=1}^{j} \binom{N_k}{n_k} {\tau_k}^{n_k} (1-\tau_k)^{N_k - n_k}. \quad (24)$$

Plugging Eqs. (23) and (24) into Eq. (22), and after performing some algebraic operations, (21) follows. ■

Lemma 2: Given the Markov chain described in Section II (Fig. 2), the transition probability from state C_{a_j}, $j = m, \cdots, M$ to state S_u ($u \leq m$) can be approximately expressed as

$$P\left[C_{a_j} \rightarrow S_u \right] = \frac{1}{P[I_k \rightarrow S_u]} \; N_u \; \frac{\tau_u}{CW_u} \left(\frac{B_m}{1 - \frac{\tau_u}{CW_u}} - \frac{A_j}{\tau_u} \right). \quad (25)$$

Proof: Following the notation defined in the proof of Lemma 1, $P[C_{a_j} \rightarrow S_u]$ is the probability $P[slot_2 \text{ is a successful trans. of class } u | slot_1 \text{ is a collision } C_{a_j}]$, or simply $P[slot_2 \text{ is } S_u | slot_1 \text{ is } C_{a_j}]$. By conditioning on the

event E_j, we have

$$P\left[C_{a_j} \rightarrow S_u\right] = \underbrace{\sum_{n_1}\sum_{n_2}\cdots\sum_{n_j}}_{\sum_{k=1}^{j} n_k \geq 2} P\left[slot_2 \text{ is } S_u | E_j\right] P\left[E_j |\ slot_2 \text{ is } C_{a_j}\right]. \tag{26}$$

A successful transmission of class u follows a collision of C_{a_j} if and only if only one station of class u chooses to transmit in $slot_2$ with probability $\frac{1}{CW_u}$. That is,

$$\begin{aligned} & P\left[slot_2 \text{ is } S_u | E_j\right] \\ &= n_u \frac{1}{CW_u}\left(1-\frac{1}{CW_u}\right)^{n_k-1} \prod_{k=1,k\neq u}^{m}\left(1-\frac{1}{CW_k}\right)^{n_k} \\ &= \frac{n_u}{CW_u - 1}\prod_{k=1}^{m}\left(1-\frac{1}{CW_k}\right)^{n_u}. \end{aligned} \tag{27}$$

The probability $P\left[E_j | slot_1 \text{ is } C_{a_j}\right]$ is the same as Eq. (24). Plugging Eqs. (27) and (24) into Eq. (26), we can derive $P\left[C_{a_j} \rightarrow S_u\right]$ as given in Eq. (25). ■

APPENDIX II
PROOF OF THEOREM 1

Theorem 1: Given the expression for system throughput, Eq. (19), the optimal solution to Problem 1 (defined in Eqs. (14) and (15)) is: for $j = 1, \cdots M$,

$$CW_j^* = \frac{\sqrt{2\beta T_D'}}{r_j} + 1, \tag{28}$$

where $\beta = \left(\sum_{j=1}^{M} N_j\, r_j\right)^2 + \sum_{j=1}^{M} N_j\, r_j^2$ and $T_D' = T_D/t_s$, that is, the duration of a successful transmission in the unit of slots. $r_1 \equiv 1$.

Proof: The throughput by class j (Eq. (19)) is

$$\eta_j = \frac{\overline{m}}{T_D}\,\frac{N_j \frac{\tau_j}{1-\tau_j} A_M}{1 - A_M\left(1-\frac{t_s}{T_D}\right)},$$

where $\overline{m}$, t_s and T_D are constant variables, representing the average packet size, the length of an idle slot and the duration of a successful transmission, respectively. $A_M = \prod_{h=1}^{M}(1-\tau_h)^{N_h}$ as defined in Eq. (1).

Let $x \triangleq \frac{\tau_1}{1-\tau_1}$, $\alpha \triangleq \frac{\overline{m}}{T_D}$, and $\theta = 1 - \frac{t_s}{T_D}$ and $A \triangleq A_M$. Then we can further simplify the above equation as

$$\eta_j = \alpha \frac{N_j \frac{\tau_j}{1-\tau_j} A}{1-\theta A}. \tag{29}$$

To fulfill the proportional bandwidth allocation requirement Eq. (15), we have

$$\begin{aligned} r_j = \frac{\eta_j}{N_j} \Big/ \frac{\eta_1}{N_1} &= \frac{\tau_j}{1-\tau_j}\,\frac{1}{x} \\ \Rightarrow \tau_j &= \frac{r_j\, x}{1 + r_j\, x}. \end{aligned} \tag{30}$$

The total system throughput is the summation of the throughput achieved by each class, i.e.,

$$\begin{aligned} \eta = \sum_{j=1}^{M}\eta_j &= \alpha \sum_{j=1}^{M} N_j\, r_j \frac{xA}{1-\theta A} \\ &= \left(\alpha \sum_{j=1}^{M} N_j\, r_j\right)\frac{x}{\frac{1}{A}-\theta}. \end{aligned} \tag{31}$$

By Eq. (30), we have $1 - \tau_j = 1 - \frac{r_j\, x}{1+r_j\, x} = \frac{1}{1+r_j\, x}$, and

$$A = \prod_{j=1}^{M}(1-\tau_j)^{N_j} = \frac{1}{\prod_{j=1}^{M}(1+r_j\, x)^{N_j}}. \tag{32}$$

Using the Taylor series to approximate A, we have

$$\begin{aligned} (A^{-1})(x) &= 1 - d_0\, x + d_1\, x^2 + o(x^2) \\ &= 1 - d_0\, x + d_1\, x^2, \end{aligned} \tag{33}$$

where

$$d_0 = -(A^{-1})'(0) = \sum_{j=1}^{M} N_j\, r_j, \tag{34}$$

$$d_1 = (A^{-1})''(0) = \frac{1}{2}\left[\left(\sum_{j=1}^{2} N_j\, r_j\right)^2 + \sum_{j=1}^{M} N_j\, r_j^2\right]. \tag{35}$$

The total system throughput can then be expressed in terms of x:

$$\eta(x) = \alpha d_0 \frac{x}{1 - d_0 x + d_1 x^2 - \theta} = \frac{\alpha d_0}{\frac{1-\theta}{x} - d_0 + d_1 x}. \tag{36}$$

Let the denominator of the above equation be defined as $y(x) \triangleq \frac{1-\theta}{x} - d_0 + d_1 x$. By setting $\frac{dy(x)}{dx} = 0$, we have $x^* = \sqrt{\frac{1-\theta}{d_1}} = \sqrt{\frac{2}{\beta T_D'}}$ (where the second equality results from $1-\theta = \frac{t_s}{T_D} = \frac{1}{T_D'}$ and $d_1 = \frac{1}{2}\beta$). Note that $\frac{d^2 y(x)}{dx^2} > 0$, i.e., $y(x)$ is minimized at x^*.

From the two relations, $\tau_1 = \frac{2}{CW_1+1}$ and $x = \frac{\tau_1}{1-\tau_1}$, it is easy to obtain $CW_1 = \frac{2}{x} + 1$. Therefore, the system throughput $\eta(x)$ is maximized at

$$CW_1^* = \sqrt{2\beta T_D'} + 1, \tag{37}$$

and the proportional bandwidth allocation is achieved at

$$CW_j^* = \frac{\sqrt{2\beta T_D'}}{r_j} + 1, \tag{38}$$

for $j = 2, \cdots, M$. ■