A Novel Approach to Contention Control in IEEE 802.11e–Operated WLANs

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

***Abstract*— In this paper, we devise, in compliance with the IEEE 802.11e protocol [1], a novel MAC-centric approach, called *MAC contention control* (MCC), to maximizing the bandwidth utilization and achieving proportional bandwidth allocation. We first show that approaches based on estimating the number of competing nodes and then setting the contention window size may not converge (and in some cases diverge) because of network dynamics. Then, by studying the optimality condition derived in our prior work [10], we identify two parameters (referred to as *control references*) that remain approximately constant when the network operates at the optimal operational point, *regardless of the number of competing nodes in each AC.* We instrument MCC to measure these control references, compare measurement results to their optimal control reference levels, and adjust the packet dequeuing rate from the interface queues in an additive-increase-multiplicative-decrease (AIMD) fashion and with respect to prespecified bandwidth allocation ratio associated with its AC. In some sense, MCC controls the rate of passing packets from the interface queues to to the MAC access function, and thus practically controls the effective number of competing nodes.**

We have conducted an extensive simulation study, and demonstrated the superiority of MCC to 802.11e in terms of both the achievable network throughput and the capability of achieving proportional bandwidth allocation. This, coupled with the fact that MCC does not require change in firmware and can be practically deployed, makes MCC a viable approach to contention control in IEEE 802.11e-operated WLANs.

I. Introduction

With the proliferation of laptops, tablet PCs, and various consumer electronic devices (such as PDAs, digital cameras and wireless media centers), IEEE 802.11-operated wireless networks have become a common means for ubiquitous interconnectness and/or Internet access. With the increase in both the number and types of wireless applications, improving the network performance with respect to two performance criteria — *bandwidth utilization efficiency* and *Quality of Service (QoS)* — has been the focus of recent research. One widely used QoS is *proportional bandwidth allocation*, i.e., a flow of class-j is guaranteed to attain throughput r_j times that a flow of class-1, where r_j is the prespecified throughput ratio of a flow of class-j over a flow of class-1.

The issues of maximizing the bandwidth utilization and provisioning QoS are challenging. Due to the broadcast nature of the wireless medium, the actual achievable network throughput largely depends on the network dynamics such as the number of competing nodes and/or their traffic load. In addition, medium access among nodes in a distributed manner, if not well coordinated, easily incurs a significant amount of overhead (e.g., 30-40% of the available bandwidth [2]). In this paper, we aim to devise, in compliance with the IEEE 802.11e protocol [1], a novel MAC-centric approach to maximizing the bandwidth utilization and achieving proportional bandwidth allocation.

The Distributed Coordination Function (DCF) defined in IEEE 802.11 is a CSMA/CA-based medium access method. It employs a binary exponential backoff algorithm to mitigate contention and resolve collision among nodes. Extensive studies [2], [4], [5], [11] have been conducted to theoretically understand and enhance the protocol capacity achieved under DCF. IEEE 802.11 Task Group E has also devised an IEEE 802.11e MAC protocol [1]. In particular, the IEEE 802.11e draft standard defines a new coordination function, called the *Hybrid Coordination Function (HCF)*, that includes a contention-based channel access method, called *Enhanced Distributed Channel Access (EDCA)*. EDCA envisions an architecture of multiple access categories (ACs). Nodes contend for medium access in essentially the same manner as they do under DCF, except that several parameters that control *how* and *when* a node gains access to the medium differ among different ACs, so as to favor/disfavor data transmission from high-priority/low-priority flows. These parameters include the *minimum idle delay before contention* (AIFS), the *minimum and maximum contention windows* (CW_{min} and CW_{max}), and the *transmission opportunity limit* (TXOP). Several efforts have been made in [7], [10], [13]–[16] to theoretically understand the achievable network throughput and QoS capacity under EDCA. It has been well recognized that tuning the CWs among different ACs is an effective means to achieving proportional bandwidth allocation. It has also been shown that the optimal operating condition (in terms of the optimal CW size) for achieving the maximal bandwidth utilization exists for both DCF (e.g., [4]) and EDCA (e.g., [10]), and that the binary exponential backoff mechanism fails to approach this optimal operating condition. Several algorithms have been proposed in [10], [14], [17] to set/tune the CW size.

Because the optimal CW size is a function of both the total number of ACs and the number of competing nodes in each AC (among other parameters), most approaches suggest that each node estimates the number of competing nodes and calculates/sets the optimal values of CWs accordingly. In this paper, we show that approaches based on estimating the number of competing nodes may not converge (and in some cases diverge), thus leading to suboptimal performance. Instead, we propose a novel MAC-centric approach, called *MAC contention control* (MCC), that is positioned between the interface queues and the MAC access function. Rather than tuning the contention window size, MCC controls the rate of passing packets to the MAC access function, and thus practically controls the effective number of competing nodes. Considering the fact that the MAC access function in most wireless devices is implemented in firmware and its parameters (such as the contention window size) cannot be readily modified, MCC is more promising to be practically deployed.

MCC is composed of three procedures: *measurement*, *feedback* and *control*. By studying the optimality condition derived in our prior work [10], we identify two parameters that remain approximately constant when the network operates at the optimal operational point, *regardless of the number of competing nodes in each AC*. These two parameters can be readily measured and reflect accurately whether or not the network operates in the optimal state. We term these parameters as the *control references*. By comparing measurement results to the optimal control reference levels, MCC generates feedback accordingly, with which the packet dequeuing rate from the interface queues is adjusted according to an additive-increase-multiplicative-decrease (AIMD) algorithm and the prespecified bandwidth allocation ratio associated with its AC. (Note that AIMD has been proved in [6] to converge to the optimal operational point in finite steps.) All the measurement, feedback and control procedures are carried out by all nodes independently and in a distributed manner.[1] We have conducted an extensive simulation study under a wide variety of network scenarios, including single/multiple traffic classes, persistent/on-off traffic, and several different packet payloads. MCC is shown to outperform the 802.11e EDCA function in terms of both the achievable network throughput and the capability of achieving proportional bandwidth allocation. We have also investigated how sensitive MCC is to the choice of control references and the effect of AIMD parameters.

The remainder of the paper is organized as follows. In Section II, we give an overview of EDCA that pertains to the problem considered in the paper. We also summarize the results from our prior work [10]. In Section III, we give a taxonomy of existing approaches to maximizing bandwidth efficiency and achieving proportional bandwidth allocation. In particular, we discuss why some of the on-line measurement mechanisms fail to converge. Following that, we elaborate on our proposed MCC approach in Section IV and evaluate its performance in Section V. Finally, we conclude the paper in Section VI with several research avenues for future study.

[1] If a WLAN operates in the infrastructure mode, the measurement and feedback procedures can be performed by the Access Point.

II. Background Material

A. Overview of EDCA

As mentioned in Section I, IEEE 802.11e defines a new HCF coordination function that includes a contention-based channel access method, called EDCA, as an extension to DCF. Nodes contend for medium access in essentially the same manner as they do under DCF. However, several parameters that control how and when a node gains access to the medium differ among different ACs, 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, flows in different ACs wait for different values of *Arbitration Interframe Space (AIFS)*. Different values of CW_{min} and CW_{max} are associated with different ACs, which enforces traffic of different priority levels 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 the right to exclusively use the medium for a period of time interval specified by TXOP. The duration of the TXOP is specified per AC, with larger (smaller) values assigned to high-priority (low-priority) traffic.

B. Criteria for Maximizing Bandwidth Utilization and Achieving Proportional Bandwidth Allocation in EDCA

We have conducted a comprehensive analytical study of EDCA in [10] and [9], and analyzed its capacity of providing differentiated services by means of of tuning CW and AIFS values among different ACs in a single-cell wireless network. The analysis is made under the *ideal channel* assumption, i.e., the channel does not introduce any other than those induced by collisions. We show that the effect of tuning AIFS values is quite substantial, i.e., high-priority flows (assigned with smaller AIFS values) can easily grab most of the bandwidth and starve low-priority flows. Tuning CW values among different ACs, on the other hand, is shown to be an effective and better-controllable means of providing proportional bandwidth allocation. Specifically, let r_j be defined as the ratio of the attainable throughput by a class-j node to that by a class-1 node. $r_1 \equiv 1$. We have the following major result [9].

Theorem 1: To achieve *proportional bandwidth allocation* (with respect to r_j) and to maximize *bandwidth utilization* in a network of M traffic classes (each configured with the same AIFS value), the CW of each class should be set as follows:

$$CW_j^* = \frac{\sqrt{2\beta T_D'}}{r_j} + 1, \qquad (1)$$

where $\beta = \left(\sum_{j=1}^{M} N_j\, r_j\right)^2 - \sum_{j=1}^{M} N_j\, r_j^2$. $T_D' = T_D/t_s$. Note that t_s is the length of a time slot, e.g., $20\,\mu secs$ in 802.11b, and T_D is the duration of a successful transmission, i.e., T_D =

DATA+SIFS+ACK+DIFS. N_j is the number of users in class j.
Eq. (1) suggests that to achieve both objectives, one can set $CW_1 = \sqrt{2\,\beta\,T'_D} + 1$ and enforce the relation between CW$_j$ and CW$_1$ as $CW_j - 1 = (CW_1 - 1)/r_j$.

C. Notations

To facilitate subsequent derivation, in this subsection we define several terms and give their corresponding expressions derived in [10]. It has been shown in [10] that when the network operates under the condition of light medium access (i.e., when the traffic load is light or the contention window size is large), medium access can be modeled in a p-persistent fashion. That is, nodes in class j transmit in any slot *independently* and *uniformly* with the attempt probability $\tau_j = \frac{2}{CW_j+1}$. Let the probability that a slot is *idle*, *a successful transmission* or *a collision* be denoted as P_I, P_S, and P_C, respectively. They can be readily derived as follows:

$$P_I = \prod_{j=1}^{M}(1-\tau_j)^{N_j} \triangleq A, \tag{2}$$

$$P_S = \sum_{j=1}^{M} N_j \frac{\tau_j}{1-\tau_j} A, \text{ and } P_C = 1 - P_I - P_S. \tag{3}$$

Recall hat Eq. (1) suggests that in order to achieve proportional bandwidth allocation r_j, it is necessary that $CW_j = (CW_1 - 1)/r_j + 1$. Let $x \triangleq \frac{\tau_1}{1-\tau_1}$. P_S can be written as:

$$P_S = \sum_{j=1}^{M} N_j\, r_j\, x\, A. \tag{4}$$

The function A^{-1} can be approximated using the second-order Taylor series as

$$(A)^{-1}(x) = \prod_{j=1}^{M}(1 + r_j\,x)^{N_j} \approx 1 + \theta\,x + \frac{1}{2}\beta\,x^2, \tag{5}$$

where $\theta = \sum_{j=1}^{M} N_j\, r_j$ and $\beta = \theta^2 - \sum_{j=1}^{M} N_j\, r_j^2$. We will use Eq. (2)-(5) in the following two sections.

III. A Taxonomy of Existing Approaches to Maximizing Utilization and/or Achieving Proportional Bandwidth Allocation

To maximize the bandwidth utilization and achieve proportional bandwidth allocation, most of the existing approaches derive the optimal value of certain protocol parameter (e.g., the CW size) for each class. For example, Theorem 1 gives the optimal value of CW$_j$ for each class j. Because the optimal value of a protocol parameter for each class is very often a function of both the total number of classes and the number of competing nodes in each class, it is necessary that each node acquires these pieces of information and calculates the optimal value accordingly. However, such information is quite difficult to acquire, and the common practice is to devise an on-line measurement algorithm to measure certain channel state, infer the required information from the measured channel state, and drive the network toward the optimal state. In what follows, we categorize several existing approaches, including those that only consider to maximize the bandwidth utilization.

$Idle_k$	The k-th sample/measurement of $Idle$.
$E[Idle]_k$	The k-th estimate of $E[Idle]$.
N	Actual value of the number of competing nodes.
$\widetilde{N}_k$	The k-th "sample" of N computed from $(E[Idle]_k, \tau_{k-1})$ according to Eq. (6).
$\widehat{N}_k$	The k-th estimate of N.
α	Parameter for the moving average algorithm.
CW_k	Optimal CW computed using $\widehat{N}_k$ according to Eq. (1).
τ_k	Attempt probability, $= \frac{2}{CW_k+1}$.

TABLE I
NOTATIONS USED IN FIG. 1

A. Approaches That Set CW to the Optimal Value

By observing certain channel state, it is possible to estimate the number of competing nodes and compute the optimal value of CW according to, for example, Eq. (1). State variables that are frequently used are, for example, *the number of consecutive idle slots between two busy periods* (denoted by $Idle$), and the *conditional collision probability* p_{cc}, defined as the probability that a node's transmission attempt results in a collision.

In the case of existence of a single traffic class (where EDCA essentially reduces to DCF), Bianchi et al. [3] use an extended Kalman Filter to estimate the number of competing nodes by measuring p_{cc}. Qiao et al. [14] consider the case of multiple traffic classes and propose to estimate the number of competing nodes in each class by observing $Idle$ and avg_wait_i, where avg_wait_i is the average number of slots between two consecutive successful class-i frame transmissions. Nodes make the estimation in a distributed manner and set their own CWs to the optimal values.

No matter which on-line measurement and estimation algorithm is used, it is important that the network converges to the desirable operating point. In what follows, we show that this may not be the case, even in the presence of only a single traffic class. Without loss of generality, we use $Idle$ to estimate the number of competing nodes, N. It is well known that $Idle$ is geometrically distributed with parameter $1 - P_I$, where $P_I = (1-\tau)^N$ according to Eq. (2). Therefore, $E[Idle] = \frac{(1-\tau)^N}{1-(1-\tau)^N}$, and N can be derived as

$$N = \frac{log(\frac{E[Idle]}{E[Idle]+1})}{log(1-\tau)} \triangleq f(E[Idle], \tau). \tag{6}$$

A typical adaptive algorithm for setting the optimal CW size is given in Fig. 1, with the terms used given in Table I. Note that the subscripts indicate the sequence of measurements and estimates. In the following proposition, we show that the algorithm given in Fig. 1 does not converge.

Proposition 1: The algorithm given in Fig. 1 does not converge.

/* Execute upon the end of each idle period.
* When the $(k+1)$-th $Idle$ measurement $(Idle_{k+1})$ is made:
*/
Begin
1. Update the estimate of E[Idle]:
   $E[Idle]_{k+1} = \alpha E[Idle]_k + (1-\alpha) Idle_{k+1}$.
2. Compute $\widetilde{N}_{k+1} = f(E[Idle]_{k+1}, \tau_k)$ according to Eq. (6).
3. Update the estimate of N:
   $\widehat{N}_{k+1} = \alpha \widehat{N}_k + (1-\alpha)\widetilde{N}_{k+1}$
4. Compute the optimal CW:
   $CW_{k+1} = \sqrt{2T'_D(\widehat{N}^2_{k+1} - \widehat{N}_{k+1})} + 1.$
5. Set the current contention window to CW_{k+1}.
   Update $\tau_{k+1} = \frac{2}{CW_{k+1}+1}$

End

Fig. 1. Pseudocode of the adaptive algorithm for setting the optimal CW.

Proof: By Step 3 in Fig. 1 we have

$$\widehat{N}_{k+1} - \widehat{N}_k = \alpha(\widehat{N}_k - \widehat{N}_{k-1}) + (1-\alpha)(\widetilde{N}_{k+1} - \widetilde{N}_k). \quad (7)$$

By Eq. (6), we have

$$\widetilde{N}_{k+1} - \widetilde{N}_k = \frac{log \frac{E[Idle]_{k+1}}{1+E[Idle]_{k+1}}}{log(1-\tau_k)} - \frac{log \frac{E[Idle]_k}{1+E[Idle]_k}}{log(1-\tau_{k-1})}. \quad (8)$$

It is reasonable to assume $E[Idle]_{k+1} \approx E[Idle]_k$ based on the following observation: In the k-th period, a node sets its current contention window to CW_k (Step 5). However, this may not necessarily change the current contention window size. The new contention window size takes effect *only* when a node completes transmission of a frame and and attempt for the next transmission. In general, $E[Idle]_k$ changes much slower than CW_k, and the larger N is, the slower the change in $E[Idle]_k$. Based on this argument, Eq. (8) can be re-written as

$$\widetilde{N}_{k+1} - \widetilde{N}_k = \frac{log\left(1 + E[Idle]_k{}^{-1}\right)}{log(1-\tau_k)\, log(1-\tau_{k-1})} \times log \frac{1-\tau_k}{1-\tau_{k-1}}. \quad (9)$$

If $\widehat{N}_k - \widehat{N}_{k-1} > 0$, then by Steps 4 and 5, $\tau_k < \tau_{k-1}$. By Eq. (9), $\tau_k < \tau_{k-1}$ implies $\widetilde{N}_{k+1} - \widetilde{N}_k > 0$. By Eq. (7), we have $\widehat{N}_{k+1} - \widehat{N}_k > 0$. Similarly, if $\widehat{N}_k - \widehat{N}_{k-1} < 0$, then $\widehat{N}_{k+1} - \widehat{N}_k < 0$. It is not difficult to see that even if the algorithm starts with $\widehat{N}_0 = N$, a small disturbance in the estimate of N may cause $\widehat{N}_k$ to continuously increase or decrease. ∎

The proposition has been corroborated by our simulation study under a wide variety of network scenarios. (The simulation setup will be described in Section V.) For demonstration purposes, we depict in Fig. 2 the CW value set by node 7 in a 10-node network with $N = 10$, $cw_0 = 31$, and $\alpha = 0.95$.

B. Approaches That Dynamically Tune CW

The major reason why the above approach diverges is because it uses open loop control. Setting an optimal state as the target and exercising closed-loop control to drive the network the the optimal state is the key idea behind approaches that *dynamically tune CWs*. For example, Calí et al. [4] identify that a balance between the bandwidth waste due to the idle period (as a result of all the nodes backing off) and the bandwidth waste due to collision (as a result of multiple nodes attempting for transmission) can be achieved, when the idle period is

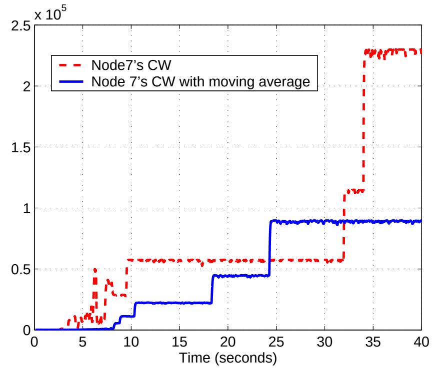


Fig. 2. An example that shows that the algorithm given in Fig.1 does not converge. $N = 10$, $\widehat{N}_0 = 10$, $CW_0 = 31$, and $\alpha = 0.95$. The blue curve corresponds to the CW value obtained by using the moving average to update CW_{k+1} in Step 4. This does not stop the growing trend of node 7's CW. Under other scenarios, we have also observed that CWs diverge in the decreasing direction.

equal to the collision period. They then tune the CW to drive the network toward achieving the balance. Heusse et al. [8] adopt an AIMD algorithm to tune the CW, so that the length of the idle period can be kept at a desirable level. Yang and Kravets [17] use a utility-based approach to tune the CW so as to achieve high bandwidth utilization and QoS performance specified by the utility function. The difficulty of employing the utility-based approach is that without the knowledge of the number of nodes, the parameters needed in the utility function may not be appropriately set.

C. Approaches That Schedule Frame Transmissions

Approaches in this category adjust, based on the measurement of the system state(s), the rate of injecting packets for channel access. Kim and Hou [12] propose such an approach for IEEE 802.11-operated WLANs, called *model-based frame scheduling (MFS)*. An additional delay is introduced, if necessary, before passing packets to the MAC access function. However, similarly to the first category of approaches, the delay is calculated based on the on-line estimate of the number of competing nodes, and hence the approach suffers from the same diverging problem.

IV. Proposed MAC-Centric Contention Control Approach

MCC is devised based on a novel interpretation of the optimality condition given in Theorem 1. Recall that Theorem 1 gives the optimal value of CW_j, assuming the prior knowledge of the number of classes M and the number of competing nodes in each class N_j. If we can control N_j to satisfy the optimal condition for class-1: $CW_1 = \sqrt{2\beta T'_D} + 1$, and maintain the relation $CW_j - 1 = (CW_1 - 1)/r_j$ required for proportional bandwidth allocation, then we can achieve both objectives *without* tuning $\{CW_j, j = 1 \cdots M\}$.

At the first glimpse, N_j, $j = 1 \cdots M$ does not seem to be tunable. However, it is important to understand that the optimality condition (Eq. (1)) is derived assuming all the nodes are back-logged (i.e., the asymptotic condition holds). If we can control the rate λ_j at which packets are passed down to the MAC access function, we can effectively control the number of competing nodes as $N'_j = N_j \lambda_j$. There are several advantages associated with bypassing the operation of tuning the contention windows. First, this simplifies the design of the MAC access function. Although the access function in IEEE 802.11 has incorporated the binary exponential backoff mechanism for collision resolution, it has been well recognized that this backoff mechanism fails to maximize the bandwidth utilization. Incorporating a complicated mechanism for tuning the contention windows will overload the design of the access function. Instead, laying a *thin* MCC layer on top of the MAC access function decouples medium access from contention control. Second, considering the fact that in most, if not all, wireless chipsets, the MAC access function is implemented in firmware (and hence the contention window size cannot be readily modified), MCC is more promising to be practically deployed.

To realize MCC, we need to determine control references and their corresponding optimal values. Control references should be readily measured, and their optimal values (defined as *the optimal reference level*) should not change dramatically with network dynamics such as the number of competing nodes and their traffic loads. In the next subsection, we identify, by studying the optimality condition derived in [10], two control references that remain approximately constant when the network operates at the optimal operational point, *regardless of the number of competing nodes in each AC.*

A. Determination of Control References

Control references N_c and $Idle$: We consider two control references that can be readily measured by individual nodes: the number of collisions between two consecutive successful transmissions N_c, and the number of consecutive idle slots $Idle$ (as defined in Section III). As discussed in Section II, if the network is stochastically stable, we can treat both N_c and $Idle$ as random variables of the geometric distribution with parameters $\frac{P_S}{1-P_I}$ and $1 - P_I$, respectively. We are interested in their means $E[.]$ and coefficients of variation $CV[.]$. Recall that for a random variable y of the geometric distribution with parameter p, we have $E[y] = p^{-1}(1-p)$, $Var[y] = p^{-2}(1-p)$, and $CV[y] \triangleq \sqrt{Var(y)}/E[y] = (1-p)^{-\frac{1}{2}}$. Hence we have

$$E[N_c] = \frac{1-P_I}{P_S} - 1, \quad CV[N_c] = \left(\frac{1-P_I}{P_S}\right)^{\frac{1}{2}}, \tag{10}$$

$$E[Idle] = \frac{P_I}{1-P_I}, \qquad CV[Idle] = {P_I}^{-\frac{1}{2}}. \tag{11}$$

Optimal reference levels of N_c and $Idle$: The optimal reference level refers to the value of $E[N_c]$ or $E[Idle]$ when the network operates at the optimal operational point. By applying Eqs. (2)–(5) to Eq. (10), we have $E[N_c] = \frac{1}{2}\frac{\beta}{\theta}\, x$. Recall that $x \triangleq \frac{\tau_1}{1-\tau_1}$. By the optimality condition Eq. (1), we obtain $x^* = \sqrt{\frac{2}{\beta T'_D}}$. Thus the optimal reference level for $E[N_c]$ can be expressed as

$$E[N_c]^* = \frac{1}{\sqrt{2\,T'_D}}\,(1-\gamma)^{\frac{1}{2}}, \tag{12}$$

where $\gamma = \frac{\sum_{j=1}^{M} N_j\, {r_j}^2}{\left(\sum_{j=1}^{M} N_j\, r_j\right)^2} \in [0, 1]$.

Similarly, the optimal reference level for $E[Idle]$ can be expressed as

$$E[Idle]^* = \frac{T'_D}{1+\sqrt{2T'_D}\,(1+\gamma)^{-\frac{1}{2}}}. \tag{13}$$

Note that $E[N_c]^*$ and $E[Idle]^*$ are determined by T'_D and γ. T'_D depends on the PHY/MAC characteristics such as the slot length t_s, interframe spaces (SIFS and DIFS), and the DATA/ACK frame size, but less on network dynamics such as the number of competing users N_j. T'_D can be obtained by runtime measurement. γ, on the other hand, depends largely on network dynamics, N_j and r_j. Fortunately, γ is bounded between 0 and 1. This results in

$$0 \le E[N_c]^* \le \frac{1}{\sqrt{2\,T'_D}}\,, \tag{14}$$

$$\frac{T'_D}{1+\sqrt{2T'_D}} \le E[Idle]^* \le \frac{T'_D}{1+\sqrt{T'_D}}. \tag{15}$$

Table II gives the ranges of $E[N_c]^*$ and $E[Idle]^*$ under the IEEE 802.11b DSSS PHY/MAC configuration with data rates of 2/5.5/11 Mbps and payload of 1460 bytes and 512 bytes, respectively. Figure 3 depicts $E[N_c]^*$ and $E[Idle]^*$ obtained from both analytical results (Eqs. (12)-(13)) and simulation. As shown in Fig. 3, both $E[N_c]^*$ and $E[Idle]^*$ do not vary significantly as the number of competing nodes in each class increases. This is the desirable feature needed for control references.

Dynamics of N_c and $Idle$: $CV[N_c]$ and $CV[Idle]$ reflect the dynamics of N_c and $Idle$. The smaller $CV[N_c]$ or $CV[Idle]$, the better N_c or $Idle$ serves as a control reference. Figure 4 depicts the values of $CV[N_c]$ and $CV[Idle]$ (denoted as $CV[N_c]^*$ and $CV[Idle]^*$) when the network operates in the optimal state. $CV[N_c]^*$ is approximately four times $CV[Idle]^*$. This implies that $E[Idle]$ is a better candidate as the control reference for MCC. This will be corroborated by the simulation study in Section V.

B. Detailed Description of MCC Procedures

Positioned between the interface queues and the MAC access function, the proposed MCC approach falls in the third category of approaches in Section III-C. Instead of dynamically tuning the CW, MCC controls the rate of dequeuing packets to the MAC access function. MCC is composed of three procedures: *measure the control reference*, *generate feedback by comparing the measured result against the range of the optimal reference level*, and *control the rate of dequeuing packets according to an AIMD algorithm and the bandwidth allocation ratio associated with the traffic class.* All the nodes execute the three procedures independently and in a distributed manner. In what follows, we elaborate on each procedure.

Measurement: Each node monitors the transmission activities on the channel and measures two control references, N_c and $Idle$. (Note that if the feedback is generated based on the estimate of $E[N_c]$ ($E[Idle]$), then it is not necessary to measure

	payload = 1460 bytes			payload = 512 bytes		
Data rate	T'_D	$E[N_c]^*$	$E[Idle]^*$	T'_D	$E[N_c]^*$	$E[Idle]^*$
2 Mbps	326.2	$[0, 0.039]$	$[12.3, 17.1]$	136.6	$[0, 0.061]$	$[7.8, 10.8]$
5.5 Mbps	136.31	$[0, 0.061]$	$[7.8, 10.8]$	67.36	$[0, 0.087]$	$[5.3, 7.3]$
11 Mbps	82.1	$[0, 0.078]$	$[5.9, 8.2]$	47.6	$[0, 0.10]$	$[4.4, 6.0]$

TABLE II

LOWER AND UPPER BOUNDS OF $E[N_c]^*$ AND $E[Idle]^*$ UNDER THE IEEE 802.11B DSSS PHY/MAC CONFIGURATION.

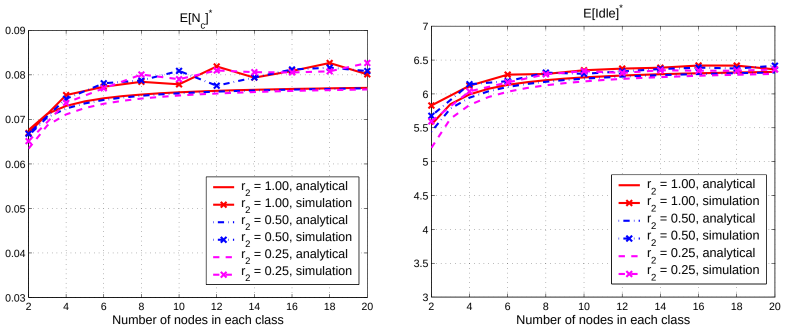


Fig. 3. Analytical and simulation results of $E[N_c]^*$ and $E[Idle]^*$ in IEEE 802.11e using 802.11b DSSS PHY parameters. DATA payload = 1460 bytes.

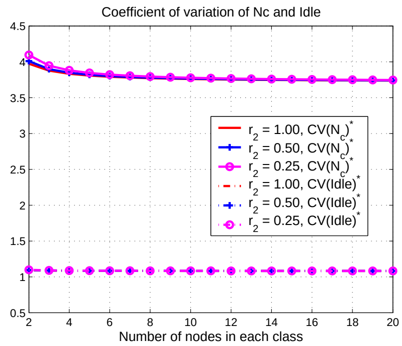


Fig. 4. Coefficients of variation of N_c and $Idle$ when the network operates in the optimal state.

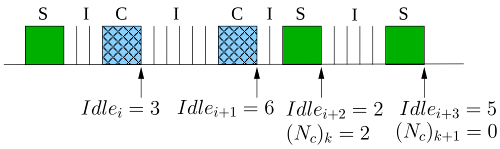


Fig. 5. How $Idle$ and N_c are measured in MCC. S – Successful transmissions, including DATA+SIFS+ACK+DIFS. I – Idle slots. C – Collisions, including DATA+send timeout+EIFS.

$E[Idle]$ ($E[N_c]$).) As shown in Fig. 5, a node counts the number of idle slots between two consecutive busy periods and generates a new record, $Idle_i$, at the end of the i-th busy period. Similarly, a node counts the number of collisions between two consecutive successful transmissions and generates a new record, $(N_c)_j$, at the end of the j-th successful transmission. At the time of generating a new record, the estimate is updated using the moving average, i.e.,

$$E[Idle]_i \leftarrow \alpha\, E[Idle]_{i-1} + (1-\alpha)\, Idle_i, \quad (16)$$

$$E[N_c]_k \leftarrow \alpha\, E[N_c]_{k-1} + (1-\alpha)(N_c)_k. \quad (17)$$

Feedback: Based on the difference between the estimates of $E[Idle]$ or $E[N_c]$ and their reference levels, each node generates the feedback. Recall that in Section IV-A we have derived the bounds of $E[Idle]^*$ and $E[N_c]^*$. Rather than setting the optimal control reference level to a fixed value, we use a range of the form $(\,E[.]_{min}, E[.]_{max}\,)$ to accommodate network dynamics. Specifically, if $E[Idle]$ is used as the control reference, we set its range according to Eq. (15). If $E[N_c]$ is used as the control reference, we set its reference range to $\left(0.5/\sqrt{2T'_D}, 1/\sqrt{2T'_D}\right)$. We will evaluate the performance of MCC with respect to the choice of optimal reference level ranges in Section V.

If $E[Idle]$ is used as the control reference, each node generates the feedback at the end of each successful transmission using the following procedure:

If $E[Idle] < E[Idle]^*_{min}$, then
  feedback = '+'; // medium is over-utilized
Else if $E[Idle] > E[Idle]^*_{max}$, then
  feedback = '-'; // medium is under-utilized.
End (18)

Control: This procedure involves two steps: determining the dequeuing rate and scheduling. The dequeuing rate, λ, is determined by the interval, d, between two dequeuing actions, i.e., $d = \lambda^{-1} DATA$, where $DATA$ is the data packet size. After each successful transmission, each node uses an AIMD algorithm to increase d (in case the feedback is '+'), to decrease d (in case the feedback is '-'), or keep d unchanged.

Specifically, let the multiplicative decrease parameter be denoted as σ and the additive increase parameter as $\epsilon\,\Delta$, where $0 < \sigma, \epsilon < 1$, $\Delta = DATA/\,(T_D + t_s\, E[Idle]^*)$, and $E[Idle]^*$ is set to the medium of its bounds given in Eq. (13). At the k-th updating point, if the feedback indicates '+', we multiplicatively decrease the rate by $c \equiv (N_c)_k + 1$ times; otherwise, if the feedback indicates '-', we additively increase the rate by $\epsilon\,\Delta$:

$$\text{AI: } \lambda_k \rightarrow \lambda_{k-1} + \epsilon\,\Delta \quad \Leftrightarrow \quad d_k \leftarrow \frac{1}{d_{k-1}^{-1} + \epsilon\Delta/T_D}, \quad (19)$$

$$\text{MD: } \lambda_k \leftarrow \sigma^c \lambda_{k-1} \quad \Leftrightarrow \quad d_k \leftarrow \sigma^{-c} d_{k-1}. \quad (20)$$

The interval d is updated by all the nodes in the same manner. However, each node uses d/r_j as its packet dequeuing interval in order to ensure proportional bandwidth allocation. After the successful transmission of its packet, a node schedules to dequeue its next packet after an interval of d/r_j.

Summary: At first glimpse, it seems that MCC bears some similarity with approaches proposed in the second or third categories in Section III-C. We would like to stress that MCC differs from existing approaches in the second category in that it does not require tuning of contention windows, CW_j, $j = 1 \cdots M$ (which cannot not readily modified in most wireless devices). Although MCC belongs to the third category, it differs from MFS in that it considers both issues of maximizing the bandwidth utilization and achieving proportional bandwidth allocation, and it uses an AIMD algorithm (whose convergence is well established in [6]) to dynamically adjust the rate of passing packets down to the MAC access function. Moreover, MCC identifies two control references that *regardless of the number of competing nodes in each AC*, remain approximately constant when the network operates at the optimal operational point. By comparing measurement results to the properly chosen range of optimal control reference levels, MCC then generates feedback accordingly. This ensures MCC is more robust to network dynamics.

V. Performance Evaluation

A. Simulation setup

We have implemented the proposed MCC mechanism in *ns-2.27*, and evaluated its performance in an IEEE 802.11b (DSSS)–operated WLAN with CBR traffic. The data rate used is 11Mbps. Unless specified otherwise, the data payload is set to 1460 bytes. Due to the space limit, we only report cases in which data transmissions are made without the RTS/CTS handshake mechanism. All the simulation runs last for 100 seconds.

We compare IEEE 802.11 MAC (or 802.11e in case of multiple classes) without and with MCC, labeled as *802.11* (or *802.11e*), and *802.11+MCC* (or *802.11e+MCC*), respectively. If $E[Idle]$ is used as the control reference, the range of $E[Idle]^*$ is chosen to be $[5.5, 8.0]$ (in compliance with Eq. (15)). If $E[N_c]$ is used as the control reference, the range of $E[N_c]^*$ is chosen to be $[0.04, 0.09]$.

Performance metrics used are: (1) *bandwidth utilization* (measured by the *throughput* normalized to the data rate), *idle period length* $E[Idle]$, and *collision cost* $E[N_c]$; and (2) *QoS performance* measured by the *throughput ratio*. We also investigate how sensitive MCC is to its tunable parameters and their effects on aggregate throughput and responsiveness to network dynamics.

B. Throughput and QoS performance

Single class with persistent traffic: In this configuration, all the nodes are of the same class and are back-logged. Figure 6(a), (b), and (c) give the aggregate throughput, the idle length ($E[Idle]$) and the collision cost ($E[N_c]$), respectively. MCC ($Idle$-based) achieves the highest throughput, and outperforms 802.11 by as much as 26% when $N = 60$. The throughput achieved by MCC (N_c-based) is slightly lower than that by MCC ($Idle$-based). This is because 802.11 incurs the largest number of collisions ((c)) and thus attains the lowest throughput. MCC (N_c-based) incurs the least number of collisions ((c)), but spends more time in the idle state than MCC ($Idle$-based) ((b)). Because in general MCC ($Idle$-based) performs better than MCC (N_c-based), we henceforth only consider MCC ($Idle$-based).

Three classes with persistent traffic: In this configuration, all three classes have the same number of nodes. The desired bandwidth allocation ratio R_2 and R_3 are set to, respectively, $R_2 = 0.5$ and $R_3 = 0.25$.

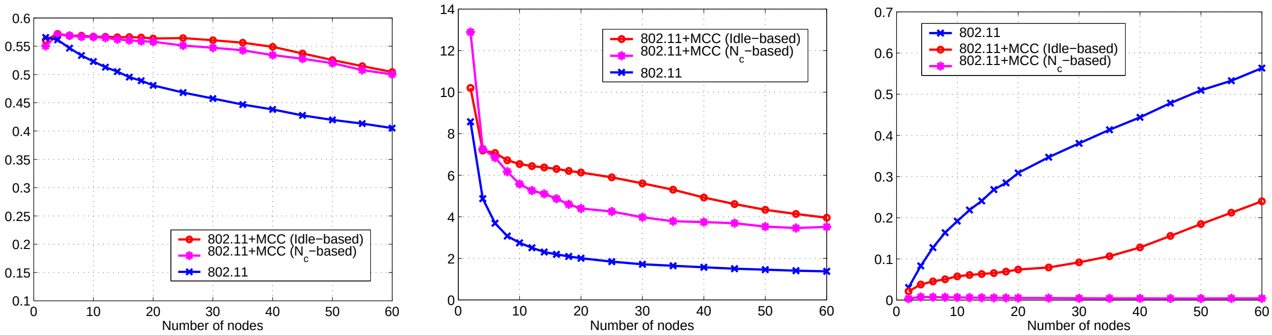
Figure 7(a) and (b) respectively give the total throughput and the throughput ratio r_2 and r_3 achieved by 802.11e and 802.11e+MCC. Again 802.11e+MCC achieves $5 - 129\%$ higher throughput than 802.11e. Moreover, 802.11e fails to achieve proportional bandwidth allocation at the desired ratio. When the number of nodes in each class increases, r_3 in 802.11e approaches 0.45, while 802.11e+MCC is able to keep both r_2 and r_3 at their desirable levels.

Two classes with on-off traffic: In this configuration, there are two classes of nodes, each with 20 nodes. While class-1 nodes generate persistent traffic, class-2 nodes generate CBR traffic in an on-off fashion with both on and off durations set to 20 seconds. The first on time is [0, 20] seconds.

Figure 8(a) and (b) respectively give the histogram of the aggregate throughput and the throughput ratio r_2. 802.11e+MCC always achieves higher throughput and better proportional bandwidth allocation at the desired level $R_2 = 0.5$. In addition, two observations are in order: (1) When the on periods start at 20 and 40 seconds, respectively, the total throughput achieved by 802.11e+MCC first drops rapidly, and then goes back to approximately the same level. It reflects responsiveness of 802.11e+MCC to the network dynamics. We will study this further in Section V-D. (2) The off periods end at 40 and 80 seconds, respectively. However, the throughput achieved by 802.11e does not catch up until 10 seconds after the off periods. This is because the interface queues have buffered a large number of packets in the on periods. This phenomenon is less severe in 802.11e+MCC, because 802.11e+MCC is able to deliver more packets in a unit of time.

C. Sensitivity to the choice of $E[Idle]^*$

In all the above cases, the reference level range of $E[Idle]^*$ is chosen to be $[5.5, 8.0]$. In this section, we study the impact of varying the reference level range on the MCC performance. We use three sets of reference ranges: $[5.5, 5.5]$, $[5.5, 8.0]$, and $[8.0, 8.0]$, and two different data payloads: 1460 and 512 bytes. Figure 9 shows the performance of MCC when there is only one class of nodes present. In general, the throughput achieved by MCC with different reference level ranges are close to each other, and MCC that uses a range of control reference levels (e.g., $[5.5, 8.0]$) outperforms those that use a fixed control reference level.

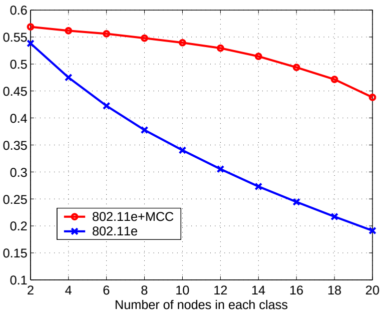


(a) Throughput (normalized)

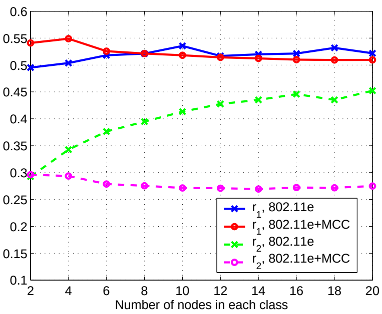
(b) Idle length: $E[Idle]$

(c) Collision cost: $E[N_c]$

Fig. 6. **Case 1: Single class with persistent traffic.** For all nodes, $CW = [32, 1024]$, i.e., $CW_{\text{min}} = 32$ and $CW_{\text{max}} = 1024$.

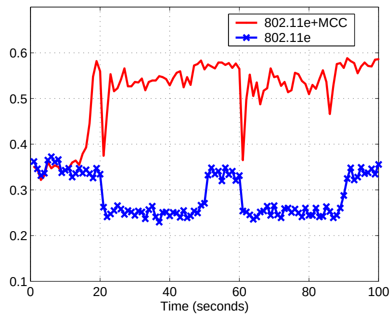


(a) Total throughput (normalized)

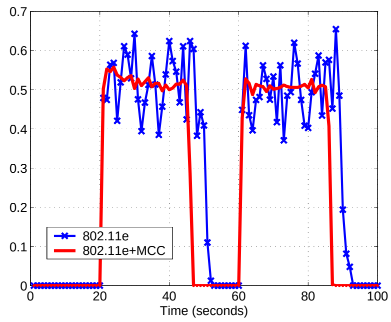


(b) Throughput ratios: r_1 and r_2

Fig. 7. **Case 2: Three classes with persistent traffic.** The desired throughput ratio is set to $R_2 = 0.5$ and $R_3 = 0.25$. For class-1 nodes, $CW_1 = [16, 48]$. For class-2 nodes, $CW_2 = [31, 93]$. For class-3 nodes, $CW_3 = [61, 183]$.

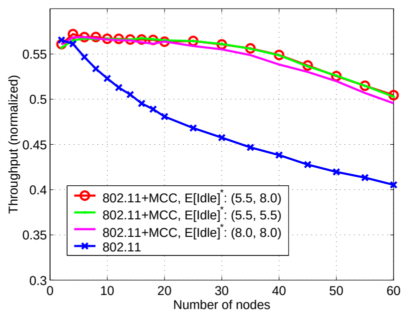


(a) Total throughput (normalized)

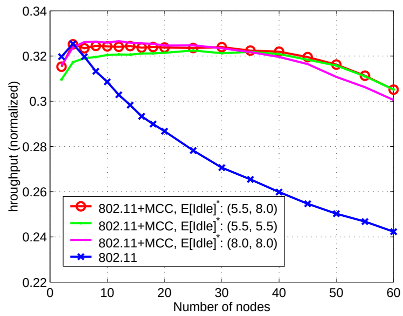


(b) Throughput ratio: r_1

Fig. 8. **Case 3: Two classes with class-2 generating on-off traffic.** The on periods are [20, 40] and [60, 80] seconds. Each class has 20 nodes. When class-2 nodes are active, the desired throughput ratio is $R_2 = 0.5$.



(a) DATA payload = 1460 bytes.



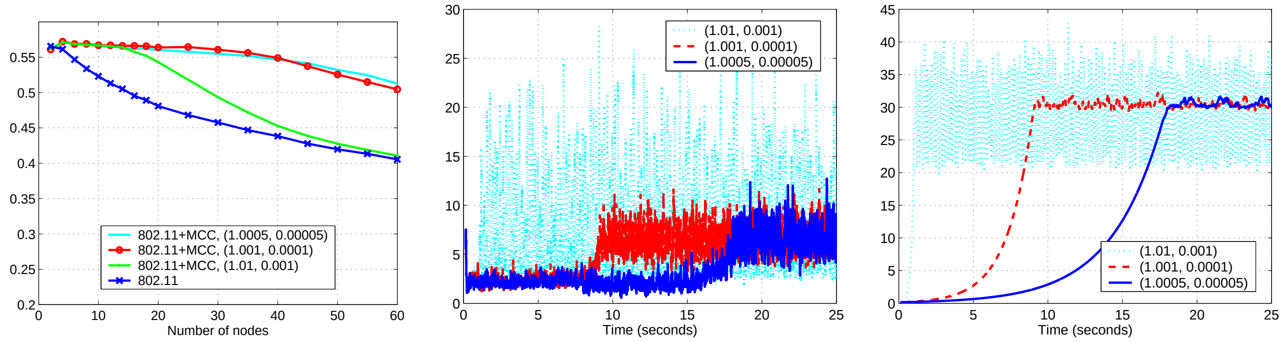
(b) DATA payload = 512 bytes.

Fig. 9. Performance of MCC with different reference level ranges of $E[Idle]^*$. There is only one class of nodes in the network.

D. Choice of the AIMD algorithm parameters

In this section, we study the impact of AIMD parameters, σ for AI and ϵ for MD, on the MCC performance. We compare MCC with three different sets of (σ^{-1}, ϵ): (1.01, 0.001), (1.001, 0.0001) – the one used in all previous simulation studies – and (1.0005, 0.00005). For notational convenience, we refer to them as MCC-AIMD-1, MCC-AIMD-2, and MCC-AIMD-3, respectively. Figure 10(a) gives the aggregate throughput attained by all the nodes in a network with one class of nodes, while Fig. 10 (b) and (c) give, respectively, the histogram of $E[Idle]$ and dequeuing interval d at a randomly chosen node (node 1 in this case) in a network of 20 nodes.

As shown in Fig. 10, MCC-AIMD-1 attains less throughput than MCC-AIMD-2 and MCC-AIMD-3. The performance degradation becomes more obvious as N increases. Correspondingly, we observe in (b) and (c) that MCC-AIMD-1 exhibits more dramatic oscillation in $E[Idle]$ and d. The advantage of MCC-AIMD-1, however, is that it converges to the equilibrium as shown in (c) — it only takes approximately 2 seconds for d to reach the desired operating range. In contrast, MCC-AIMD-2 and MCC-AIMD-3 attain larger throughput, exhibit small oscillation in $E[Idle]$ and d, but take more time to converge and respond to network dynamics. In summary, small AIMD parameters are preferred for networks with a large number of competing nodes for the advantage of larger



(a) Throughput (normalized). (b) Measured $E[Idle]$. (c) Dequeuing interval (ms).

Fig. 10. Performance of MCC with different AIMD parameters (σ^{-1}, ϵ). There is only one class of nodes present. The total simulation time is 100 seconds, but for better visual effect, we show in (b) and (c) only results in [0, 25] seconds.

throughput; however, MCC with small AIMD parameters is less responsive to network dynamics.

VI. Conclusion

We have proposed a MAC-centric contention control (MCC) mechanism for maximizing bandwidth utilization and achieve proportional bandwidth allocation among different classes. Positioned between the interface queue and the MAC access function, MCC operates by having each node (1) measure at runtime bandwidth the control reference (e.g., the number of collisions between two consecutive successful transmissions N_c or the number of consecutive idle slots $Idle$, (2) generate feedback by comparing the measured results to the desired range of optimal control level, and (3) adjust the rate of dequeuing packets from the interface queues for medium access in an additive-increase-multiplicative-decrease fashion. MCC has the following advantages: (1) it decouples contention control and medium access and thus simplifies the design; (2) without tuning the contention window size, it is more practical for MCC to be deployed as an "add-on" component in existing wireless hardware and firmware; and (3) it identifies two control references, N_c and $Idle$, that *regardless of the number of competing nodes in each AC*, remain approximately constant when the network operates at the optimal operational point. This ensures the robustness of MCC to network dynamics.

We have conducted an extensive simulation study under various network scenarios (single or multiple traffic classes with varying prespecified bandwidth allocation ratios, persistent or on-off traffic, and different data payloads). The proposed MCC mechanism outperforms conventional 802.11/802.11e significantly in terms of both the aggregate throughput and the capability of meeting the prespecified bandwidth allocation ratio. The simulation study has also shown that MCC is robust to the choice of the reference level range, but is more sensitive to the choice of AIMD parameters. Small values of AIMD parameters are preferred in the case of a large number of competing nodes, but are less responsive to network dynamics. Devising an adaptive AIMD algorithm that adapts its parameters in response to the network status is part of our future research. We are also in the process of implementing MCC in the link control layer of the device driver, and conducting an empirical study in real networking environments.

References

[1] 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.

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

[3] G. Bianchi and I. Tinnirello. Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network. In *Proc. of IEEE INFOCOM*, April 2003.

[4] 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.

[5] M. M. Carvalho and J. J. Garcia-Luna-Aceves. Delay analysis of IEEE 802.11 in single-hop networks. In *Proc. of IEEE ICNP*, Nov. 2003.

[6] D. Chiu and R. Jain. Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN*, 17(1):1–14, June 1989.

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

[8] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs,.

[9] C. Hu and J. C. Hou. A comprehensive study of the Enhanced Distributed Control Access (EDCA) function. In *Tech. Rep. No. UIUCDCS-R-2006-2711. http://lion.cs.uiuc.edu/∼chunyuhu*, April 2006.

[10] C. Hu, H. Kim, J. C. Hou, D. Chi, and S. S. Nandagopalan. Provisioning quality controlled medium access in ultrawideband WPANs. In *Proc. of IEEE INFOCOM*, April 2006.

[11] H. Kim and J. C. Hou. Improving protocol capacity with model-based frame scheduling in IEEE 802.11-operated WLANs. In *Proc. of MOBICOM*, 2003.

[12] H. Kim and J. C. Hou. Improving protocol capacity with model-based frame scheduling in IEEE 802.11-operated WLANs. *IEEE JSAC*, 22(10):1987–2003, 2004. Special issue on design, implementation and analysis of communication protocols.

[13] Z. Kong, D. H. Tsang, B. Bensaou, and D. Gao. Performance analysis of IEEE 802.11e contention-based channel access. *IEEE JSAC*, 22(10), 2004.

[14] D. Qiao and K. Shin. Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF. In *Proc. of International Workshop on Quality of Service*, May 2002.

[15] J. W. Robinson and T. S. Randhawa. Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function. *IEEE JSAC*, 22(5), 2004.

[16] Y. Xiao. An analysis for differentiated service in IEEE 802.11 and IEEE 802.11e wireless LANs. In *Proc. of IEEE ICDCS*, Mar. 2004.

[17] Y. Yang and R. Kravets. Thoughput guarantees for multi-priority traffic in ad hoc networks. In *Proc. of IEEE MASS*, 2005.