UMAV: A Simple Enhancement to the IEEE 802.11 DCF

Daji Qiao and Kang G. Shin
Real-Time Computing Laboratory
The University of Michigan
Email: {dqiao,kgshin}@eecs.umich.edu

Abstract— **In this paper, we propose a simple enhancement to the Distributed Coordination Function (DCF) of the IEEE 802.11 MAC protocol, called the *Unified Multiple Access with Variable contention window* (UMAV), to improve the throughput performance of the 802.11 DCF systems. The basic ideas of UMAV are that (1) a new P(polling)-mode is introduced for frame transmissions in addition to the conventional C(contention)-mode used in the DCF, so as to reduce the number of contending stations; and (2) an enhanced uniform backoff scheme is used to replace the binary exponential backoff scheme of the DCF, so as to achieve a low frame collision probability and maximize the channel utilization. Our in-depth simulation shows that UMAV outperforms the IEEE 802.11 DCF significantly in terms of both the aggregate system throughput and the average station access delay under various load conditions. Moreover, UMAV can react properly and quickly to the traffic load fluctuations, and achieve excellent fairness, a desirable byproduct.**

I. Introduction

The IEEE 802.11 standard [1] is the first international standard for wireless local-area networks (WLANs), and it has been widely used in most commercial WLAN products available in the market, e.g., the popular Agere ORiNOCO cards [2]. The IEEE 802.11 MAC specifies two different medium access control (MAC) mechanisms in WLANs: the contention-based Distributed Coordination Function (DCF) and the polling-based Point Coordination Function (PCF). At present, only the mandatory DCF is implemented in the 802.11-compliant products and one main problem with an 802.11 DCF system is its poor throughput performance, which becomes worse when the number of contending stations increases. This is because the DCF applies a slotted binary exponential backoff scheme to resolve collisions, which, unfortunately, does not perform well due mainly to its heuristic nature.

In order to increase the throughput performance of the IEEE 802.11 DCF systems, many enhanced backoff schemes [3][4] have been proposed to reduce the frame collision probability and the wasted idle backoff slots. In [3], a simple Markov chain model was presented to compute the saturation throughput of an 802.11 DCF system. From this paper, we have an important observation: the critical point of improving the system throughput is not how to design a set of new backoff rules, but how to adjust the parameters of a backoff scheme so that each station can be tuned to run at its optimal point. Assuming a uniform backoff scheme, the authors of [3] showed that the optimal contention window size (*cw_opt*) is actually a function of the number of contending stations. The authors of [4] analyzed the performance of an 802.11 DCF system from a different angle and proposed a similar idea to adjust the contention window size at each station based on the estimation of the number of active stations.

However, due to the very nature of DCF's contention-based access mechanism, there exists a theoretical limit that the aggregate throughput of an 802.11 DCF system is bounded by and can never be improved over. The authors of both [3] and [4] indicated such a theoretical upper bound of the IEEE 802.11 DCF capacity. Therefore, in order to further improve the system throughput, a *hybrid* protocol by mixing contention with polling is a must, because polling utilizes the wireless channel much more efficiently when the traffic load is heavy. The IEEE 802.11 Working Group (WG) recognized this problem and then included the polling-based PCF as part of the standard. Choi [5] showed that the PCF can indeed achieve a higher maximum throughput than the DCF when the system is overloaded. However, the original PCF has many open problems and has never been implemented in any 802.11-compliant products. Recently, a new Hybrid Coordination Function (HCF) [6][7] is being studied/proposed by the IEEE 802.11 Task Group E (TG-e) to replace the PCF. However, before the HCF is finalized and introduced to the market, the DCF-based products are expected to continue their dominance of the market. In this paper, we focus on how to improve the throughput performance of the existing 802.11 DCF systems, by proposing a simple enhancement to the DCF. We are not claiming that the proposed enhancement can, or will, replace the HCF. It is just a practical solution that is simple, effective, and easy to deploy with the available DCF devices.

Our scheme is called the *Unified Multiple Access with Variable contention window* (UMAV), and the key idea is to introduce a new P(polling)-mode for frame transmissions in addition to the C(contention)-mode used in the DCF. In UMAV, only the head-of-line frame of each data frame burst — in real-

The work reported in this paper was supported in part by AFOSR under Grant No. F49620-00-1-0327.

ity, data frames are often generated in bursts and queued at the wireless station before they are transmitted — is transmitted using the conventional C-mode while all the follow-on frames are transmitted in P-mode. This way, the number of contending stations is reduced. Moreover, the contention window size for each wireless station is carefully selected to achieve a low frame collision probability and maximize the channel utilization.

The authors of [8] also applied a hybrid idea — similar to ours — in the Distributed-Queuing Request Update Multiple Access (DQRUMA) protocol in which wireless stations need to send requests to the access point (AP) only for the frames that arrive at an empty buffer. For those frames that arrive at a non-empty buffer, transmission requests are piggybacked with preceding frame transmissions, thus ensuring collision-freedom. However, this protocol simply uses a "harmonic" slotted ALOHA algorithm for the transmission requests. In addition, DQRUMA assumes that uplink and downlink communications are physically separate (i.e., using different frequency channels), which does not hold for the IEEE 802.11 WLAN.

The rest of this paper is organized as follows. Section II introduces the DCF of IEEE 802.11 MAC, discusses the WLAN architecture of our interest, and states the assumptions to be used. The details of UMAV are presented in Section III. Section IV presents and discusses the simulation results, and the paper concludes with Section V.

II. System Overview

A. DCF of IEEE 802.11 MAC

The DCF [1], as the basic access mechanism of the IEEE 802.11 MAC, achieves automatic medium sharing between compatible stations through the use of Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA). Before a station starts transmission, it senses the wireless medium to determine if it is idle. If the medium appears to be idle, the transmission may proceed, else the station will wait until the end of the in-progress transmission. The CSMA/CA mechanism requires a minimum specified gap/space between contiguous frame transmissions. A station will ensure that the medium has been idle for the specified inter-frame interval before attempting to transmit.

The Distributed Inter-Frame Space (DIFS) is used by stations operating under the DCF to transmit data frames. A station using the DCF has to follow two medium access rules: (1) the station will be allowed to transmit only if its carrier-sense mechanism determines that the medium has been idle for at least DIFS time; and (2) in order to reduce the collision probability among multiple stations accessing the medium, the station will select a random backoff interval after deferral, or prior to attempting to transmit another frame after a successful transmission.

One important characteristic of the IEEE 802.11 MAC is that an acknowledgment (Ack) frame will be sent by the receiver upon successful reception of a data frame. Only after receiving an Ack frame correctly, the transmitter assumes successful delivery of the corresponding data frame. The Short Inter-Frame Space (SIFS), which is smaller than DIFS, is the time interval between reception of a data frame and transmission of its Ack frame. Using this small gap between transmissions within the frame exchange sequence prevents other stations — which are required to wait for the medium to be idle for a longer gap (e.g., at least DIFS time) — from attempting to use the medium, thus giving priority to completion of the in-progress frame exchange sequence. Moreover, the DCF defines an optional mechanism, which requires that the transmitter and receiver exchange short Request-To-Send (RTS) and Clear-To-Send (CTS) control frames prior to the actual data frame transmission.

The DCF adopts a *slotted binary exponential backoff* mechanism to select the random backoff interval (in unit of *tSlotTime*). This random number is drawn from a uniform distribution over the interval [0, $CW-1$], where *CW* is the contention window size and its initial value is *aCWmin*. In the case of an unsuccessful transmission, *CW* is doubled. Once *CW* reaches *aCWmax*, it will remain at this value until it is reset to *aCWmin*. In the case of a successful transmission, the *CW* value is reset to *aCWmin* before the random backoff interval is selected. Each station decrements its backoff counter every *tSlotTime* interval after the wireless medium is sensed to be idle for DIFS time. If the counter has not reached zero and the medium becomes busy again, the station freezes its counter. When the counter finally reaches zero, the station starts its transmission.

Fig. 1 illustrates such an operation of decrementing the backoff counter. After the successful transmission and acknowledgment of frame A1, station A waits for DIFS time and selects a backoff interval equal to 6, before attempting to transmit the next frame A2. Assume that station B selects a smaller backoff interval equal to 3 after it has sensed the medium to be idle for DIFS time. Since the backoff counter of station B reaches zero before that of station A, frame B1 is transmitted after exchanging the RTS and CTS frames. As a result of the medium sensed busy, the backoff counter of station A is frozen at 3, and decrements again after the medium is sensed idle for DIFS time.

B. Network Architecture and Assumptions Used

Basic Service Set (BSS) is the basic building block of an IEEE 802.11 WLAN. It consists of a set of stations controlled by a single coordination function. There are two types of BSS [9]: independent BSS and infrastructure BSS. When all the stations in a BSS are wireless stations and there is no connection to the wired network, the BSS is called an independent BSS (IBSS) and is often referred to as an ad hoc net-

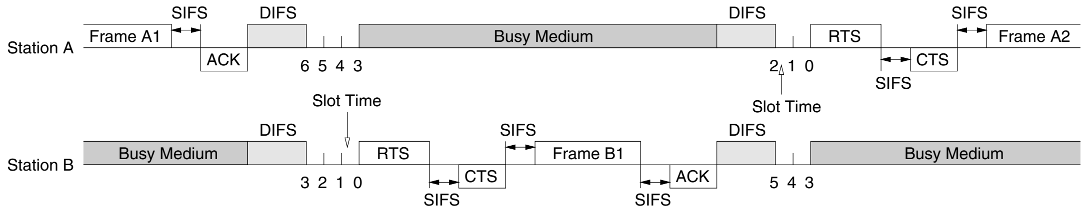


Fig. 1. An example of data frame transmissions and backoff decrements under the IEEE 802.11 DCF

work. An IBSS is typically a short-lived network with a small number of stations, which is created for a particular purpose — for example, to share data among a small group of people during a meeting. On the other hand, when a BSS includes an access point (AP), the BSS is no longer independent and is called an infrastructure BSS. The AP provides both the connection to the wired network, if any, and the local relay function within the BSS. The infrastructure BSS is the most popularly-deployed WLAN architecture, which can be often found in in-door office, home, or public access environments. For this reason, we focus on the throughput enhancement in an infrastructure BSS. Moreover, since all the downlink (AP-to-station) transmissions are centrally-controlled by the AP, we are more concerned about how to reduce the frame collision probability when the wireless stations contend for uplink (station-to-AP) transmissions.

In this paper, the throughput performance of an 802.11 DCF system is analyzed under the assumptions of ideal channel conditions (i.e., no transmission errors and no hidden terminals) and homogeneous contending traffic flows with identical service requirements. See a companion paper [10] for details on how to achieve the weighted fairness among traffic flows with heterogeneous characteristics while maximizing the channel utilization. Besides, an RTS/CTS frame exchange is required prior to each data frame transmission. Although the analysis and conclusions presented in this paper do not depend on the technology adopted at the physical layer (PHY), the PHY does determine some network parameters, such as SIFS and DIFS. In the simulation, we assume that each wireless station operates at the IEEE 802.11a PHY mode-8, and the related network parameters are summarized in Table I. For more details of the IEEE 802.11a PHY, refer to [11] and [12].

TABLE I
IEEE 802.11A PHY MODE-8 PARAMETERS AND ADDITIONAL NETWORK PARAMETERS USED IN THE SIMULATION

Parameters	Value	Comments
rTransmit	54Mbps	data transmission rate
BpS	27	bytes per OFDM symbol
tSlotTime	9μs	an idle slot time
tSIFSTime	16μs	SIFS time
tDIFSTime	34μs	DIFS = SIFS + 2 $\times$ Slot
tPropDelay	1μs	propagation delay
aRTSLength	20 octets	RTS frame length
aCTSLength	14 octets	CTS frame length
aPayload	2304 octets	data payload length
aAckLength	14 octets	Ack frame length
aMACOverhead	28 octets	MAC sublayer overhead
tPHYOverhead	20μs	PHY layer overhead
tSymbol	4μs	OFDM symbol interval

III. UMAV: UNIFIED MULTIPLE ACCESS WITH VARIABLE CONTENTION WINDOW

In this section, we present the details of UMAV, our simple enhancement to the IEEE 802.11 DCF. The key idea is that, in order to reduce the number of contending stations, a new P(polling)-mode is introduced for frame transmissions in addition to the C(contention)-mode used in the DCF. Besides, the contention window size for each wireless station is carefully selected according to the on-line estimation of the number of contending stations, so as to achieve a low frame collision probability and maximize the channel utilization.

A. Frame Transmission: C-Mode vs. P-Mode

Burstiness is one of the common features seen in data traffic flows, i.e., data frames are often generated in bursts by the source application and queued at the wireless station until they are transmitted. We classify the data frames into the following two categories depending on the queuing status when a data frame is generated and arrives the queue: *head-of-line frames* if the queue is empty and *follow-on frames* if the queue is non-empty. In an 802.11 DCF system, a wireless station is required to contend for each of its uplink frame transmissions, regardless whether it is a head-of-line frame or a follow-on frame. As a result, when the system is heavily-loaded, the network suffers a high degree of contention for the wireless medium and presents poor throughput performance consequently. One natural idea to alleviate this problem is to reduce the number of contending stations. In order to do so, UMAV allows a wireless station to piggyback its local queuing status in the preceding frame transmissions, and therefore, the follow-on frames may be transmitted using a poll-response-like scheme, which we called the P(polling)-mode in comparison to the conventional C(contention)-mode used in the DCF.

```
while (the wireless network is alive) {
   if (ℒ == ∅)    bkoff_value := ∞;
   if (ℒ ≠ ∅) && (bkoff_value == ∞)
      bkoff_value := (cw_opt − 1) × rand() / RAND_MAX;
   ⟨ monitor medium activity, then decrement bkoff_value and/or update cw_opt if necessary ⟩
   switch (one of the three monitored events happens) {
      case (an RTS frame ℛ is received successfully):
            ⟨ transmit frame CTS(ℛ) back to station TA(ℛ) ⟩
            break;
      case (a data frame 𝒟 is received successfully):
            if (more_data(𝒟) == 1)
               ℒ := ℒ ∪ TA(𝒟);
            if (more_data(𝒟) == 0) && (TA(𝒟) ∈ ℒ)
               ℒ := ℒ − TA(𝒟);
            ⟨ transmit frame Ack(𝒟) back to station TA(𝒟) ⟩
            break;
      case (bkoff_value == 0):
            bkoff_value := (cw_opt − 1) × rand() / RAND_MAX;
            ⟨ transmit a POLL frame to next_sta(ℒ) according to the transmission policy ⟩
            break;
   }
}
```

Fig. 2. The algorithm executed by the AP to update its polling list

UMAV defines new POLL frames to support P-mode frame transmissions. The POLL frames have the same frame format as the RTS frames except with value "0011" in the *subtype* subfield[1] of the *frame control* field [9]. The AP maintains a polling list $\mathcal{L}$, which consists of the wireless stations that have follow-on frames waiting to be transmitted in P-mode. The AP only sends the POLL frames to those stations on its polling list, and since the AP is also equipped with a DCF device, it has to contend for the wireless medium to transmit the POLL frames. As a result, collisions may occur to the RTS frames as well as to the POLL frames. Fig. 2 shows the pseudo-coded algorithm executed by the AP to update its polling list according to the information carried in the incoming data frames. As shown in the pseudo-code, the AP adds a station to $\mathcal{L}$ when it receives from the station a data frame with the *more data* bit set to 1 in the *frame control* field. On the other hand, the AP removes a station from $\mathcal{L}$ when it receives from the station a data frame with the *more data* bit set to 0.

In UMAV, each wireless station maintains a local *xmit_mode* flag that is initially set to C-mode. Before a frame transmission attempt, a wireless station checks whether there are more data frames waiting in the queue, and sets the *more data* bit in the current data frame accordingly. Then, the data frame is transmitted to the AP using the mode specified by the *xmit_mode* flag. Figs. 3 and 4 illustrate the different timings of successful uplink frame transmissions in C-mode and P-mode, respectively. After the frame transmission attempt, the *xmit_mode* flag is updated to P-mode if the queue is non-empty, or to C-mode otherwise.

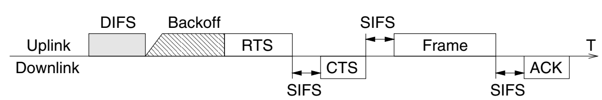


Fig. 3. C-mode uplink frame transmission

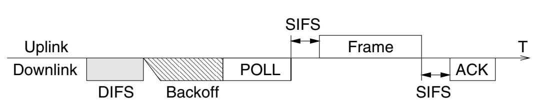


Fig. 4. P-mode uplink frame transmission

Actually, for those frame transmissions in P-mode, the AP could schedule them according to any desired transmission policy. The basic idea is that the wireless stations communicate their QoS requirements to the AP, and based on this information, the AP determines the best scheduling strategy to optimize the given performance metrics, such as throughput, delay, or frame loss rate [13]. In this paper, since we have made the assumption of homogeneous characteristics and QoS requirements for the contending traffic flows, the AP simply employs the round-robin transmission policy.

[1] In the IEEE 802.11 standard, the RTS frames have value "1011" in the *subtype* subfield of the *frame control* field, while the value of "0011" is reserved for future use.

0-7695-1874-5/03 $17.00 © 2002 IEEE

In UMAV, the mode selection for frame transmissions depends heavily on the traffic load. When the system is lightly-loaded, most new frames see empty queues upon their generation and are transmitted in C-mode. When the system is heavily-loaded, most new frames appear as follow-on frames and are transmitted in P-mode. In particular, when the network is running under the saturation condition (i.e., the queues of active wireless stations are never empty), all the frames will be transmitted in P-mode after the system gets stabilized. In this extreme case, UMAV works exactly like a polling scheme. On the other hand, the worst-case traffic scenario happens if all the frames appear as head-of-line frames. In this case, each data frame has to be transmitted in C-mode, and UMAV works exactly like a contention-based scheme. Since UMAV selects the optimal contention window size for C-mode frame transmissions — the details will be presented in the next section, we claim that the throughput performance of UMAV is no worse than that of any other purely contention-based scheme, including the DCF.

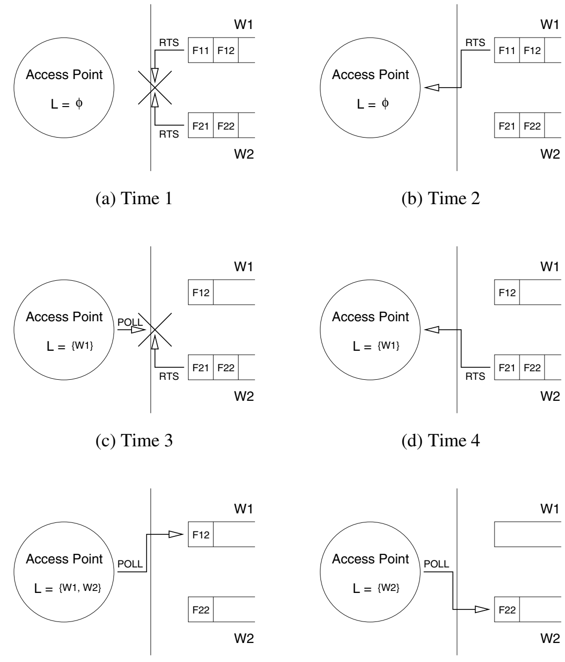
Fig. 5 presents a simple example to illustrate how UMAV works. Assume there are two wireless stations, W1 and W2, in the network and each station has two data frames waiting for transmission. Initially, W1 and W2 attempt to transmit their head-of-line frames, F11 and F21 respectively, in C-mode. After the RTS frame collision, W2 backs off and W1 continues its transmission attempt. At Time 2, W1 successfully transmits its RTS frame to the AP, and as a result, the AP adds W1 to its polling list. The AP has to contend with W2 to send its POLL frames to W1. We can see that the POLL frames may also collide with the RTS frames. Eventually, at Time 4, an RTS frame from W2 gets through, and W2 is added to the AP's polling list. Both follow-on frames, F12 and F22, will then be transmitted in P-mode to avoid further collisions.

B. Optimal Contention Window Selection

Although introducing P-mode frame transmissions reduces the number of contending stations, it does not eliminate the collisions totally. As shown in Fig. 5, collisions may occur to both RTS frames and POLL frames. Therefore, as the second enhancement to the DCF, UMAV replaces the original binary exponential backoff scheme with an enhanced uniform backoff scheme to resolve collisions. It selects the optimal contention window size for each station based on the following analysis, which is similar to what was presented in [3], so as to maximize the channel utilization.

Let n be the number of contending stations in the network. Assume that one of the contending stations u uses a contention window CW_u to access the wireless medium,[2] and initially, its backoff value $b_u(t)$ is uniformly selected from the range

[2]Since UMAV applies a uniform backoff scheme, station u will use this contention window CW_u to select the backoff intervals for all of its frame transmission attempts.



(a) Time 1 (b) Time 2

(c) Time 3 (d) Time 4

(e) Time 5 (f) Time 6

Fig. 5. An example of frame transmissions under the UMAV

[0, CW_u-1]. As illustrated in Fig. 1, $b_u(t)$ is decremented at the end of each time slot, which could be either an idle period of length *tSlotTime*, or a busy period due to a collision, or a busy period due to a successful frame transmission. Note that t is a discrete time point corresponding to the end of a time slot. Then, as indicated in [3], the stochastic process $b_u(t)$ can be modeled by the following discrete-time Markov chain:

$$P\{b_u(t+1)=k\} = \begin{cases} P\{b_u(t)=k+1\} + \frac{P\{b_u(t)=0\}}{CW_u} & \text{for } k=0,\cdots,CW_u-2, \\ \frac{P\{b_u(t)=0\}}{CW_u} & \text{for } k=CW_u-1, \end{cases} \tag{1}$$

where the term $P\{b_u(t)=k+1\}$ corresponds to decrementing the backoff value at the end of each time slot. The term $\frac{P\{b_u(t)=0\}}{CW_u}$ accounts for the fact that, after a frame transmission attempt, the new backoff value is uniformly selected from the range [0, CW_u-1], regardless whether the frame transmission was successful or not. The steady state probabilities of this Markov chain are:

$$\lim_{t\to\infty} P\{b_u(t)=k\} = \frac{2\cdot(CW_u-k)}{CW_u\cdot(CW_u+1)}. \tag{2}$$

Recall that, when the backoff counter finally reaches zero, the station starts its transmission. Therefore, the probability that station u transmits in a randomly-chosen time slot is

$$p_u = \lim_{t\to\infty} P\{b_u(t)=0\} = \frac{2}{CW_u+1}. \tag{3}$$

The probability that at least one station attempts to transmit in a slot, or equivalently, the probability that a slot is not idle, is given by

$$P_{tr} = 1 - (1 - p_u)^n. \tag{4}$$

Since the probability of a successful transmission is given by

$$SU = n \cdot p_u \cdot (1 - p_u)^{n-1}, \tag{5}$$

and $(1 - P_{tr})$ is the probability of an idle slot, the aggregate throughput can be calculated as

$$\mathcal{T} = \frac{SU \cdot \mathit{aPayload}}{SU \cdot \ell_{succ} + (P_{tr} - SU) \cdot \ell_{coll} + (1 - P_{tr}) \cdot \mathit{tSlotTime}}, \tag{6}$$

where *aPayload* is the data payload length, ℓ_{succ} is the length of a successful frame transmission, ℓ_{coll} is the collision length, and *tSlotTime* is the length of an idle time slot. Eq. (6) can be rewritten as

$$\mathcal{T} = \frac{\mathit{aPayload}}{\ell_{succ} - \ell_{coll} + \frac{P_{tr} \cdot \ell_{coll} + (1 - P_{tr}) \cdot \mathit{tSlotTime}}{SU}}. \tag{7}$$

Since *aPayload*, ℓ_{succ}, ℓ_{coll}, and *tSlotTime* are constant for all the stations, maximization of the aggregate throughput is equivalent to maximization of

$$\begin{aligned} \mathcal{T}' &= \frac{SU}{P_{tr} \cdot \ell_{coll} + (1 - P_{tr}) \cdot \mathit{tSlotTime}} \\ &= \frac{n \cdot p_u \cdot (1 - p_u)^{n-1}}{[1 - (1 - p_u)^n] \cdot \ell_{coll} + (1 - p_u)^n \cdot \mathit{tSlotTime}} \end{aligned} \tag{8}$$

It is clear from Eqs. (8) and (3) that $\mathcal{T}'$ depends on CW_u via p_u. By solving the equation

$$\frac{d\,\mathcal{T}'}{d\,p_u} = 0, \tag{9}$$

we get an approximate solution to the optimal value of p_u to maximize the aggregate throughput:

$$p_u^* \approx \frac{1}{n} \cdot \sqrt{\frac{2 \cdot \mathit{tSlotTime}}{\ell_{coll}}}, \tag{10}$$

and the corresponding optimal contention window size is

$$CW_u^* \approx n \cdot \sqrt{\frac{2 \cdot \ell_{coll}}{\mathit{tSlotTime}}} - 1. \tag{11}$$

Note that the collision length in the case with RTS/CTS support is given in [12] by

$$\begin{aligned} \ell_{coll(rts/cts)} = \ & \mathit{tPHYOverhead} \\ & + \left\lceil \frac{\mathit{aRTSLength} + 2.75}{\mathit{BpS}} \right\rceil \cdot \mathit{tSymbol} \\ & + \mathit{tPropDelay} + \mathit{tDIFSTime}. \end{aligned} \tag{12}$$

C. On-line Load Estimation

UMAV requires that each station keeps sensing the channel and monitoring the activities on the wireless medium when it is not transmitting. Therefore, each station knows whether at each time slot the wireless medium is busy or idle, and whether a busy period corresponds to a collision or not. Let *avg_idle* and *avg_wait* denote the average number of consecutive idle slots on the wireless medium and the average number of time slots between two consecutive successful frame transmissions, respectively, and they can be calculated as

$$\mathit{avg\_idle} = \frac{1}{P_{tr}} - 1, \tag{13}$$

and

$$\mathit{avg\_wait} = \frac{1}{SU} - 1. \tag{14}$$

Here, P_{tr} is the probability that at least one station attempts to transmit in a slot and it is given by Eq. (4). SU is the probability of a successful frame transmission and it is given by Eq. (5). Notice that we have the following relation:

$$\frac{1 - P_{tr}}{SU} = \frac{1 - p_u}{n \cdot p_u}, \tag{15}$$

and hence,

$$n = \frac{(CW_u - 1) \cdot (\mathit{avg\_idle} + 1)}{2 \cdot \mathit{avg\_idle} \cdot (\mathit{avg\_wait} + 1)}. \tag{16}$$

Based on the measurements of *avg_idle* and *avg_wait* by monitoring the medium activities, each station can estimate the value of n using Eq. (16) and then determine the optimal contention window size using Eq. (11).

Fig. 6 shows the pseudo-coded algorithm executed by each station to adjust its contention window size. Each station maintains two random variables, "IDLE" and "WAIT". The optimal contention window size, *cw_opt*, is initialized to *cw_start*, a design parameter. Let an idle-busy-cycle be the time interval between the ends of two adjacent busy periods on the wireless medium. The observation window size w_{obs}, another design parameter, represents the number of idle-busy-cycles within which the measurements of *avg_idle* and *avg_wait* are taken, and the count w_{count} for monitored idle-busy-cycles is reset to 0. As shown in the pseudo-code, "IDLE" is updated after each idle-busy-cycle, while "WAIT" is updated only if the busy period corresponds to a successful frame transmission. At the end of each observation window, the number of contending stations, n, is estimated, and the contention window size is adjusted according to this estimate. Notice that t_{curr} and t_{prev} are the discrete time points measured in time slots, and α and β are both smoothing factors.

```
cw_opt := cw_start;  w_count := 0;
while (the wireless network is alive) {
  if (a new idle-busy-cycle has been monitored) {
    w_count := w_count + 1;
    IDLE := α · IDLE + (1 − α) · new_idle;
    if (busy period is due to a successful transmission) {
      WAIT := α · WAIT + (1 − α) · [t_curr − t_prev − 1];
      t_prev := t_curr;
    }
    if (w_count = 0 mod w_obs) {
      n_est = β · n_est + (1 − β) · ((cw_opt−1)·(IDLE+1)) / (2·IDLE·(WAIT+1));
      ⟨ calculate cw_opt using Eq. (11) based on n_est ⟩
      ⟨ reset IDLE, WAIT, and t_prev ⟩
    }
  }
}
```

Fig. 6. The algorithm executed by each station to adjust the contention window size

IV. Performance Evaluation and Discussion

We use simulation to evaluate the effectiveness of the proposed UMAV scheme. In the simulation, we consider a wireless network with an AP and 10 wireless stations contending for the shared medium. The burst arrival process at each wireless station is assumed to be Bernoulli [8], with a deterministic number of data frames per burst. In this section, we present the simulation results for three different burst models: one frame per burst, two frames per burst, and five frames per burst. Obviously, "one frame per burst" is the worst among the three for UMAV, since the data frames generated by this model are most likely to become head-of-line frames. On the other hand, "five frames per burst" is the one that favors UMAV most.

The three testing schemes under consideration are: (i) UMAV; (ii) UMAV w/o P-mode, in which all the data frames are transmitted in C-mode; and (iii) the IEEE 802.11 DCF with *aCWmin* = 16 and *aCWmax* = 1024, as specified in [11]. The network parameters used in the simulation are listed in Table I. In the simulation runs of UMAV and UMAV w/o P-mode, the observation window w_{obs} is set to 50, while the smoothing factors α and β are chosen to be 0.9 and 0.7, respectively. The duration of each simulation run is 6 minutes.

In the first part of the simulation, the testing schemes are compared in terms of the *aggregate system throughput* and the *average station access delay*. The station access delay is defined as the time elapsed from a data frame reaching the head of the queue until the frame is successfully transmitted. For simplicity, the simulation results for the station access delay are measured in time slots. For example, in Fig. 1, the access delay of station A when transmitting frame A2 is 6 time slots. Note that such results illustrate the same information as those in absolute time units, or in seconds.

Fig. 7 shows the simulation results for the "one frame per burst" model. Along the X-axis are the generating probabilities (*p_ber*) of the Bernoulli burst arrival process. We have two observations. First, UMAV w/o P-mode only shows a marginal, if any, throughput improvement over the DCF. This observation is surprising at the first sight, but rather reasonable. In UMAV w/o P-mode, the contention window size is carefully selected to achieve a low frame collision probability and maximize the aggregate system throughput. As shown in Fig. 7(a), when *p_ber* > 0.04, throughput is almost constant around 33Mbps, which is the maximum possible with any purely contention-based access protocol for our simulated network. On the contrary, in the DCF, the slotted binary exponential backoff mechanism is heuristic and may result in more collisions. However, due to the introduction of the RTS/CTS mechanism, the collision length is reduced drastically since collisions only occur to the RTS frames that are much shorter than the data frames. As a result, the more collisions resulted in the DCF do not have significant impact on the throughput performance. For the same reason, UMAV w/o P-mode also shows comparable delay performance to the DCF under heavy loads. On the other hand, it does show better delay performance under light loads, because a wireless station running UMAV w/o P-mode can select a very small contention window when the system is lightly-loaded, and transmit its data frames shortly after they are generated, instead of waiting for a longer backoff period as required in the DCF.

Second, UMAV shows similar performance to UMAV w/o P-mode under light loads, because most new frames see empty queues upon their generation and are transmitted in C-mode. However, UMAV achieves significantly higher throughput and shorter access delay under medium to heavy loads. The rationale behind these is that, when the traffic load increases, more and more frames appear as follow-ons and are transmitted in

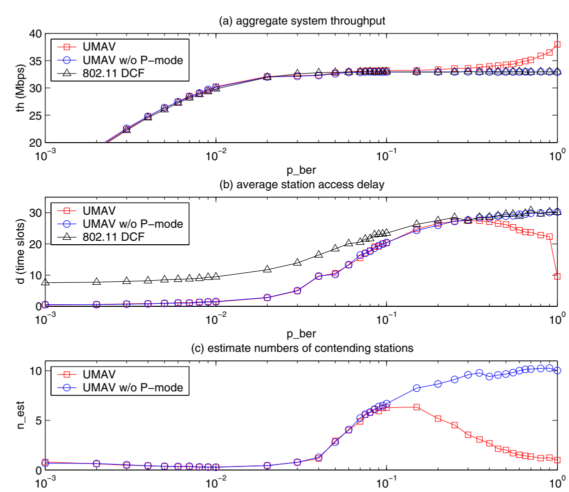


Fig. 7. Comparison results for "one frame per burst" model

P-mode, and the number of contending stations starts decreasing. Fig. 7(c) compares the estimated numbers of contending stations under different traffic load conditions for UMAV and UMAV w/o P-mode. Clearly, the hump curve for UMAV supports the facts explained above. In contrast, without P-mode, the number of contending stations increases monotonically. When *p_ber* reaches 1, i.e., the system is saturated, UMAV is reduced to polling and achieves the highest throughput (∼38Mbps) and a short access delay (∼9 time slots). Note that 9 time slots is the shortest possible station access delay under the saturation condition of our network configuration, where 10 wireless stations are transmitting in a round-robin fashion.

Figs. 8 and 9 plot the simulation results for the "two frames per burst" model and the "five frames per burst" model, respectively. As expected, the performance improvements of UMAV over UMAV w/o P-mode and the DCF are more pronounced due to the increasing percentage of the P-mode frame transmissions. Fig. 10 compares the testing schemes using the (throughput, delay) plots. Notice that the plots of both UMAV w/o P-mode and the DCF are quite similar for different burst models, and the throughput bound of ∼33Mbps can be clearly observed in the figure. On the other hand, thanks to its mixture of contention with polling, UMAV achieves higher throughput with shorter access delay. As shown in Fig. 10(c), under the "five frames per burst" model, UMAV can even achieve throughput of 36Mbps with access delay less than 3 time slots. The saturation points (∼38Mbps, ∼9 time slots) for UMAV are singled out by the left arrows in the figure.

In the second part of the simulation, we study the behavior of the load estimation function of UMAV in presence of traffic fluctuations. The burst model used in the simulation is "one frame per burst". The variation pattern of *p_ber* is represented by the stair case curve in Fig. 11(a), and the estimated numbers of contending stations in UMAV and UMAV w/o P-mode are plotted in Figs. 11(b) and (c), respectively. Clearly, these estimates follow the load variations and react to the changes quickly. The figure also shows that UMAV results in less contending stations under various traffic load conditions, which is consistent with the previous observations in Fig. 7.

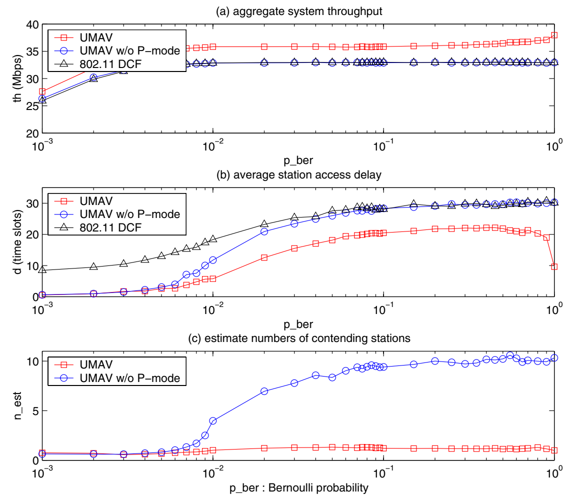


Fig. 9. Comparison results for "five frames per burst" model

We also evaluate the fairness of the testing schemes using a measure called the *fairness index* as follows. Let τ_s denote the share of bandwidth occupied by the wireless station s. The fairness index is then defined as:

$$\text{fairness index} = \frac{\mu(\tau_s)}{\mu(\tau_s) + \sigma(\tau_s)}, \tag{17}$$

where $\mu(\tau_s)$ and $\sigma(\tau_s)$ are, respectively, the mean and the standard deviation of τ_s over all the active contending stations.

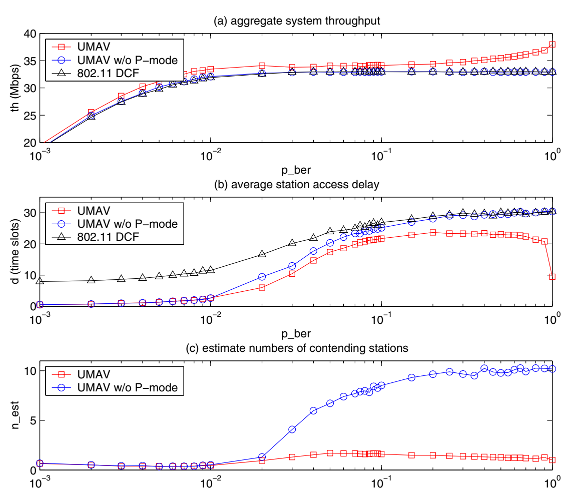


Fig. 8. Comparison results for "two frames per burst" model

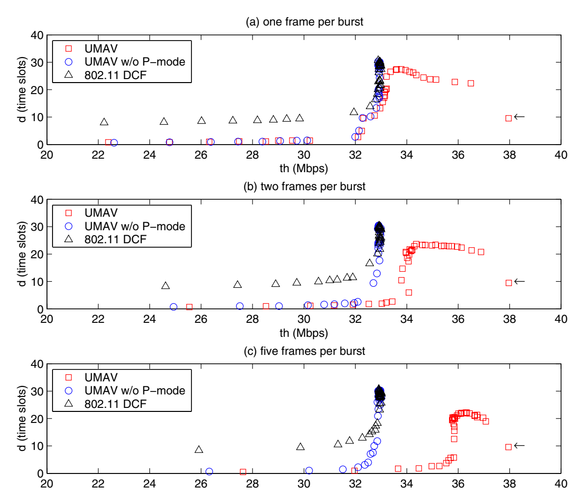


Fig. 10. Aggregate system throughput vs. average station access delay

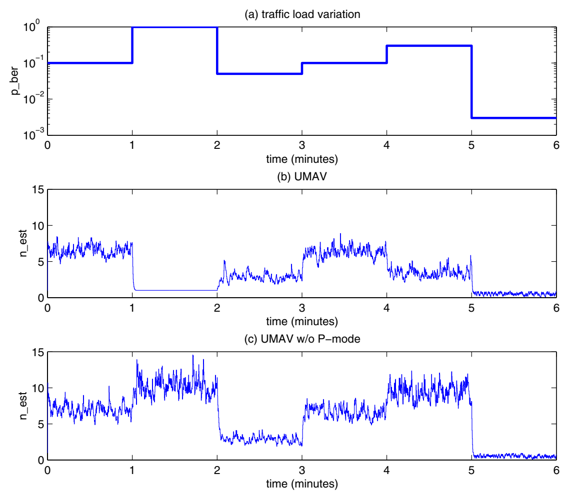


Fig. 11. An example of the load estimation function of UMAV

When the perfect fairness is achieved, τ_s is the same for all the wireless stations, and the fairness index is equal to 1. In general, the fairness index is a real value between 0 and 1, and the closer to 1 the fairness index, the fairer. The comparison results are plotted in Fig. 12. We observe that UMAV presents the best fairness performance among the three, and in particular, it outperforms the DCF significantly under medium to heavy load conditions. This is because in UMAV, a uniform optimal contention window is used by all the wireless stations for their C-mode frame transmissions, and the round-robin transmission policy for the P-mode frame transmissions also results in fair medium sharing among the wireless stations on the polling list.

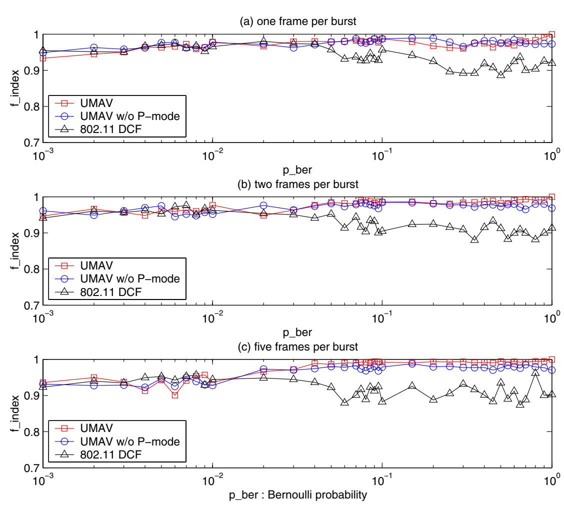


Fig. 12. Comparison of the testing schemes using the fairness index

V. Conclusion

In this paper, we propose a simple enhancement to the IEEE 802.11 DCF, called the *Unified Multiple Access with Variable contention window* (UMAV), to improve the system throughput. The basic ideas of UMAV are that (1) the number of contending stations is reduced by introducing the P-mode frame transmissions; and (2) the contention window size for each wireless station is carefully selected according to the on-line load estimation, so as to achieve a low frame collision probability and maximize the channel utilization. We evaluate the throughput and delay performances of UMAV using simulation. The simulation results show that UMAV outperforms the IEEE 802.11 DCF significantly in terms of both throughput and delay under various load conditions, and it is able to react properly and quickly to the traffic load fluctuations. In addition, UMAV provides excellent fairness, a byproduct, which is not one of the original design goals.

References

[1] IEEE 802.11, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Standard, IEEE, Aug. 1999.

[2] Agere Systems, *Brochure for ORiNOCO PC Card*, Oct. 2001.

[3] Giuseppe Bianchi, Luigi Fratta, and Matteo Oliveri, "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs," in *Proc. IEEE PIMRC*, Taipei, Taiwan, Oct. 1996, pp. 407–411.

[4] Frederico Cali, Marco Conti, and Enrico Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 785–799, Dec. 2000.

[5] Sunghyun Choi, "PCF vs. DCF: Limitations and Trends," IEEE 802.11-01/154, Jan. 2001.

[6] IEEE 802.11e/D2.0a, *Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, Draft Supplement to IEEE 802.11 Standard-1999 Edition, Nov. 2001.

[7] Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," in *Proc. European Wireless (EW'2002)*, Florence, Italy, Feb. 2002.

[8] Mark J. Karol, Zhao Liu, and Kai Y. Eng, "Distributed-Queueing Request Update Multiple Access (DQRUMA) for Wireless Packet (ATM) Networks," in *Proc. IEEE ICC'95*, Seattle, WA, 1995, pp. 1224–1231.

[9] Bob O'Hara and Al Petrick, *The IEEE 802.11 Handbook: A Designer's Companion*, Standards Information Network, IEEE Press, 1999.

[10] Daji Qiao and Kang G. Shin, "Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF," in *Proc. IWQoS'02*, Miami Beach, FL, May 2002, pp. 227–236.

[11] IEEE 802.11a, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band*, Supplement to IEEE 802.11 Standard, Sep. 1999.

[12] Daji Qiao and Sunghyun Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," in *Proc. IEEE ICC'01*, Helsinki, Finland, Jun. 2001.

[13] Daji Qiao and Kang G. Shin, "A Two-Step Adaptive Error Recovery Scheme for Video Transmission over Wireless Networks," in *Proc. IEEE INFOCOM'00*, Tel-Aviv, Israel, Mar. 2000, vol. 3, pp. 1698–1704.