A Credit-based Distributed Protocol for Long-term Fairness in IEEE 802.11 Single-Hop Networks

Yan Wu, and Sonia Fahmy, *Member, IEEE*

***Abstract*— Fair bandwidth allocation is critical in wireless communication networks, since the wireless channel is often shared by a number of stations in the same neighborhood. With fair scheduling, bandwidth can be shared by competing flows in proportion to their assigned weights. In this paper, we propose a credit-based distributed protocol for fair allocation of bandwidth in IEEE 802.11 wireless LANs. Our protocol is derived from the Distributed Coordination Function in the IEEE 802.11 medium access control (MAC) protocol. Analytical and simulation results demonstrate that the protocol achieves the desired bandwidth allocations. An important feature of our protocol is its *backward compatibility*, which allows legacy IEEE 802.11 stations to coexist with stations adopting the new MAC protocol.**

***Index Terms*— 802.11, Fairness, Medium access control (MAC), Wireless local area networks (WLANs)**

I. Introduction

IEEE 802.11 Wireless Local Area Networks (WLANs) have become increasingly prevalent in recent years. In IEEE 802.11 WLANs, a channel is shared by all stations in the neighborhood of an access point (AP). Dividing the limited channel bandwidth fairly among stations is an important and challenging problem. For example, consider a WLAN user sharing files with other peers outside the WLAN using systems such as the BitTorrent peer-to-peer system. The more data the user sends to its peers, the more data it can receive from peers. Therefore, the user may want to send data as quickly as possible, in order to receive more data. When WLAN users are sharing files with peers outside the WLAN, dividing the limited wireless channel bandwidth among the users fairly becomes crucial, especially if users will be charged (either directly or indirectly) for the service.

Ideally, bandwidth should be shared by all competing users proportional to a "weight" assigned to each user. Users who pay a higher price must be assigned larger weights, so that they can obtain higher bandwidth. The key challenge in WLAN channels is that there is no centralized scheduling server, as in the case of a router output port in a wireline environment. Instead, the scheduling operation is distributed among wireless stations with data to send. It is therefore necessary to design a fully distributed scheduling algorithm to allocate bandwidth fairly. In addition, considering the ubiquity of IEEE 802.11 WLANs and users, this scheduling algorithm must *inter-operate* with legacy stations in order to be gradually deployable.

In this paper, we consider a typical single-hop wireless LAN environment, in which all the stations are in the same neighborhood, and share the same channel. We propose a fully distributed scheduling algorithm, which we refer to as Distributed Deficit Credit (DDC), to allocate bandwidth in proportion to the flow weights. The algorithm is an extension of the Distributed Coordination Function (DCF) of the IEEE 802.11 medium access control (MAC) protocol. An important feature of our algorithm is its backward compatibility with the current 802.11 MAC protocol.

The remainder of this paper is organized as follows. Section II describes the basic features of the Distributed Coordination Function in IEEE 802.11. Section III reviews prior work on fair queuing, especially in IEEE 802.11 networks. Section IV describes our proposed algorithm. Simulation results are given in Section V. Section VI gives a brief summary of our work, and our plans for future work.

II. IEEE 802.11 Distributed Coordination Function

IEEE 802.11 medium access control (MAC) includes a mandatory contention-based channel access function called Distributed Coordination Function (DCF), and an optional centrally controlled channel access function called Point Coordination Function (PCF). The DCF is designed for asynchronous data transmission and is fully distributed. In contrast, the PCF is intended for transmission of both real time traffic and asynchronous data traffic. PCF is a centralized, polling-based access mechanism controlled by the AP.

In this work, we focus on *distributed* mechanisms for *proportional* bandwidth allocation. Hence, we summarize the DCF in this section. For a more detailed discussion, please refer to the IEEE 802.11 standard [1].

The DCF is based on the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol. A station with a new packet to transmit first senses the channel. If the channel is sensed to be idle for a time interval equal to the DCF inter-frame space (DIFS), the station transmits. Otherwise, the station continues to sense the channel until it is sensed idle for a period of DIFS.

DCF adopts an exponential backoff scheme. A backoff counter is chosen uniformly in the range [0, CW$-$1], where CW is the contention window. A backoff time is computed as $Tbackoff = backoff\_counter \times Tslot$, where $Tslot$ is the slot

Manuscript received on January 10, 2005. This research has been sponsored in part by NSF grant ANI-0238294 (CAREER) and the Schlumberger Foundation technical merit award.

Yan Wu is with the Department of Computer Science, Purdue University, 250 N. University St., West Lafayette, IN 47907–2066, USA (phone: +1-765-494-2957; e-mail: wu26@cs.purdue.edu).

Sonia Fahmy is with the Department of Computer Science, Purdue University, 250 N. University St., West Lafayette, IN 47907–2066, USA (phone: +1-765-494-6183; fax: +1-765-494-0739; e-mail: fahmy@cs.purdue.edu).

time. At the first packet transmission attempt, CW is set to a value CWmin, which denotes the minimum contention window size. After each unsuccessful transmission, CW is doubled until a predefined maximum size (CWmax) is reached.

The backoff counter is decremented once every *Tslot* time, as long as the channel is sensed idle. The counter is frozen when a transmission is detected, and reactivated when the channel is sensed idle again for more than a DIFS period of time. The station transmits when the backoff counter reaches zero. If two or more stations transmit at the same time, collision occurs.

Since CSMA/CA does not rely on a station to detect a collision by hearing its own transmission, an ACK is transmitted by the destination station to signal successful packet reception. If the ACK is not received, the station assumes that the transmitted frame is not received and reschedules the packet transmission according to the backoff process.

The 2-way handshake mechanism described above is called the basic access mechanism. The DCF MAC protocol defines an additional RTS/CTS mechanism: When the backoff counter reaches zero, the station does not transmit the data frame right away, but sends a request-to-send (RTS) frame. When the destination station receives the RTS frame, it responds with a clear-to-send (CTS) frame. The source station transmits the data frame after receiving the CTS frame. The RTS/CTS mechanism is effective in terms of system performance when the packet length is large, since it reduces the collision time.

III. Related Work

Proportional bandwidth allocation in wireline environments has been extensively studied in the last decade. Generalized Processor Sharing (GPS) [2] assumes multiple flows are served simultaneously, and the traffic is infinitely divisible. Under this assumption, it is shown that GPS can achieve proportional allocation of bandwidth within an infinitely small time interval. Clearly, GPS is an idealized fairness model that cannot be practically implemented. A number of packetized approximations of GPS have been proposed in the past, including Weighted Fair Queuing [3], Self-Clocked Fair Queuing [4], Virtual Clock [5], Start-Time Fair Queuing [6] and Deficit Round Robin [7]. An exact service sequence is provided in [3]–[6] by serving packets in the order of a computed "virtual time tag" associated with each packet. In contrast, Deficit Round Robin (DRR) [7] uses a credit-based approach to provide proportional bandwidth allocation at time scales larger than a round.

A. Scheduling in Cellular Networks

In the context of wireless cellular networks, several studies have been conducted on fair queuing. Lu et al. [8] proposed a mechanism referred to as wireless packet scheduling (WPS), which extends the scheduling policies of wireline networks to wireless networks. Opportunistic scheduling was proposed in [9], [10]. In these studies, the wireless channel is used opportunistically to achieve an optimal use of resources, yet provide fairness among users.

B. Fairness in IEEE 802.11 WLANs

A number of studies have investigated service differentiation and fairness mechanisms in IEEE 802.11 WLANs. Deng and Chang [11] proposed a scheme that differentiates among priority classes by adjusting the backoff window: higher priority classes use a smaller backoff window than lower priority classes. Aad and Castelluccia [12] proposed a service differentiation mechanism that uses different inter-frame spaces. Veres et al. [13] used the initial backoff window size and the maximum window size to differentiate among users. Xiao [14] proposed an analytical model to evaluate backoff-based priority schemes.

Recently, fairness between the uplink and the downlink in IEEE 802.11 WLANs has received attention. Pilosof et al. [15] observed unfairness between the uplink and the downlink TCP flows. Uplink flows receive significantly higher throughput than downlink flows. They find that the buffer size at the AP plays a key role in the observed unfairness, and propose a solution based on TCP receiver window manipulation. Kim and Fang [16] identified the fairness problem between uplink and downlink traffic flows in IEEE 802.11 DCF. Since in DCF, the AP and the stations have equal access to the channel, when the downlink has a higher traffic load than the uplink, the downlink becomes a bottleneck. To solve this problem, they propose a controllable resource-allocation scheme between uplink and downlink flows, which adapts the parameters according to the dynamic traffic load. The scheme also improves the system utilization by reducing the collision probability. Dunn et al. [17] proposed a scheme that exploits IP path Maximum Transmission Unit (MTU) discovery to fairly allocate bandwidth. Bandwidth allocation is achieved by assigning different MTU values to stations. Experiments show that this method works well when IP is the only network layer protocol and all stations use IP MTU discovery.

Scheduling in PCF has also been well studied. Coutras et al. [18] modeled real time traffic as a Markov modulated fluid process, and proposed a scheme to manage the time of polling for each station. Sharon and Altman [19] proposed a scheme referred to as simultaneous transmit response polling (STRP), which reduces the polling overhead caused by stations having no data to transmit. Other priority-based polling schemes have been studied in [20], [21].

C. Proportional Sharing in DCF

Several studies have investigated algorithms to provide proportional sharing of bandwidth in IEEE 802.11 WLANs using *distributed* control. Distributed Fair Scheduling (DFS) [22] is proposed to emulate Self-Clocked Fair Queuing (SCFQ) [4] in IEEE 802.11 DCF. The essential idea in DFS is to select a backoff interval that is proportional to the finish tag of the packet to be transmitted. DFS modifies the computation of the backoff counter to:

$$backoff\_counter = scaling\_factor \times pkt\_size/w \times \rho.$$

In this formula, the *scaling_factor* denotes a fixed constant (same value at all stations), and allows the choice of suitable scales. The *pkt_ size* is the size of the outgoing packet; w is the assigned weight of the station; and ρ is a random variable uniformly chosen in the range [0.9, 1.1]. The purpose of ρ is to randomize the *backoff_counter* and reduce the probability of collision. The intuition behind DFS is that packets from different stations are served approximately in increasing order of their finish tags, which emulates SCFQ. When collisions occur, however, the exact service sequence may not be maintained.

Banchs and Perez [23] proposed Distributed Weighted Fair Queuing (DWFQ) for 802.11 WLANs. In DWFQ, each station maintains a label L, defined as $L = r/w$, where r is bandwidth experienced by the station and w is its assigned weight. The label is included in the header of each outgoing packet. Stations listen to every packet. For each observed packet, if the station's own label is smaller than the observed label, the station decreases its CW by a small amount; otherwise, it increases its CW. The basic idea behind this dynamic adjustment is that the smaller the CW, the higher the throughput. Compared to the current 802.11 MAC protocol, this algorithm is more complex as it requires that the station listens to all packets in the network. In addition, as an adaptive algorithm, the stability and efficiency of the system highly depends on the appropriate choice of parameters, which is a non-trivial task.

An important problem in both DFS and DWFQ is that additional fields need to be inserted into the header of MAC frames.[1] Unlike the Internet Protocol (IP), the 802.11 MAC frame header does not include optional fields to accommodate additional information. Thus, legacy 802.11 devices will not understand the MAC frame format of new devices when they communicate with each other, which results in a *backward compatibility* problem. Due to the widespread deployment of 802.11 WLANs, it is crucial that new devices seamlessly communicate with legacy devices.

We now propose a new algorithm, Distributed Deficit Credit (DDC), to achieve proportional sharing of bandwidth in IEEE 802.11 wireless LANs. Based upon a verified assumption, we will prove that under ideal channel conditions, long-term throughput fairness is achieved. DDC is robust to moderate levels of transmission errors. In addition, DDC does not require any changes to the MAC frame format, which allows legacy 802.11 stations to seamlessly coexist with the DDC-enhanced stations (i.e., devices implementing the DDC algorithm).

IV. Distributed Deficit Credit

The objective of DDC is to achieve *long-term* proportional sharing of bandwidth in a distributed environment. We consider a single-hop 802.11 WLAN, where all the stations are within the same neighborhood and can hear each other, i.e., hidden terminal problems are rare. To simplify our discussion, we first consider ideal channel conditions, i.e., the case when the channel is error-free with no capture effects. Considering the short range of a typical single-hop wireless LAN, this assumption is a reasonable approximation. In Section IV-C, we will discuss how to handle channel errors and capture effects.

A. Preliminaries

DDC is based upon two key ideas: (i) using the notion of "credit," adapted from the Deficit Round Robin (DRR) [7] scheduling mechanism, and (ii) exploiting the 802.11 DCF, which inherently exhibits long-term fairness in channel access. We briefly describe these two ideas in this section.

1) Deficit Round Robin (DRR): In DRR, the scheduler associates with each flow a *deficit counter* initialized to zero, and a value *quantum*. The scheduler serves a *quantum* of bits from each flow. For each head-of-line packet, if its size is smaller than the *deficit counter* + *quantum*, it is served and the deficit counter is reduced by the packet size. Otherwise, the packet remains in the queue, and the value of *quantum* is added to the *deficit counter* of the flow.

The throughput of each DRR flow has been proven to be asymptotically proportional to its *quantum* [7]. One interesting feature of DRR is that it only requires local information, which easily lends itself to a distributed implementation.

2) IEEE 802.11 DCF Long-term Fairness in Channel Access: As described in Section II, all stations within the same IEEE 802.11 neighborhood compete to access the channel. At a given time instance, a station can gain access to (i.e., win) the channel, depending on its own as well as other stations' backoff phase. For example, if a station has experienced numerous collisions and increased its CW to CWmax, then in the short time period that immediately follows, it may have a lower opportunity than others to access the channel. In other words, the 802.11 DCF is unfair over short time scales. The system, however, exhibits symmetry under ideal channel conditions. In the long run, all stations within the same neighborhood have an equal opportunity of winning the channel. Based upon this observation, we make the following assumption on channel access fairness.

Channel Access Fairness Assumption: Let $N_i(t)$ be the number of times that station i wins the channel in time interval [0,t], $i = 1, \cdots, n$. We assume that

$$\lim_{t\to\infty} \frac{N_1(t)}{t} = \lim_{t\to\infty} \frac{N_2(t)}{t} = \cdots = \lim_{t\to\infty} \frac{N_n(t)}{t}.$$

Our simulation results validate this long-term fairness property, as illustrated in Figure 1. In this scenario, the WLAN includes 10 stations. All stations are backlogged during the simulation. The channel bandwidth is 11 Mbps, and the packet size is 1000 bytes. Figure 1 depicts the average number of channel accesses per second for all stations over 3 different time intervals t. In the figure, when the time interval t is short, the curve oscillates, which implies short-term unfairness. As the time interval length increases, the curve becomes more flat. This result supports our assumption of long-term fairness in channel access.

Clearly, an equal opportunity to access the channel does *not* imply throughput fairness. Given that each station has an equal opportunity to access the channel, if two stations

[1] In DFS, 3 mapping schemes are defined: Linear, EXP and SQRT. In the Linear scheme, packets do not carry additional information. The Linear scheme, however, may result in poor throughput. For this reason, the EXP and SQRT schemes are defined, both of which require each packet to carry a virtual time tag in the frame header.

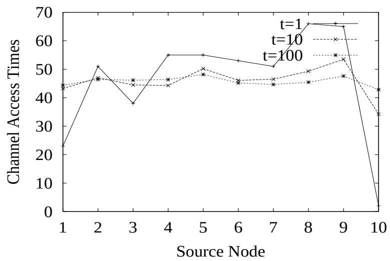


Fig. 1. Long-term fairness in channel access

have different packet sizes, then in the long run, the one with larger packet sizes will have higher throughput. This means that to provide throughput fairness, DDC must be able to accommodate different packet sizes.

B. Distributed Deficit Credit (DDC)

We now describe the DDC algorithm. For simplicity of exposition, we assume that all packets at a station belong to a single flow. In Section IV-C, we will see that the algorithm can be easily extended to support multiple flows at a single station.

Consider a WLAN with *n* stations. A weight w_i is assigned to each station i to indicate the share given to i, $i = 1, \cdots, n$. The minimum possible weight is 1. Each station maintains a variable, $d\_credit_i, i = 1, \cdots, n$, which is initialized to 0. We also select a quantum Q, such that Q is larger than the maximum possible packet size. We will later see that with minimum weight equal to 1, each time a station wins the channel, it can transmit at least one packet.[2]

The channel access scheme is unchanged from the standard 802.11 DCF. This includes channel sensing, computation of the backoff counter, and freezing and resuming the backoff process. The primary difference between DDC and standard 802.11 DCF is when a station wins the channel. In 802.11 DCF, when a station wins the channel, only one data packet is transmitted. In contrast, when a DDC station wins the channel, it can transmit multiple packets without releasing the channel. More specifically, let $bytes_i^k$ be the number of bytes sent out by station i on the k^{th} time it wins the channel. The first time a station wins the channel, it attempts to transmit packets continuously, subject to the restriction that $bytes_i^1 \leq w_i \times Q$. If there are still packets left, then the remaining amount $w_i \times Q - bytes_i^1$ is stored in d_credit_i. Otherwise, d_credit_i is set to zero. The next time the station wins the channel, the amount of traffic it is allowed to send is $d\_credit_i + w_i \times Q$. Similarly, the remaining amount $d\_credit_i + w_i \times Q - bytes_i^2$ is stored in d_credit_i if there are packets left. Otherwise, d_credit_i is reset to 0. This process continues as long as the station has packets to transmit. Figure 2 gives the pseudo-code of DDC for station i.

We now analyze the basic properties of DDC.

[2]An alternative solution is that when the packet size is larger than Q, we fragment it into multiple segments and transmit them one by one. This, however, incurs high implementation complexity, since it requires implementing segmentation/assembly functions in DDC. Therefore, we choose not to fragment packets in DDC.

Initialization: $d\_credit_i = 0$;
When station i occupies the channel:
  $d\_credit_i = d\_credit_i + w_i \times Q$;
  do
    $p = head(i)$;
    if ($size(p) < d\_credit_i$) then
      send (p);
      if ACK received then
        dequeue p and free the buffer;
        $d\_credit_i = d\_credit_i - size(p)$;
      else break;
    else break;
  while ($d\_credit_i > 0$) and (i has packets);
  if no packets backlogged then
    $d\_credit_i = 0$;
  release the channel;

Fig. 2. Pseudo-code for Distributed Deficit Credit (DDC)

Theorem 1: Suppose station i is backlogged during the execution of DDC. Under ideal channel conditions, after the N^{th} time i uses the channel, the difference between $N \times w_i \times Q$ and the total bytes that it has transmitted is bounded by Q.

Proof: Let $d\_credit_i^k$ be the value of d_credit_i after the k^{th} time i uses the channel ($d\_credit_i^0 = 0$). Let $bytes\_i^k$ be the amount of traffic sent by i during the k^{th} time it occupies the channel. From the description of the DDC algorithm, we have

$$d\_credit_i^k + bytes_i^k = d\_credit_i^{k-1} + w_i \times Q.$$

Therefore,

$$\begin{aligned} \sum_{k=1}^{N} bytes_i^k &= N \times w_i \times Q + d\_credit_i^0 - d\_credit_i^N \\ &= N \times w_i \times Q - d\_credit_i^N. \end{aligned}$$

From the algorithm, we know that in order for i to finish using the channel, $d\_credit_i^N$ must be less than the current packet size, which must be less than Q. Therefore, we have

$$|\sum_{k=1}^{N} bytes_i^k - N \times w_i \times Q| < Q.$$

■

Theorem 2: Suppose stations $1, \cdots, n$ are backlogged during the execution of DDC. Let $c_i(t), i = 1, \cdots, n$ be the throughput of station i during time period t. Then, as $t \to \infty$, the average throughput (bytes/second) of station i is proportional to w_i, i.e.,

$$\lim_{t \to \infty} \frac{c_1(t)}{t} : \frac{c_2(t)}{t} : \cdots : \frac{c_n(t)}{t} = w_1 : w_2 : \cdots : w_n.$$

Proof: Let $N_1(t), N_2(t), \cdots, N_n(t)$ be the number of times stations $1, 2, \cdots, n$ win the channel, respectively. As $t \to \infty, N_i(t) \to \infty$, and from Theorem 1, we have

$$|c_i(t) - N_i(t) \times w_i \times Q| < Q.$$

Thus,

$$\begin{aligned}\lim_{t\to\infty}\frac{c_i(t)}{t} &= \lim_{t\to\infty}\{\frac{c_i(t)}{N_i(t)}\times\frac{N_i(t)}{t}\}\\ &= \lim_{t\to\infty}\frac{c_i(t)}{N_i(t)}\times\lim_{t\to\infty}\frac{N_i(t)}{t}\\ &= w_i\times Q\times\lim_{t\to\infty}\frac{N_i(t)}{t}.\end{aligned}$$

From the channel access fairness assumption,

$$\lim_{t\to\infty}\frac{N_1(t)}{t}=\lim_{t\to\infty}\frac{N_2(t)}{t}=\cdots=\lim_{t\to\infty}\frac{N_n(t)}{t}.$$

It is easy to see that

$$\lim_{t\to\infty}\frac{c_1(t)}{t}:\frac{c_2(t)}{t}:\cdots:\frac{c_n(t)}{t}=w_1:w_2:\cdots:w_n.$$

■

Therefore, we have shown that DDC can provide long-term bandwidth allocations in proportion to the station weights.

C. Deployment Considerations

DDC is a fully distributed algorithm. The only additional cost associated with DDC is updating the deficit credit counter. From the algorithm, updating the deficit credit counter is clearly O(1). In this section, we discuss a number of practical issues with DDC implementation and DDC behavior in realistic scenarios.

1) Occupying the Wireless Channel: Choi et al. have proposed Contention Free Burst (CFB) in a draft proposal to the IEEE 802.11e committee [24]. In CFB, a station is allowed to transmit multiple MAC frames as long as the entire transmission time does not exceed a predefined limit.

DDC fits well into this mechanism. In our implementation, we leave a gap of length SIFS between consecutive frames. Since SIFS is the smallest inter-frame space, this will prevent other stations from accessing the channel and its continuous occupation. From the pseudo-code, it is clear that one cannot transmit more than $(w+1)\times Q$ bytes during one transmission, which prevents one station from occupying the channel for too long.

2) Multiple Flows per Station: Thus far, we have assumed that all packets at a station belong to a single flow. In practice, the same station may have multiple active flows, each of which with a different weight assigned to it. To accommodate this case, we modify DDC as follows.

Consider a station having n active flows with weights $W_1,\cdots,W_n$. We set the weight of the station to be $W=\sum_{i=1}^{n}W_i$. A DRR scheduler is used at the station with weights $W_1,\cdots,W_n$. In this manner, the total bandwidth a station receives is proportional to the sum of the flow weights, and the bandwidth is further divided among multiple flows in proportion to their weights.

3) Impact of Non-ideal Channel Conditions: In our previous discussion, we have assumed ideal channel conditions, i.e., an error free channel and no capture effects. We now consider the effects of transmission errors and capture effects.

Transmission Errors: The effect of transmission errors is twofold. First, if the channel for one station is significantly worse than that of others, then more of its packets may be lost due to transmission errors. Due to this, the deficit credit counter cannot be increased as frequently as other stations, which means this station receives lower credit than other stations. To address this problem, we can use the RTS/CTS access mechanism. Since the RTS/CTS frame is very short, the possibility that the RTS/CTS frame is corrupted is quite low, which helps alleviate the problem.

Second, when a station has successfully occupied the channel, frames (whether RTS/CTS/DATA/ACK frames) may be lost and the station cannot finish transmitting all its packets. This means the station cannot use its credit. To address this problem, the value of *d_credit* is not reduced until the ACK is received. Therefore, if a packet gets corrupted during transmission, the credit is maintained for later use. In our simulations, we have studied the performance of DDC in the presence of transmission errors. Our results show that DDC is robust to moderate levels of transmission errors (bit error rate = 10^{-6}).

The problem of transmission errors is mitigated by using error correction codes. In the draft specifications of IEEE 802.11e [25], a (224, 208) shortened Reed Solomon Code is proposed, which splits the MSDU (MAC Service Data Unit) into multiple blocks no larger than 208 bytes each, and then encodes each block. Each block can correct up to 8 bytes of error. Therefore, most of the channel errors can be corrected by this code.

Capture Effects: Capture effects have been observed in wireless environments [26]. Among competing connections, the one with the strongest SNR is able to capture the channel. Studies of 802.11 [26] show that the capture effect is prevalent in a hidden terminal scenario. In contrast, capture effects are relatively minor in single-hop scenarios. Since our primary focus is on a single-hop WLAN, where the stations are in the same neighborhood and share the same channel, the capture effect is minimal. To completely compensate for capture effects, additional power control mechanisms may be necessary.

V. Performance Evaluation

In this section, we investigate the performance of the DDC algorithm. We simulate DDC using a modified version of the ns-2 simulator [27]. The DDC algorithm is incorporated into the current implementation of 802.11 MAC DCF. We simulate a WLAN with $n+1$ stations from station 0 (the access point) to station n, where $n\leq 100$. We have *n* flows where each flow i is from station i to station $0, i=1,2,\cdots,n$.

We use the following parameters unless otherwise specified: (1) channel bandwidth is 11 Mbps, (2) packet size is 1000 bytes, which is the length of the MSDU and does not include the MAC layer header and physical layer header, (3) quantum Q is 1200 bytes, (4) all flows are backlogged at the MAC layer (this simplifies the interpretation of the results), (5) simulation time is 100 seconds (to study the long-term behavior of the algorithm), (6) the RTS/CTS mechanism is used, since it increases bandwidth efficiency in case of collisions, (7) *n* is 10, which corresponds to a typical WLAN. We use Direct-

Sequence Spread Spectrum (DSSS) for multiple access. Table I summarizes the parameters used in the simulations.

TABLE I
SIMULATION PARAMETERS

CWmin	CWmax	ACK	CTS	RTS	slot	SIFS	DIFS
32	1024	38B	38B	44B	20μs	10μs	50μs

A. Convergence of Bandwidth Allocations

We first consider the simple case when all *n* flows have identical weights, i.e., $w_1 = w_2 = \cdots = w_n = 1$. Figure 3 shows the average throughput (in bytes/second) for all 10 stations at different time scales.[3] Ideally, the curve should be completely flat. As we can see from the figure, when the time scale is small, e.g., $t = 1$, the curve significantly oscillates, which denotes short-term unfairness. As the time scale increases, the curve becomes more flat, which shows that the DDC algorithm behaves as expected.

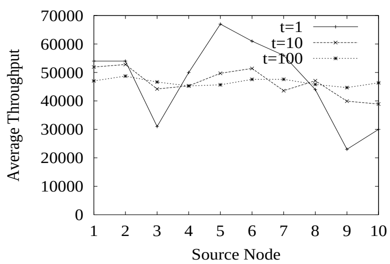


Fig. 3. Convergence of bandwidth allocation

B. Proportional Allocation of Bandwidth

We now study the performance of DDC with different flow weights. The weights of stations $1, 2$, and 3 are set to $8, 4$, and 2, respectively, while the weights of all other stations are set to 1. Figure 4 depicts the *average throughput/weight* ratio for all the stations. It can be seen that the *average throughput/weight* ratio of all stations is quite similar. We have also simulated the situation where the flows have different packet sizes, and have observed similar results. Thus, DDC achieves proportional allocation of bandwidth, and the performance is independent of packet sizes.

C. Effect of the Quantum Q

An important parameter in DDC is the quantum Q. We set Q to 3 values: 1200, 3000, or 10000, to study its effect on aggregate throughput and fairness. All 10 stations have identical weights of 1. Table II lists the aggregate throughput (in bytes/second), and figure 5 depicts the *average* throughput for different values of Q. Observe that for larger values of Q, the aggregate throughput becomes larger. The figure, however,

[3]In our experiments, we have also simulated the original 802.11 MAC for comparison. Results (not included here for brevity) have shown that DDC achieves higher throughput than the original 802.11 MAC. Thus, in the discussion, we focus on the fairness performance.

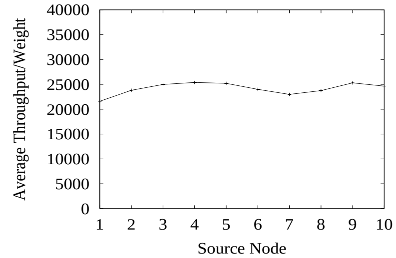


Fig. 4. Performance with variable flow weights

shows that the curve oscillates and fairness is degraded. This is because when Q is large, a station can hold the channel for a long time before it releases the channel. Therefore, a relatively shorter time is wasted by idle slots and collisions, which results in a more efficient use of the channel. In contrast, given a fixed time interval, a larger Q means that each station accesses the channel fewer times on the average, which makes the effect of any difference among stations more pronounced. Thus, the choice of Q exhibits a tradeoff among efficiency and fairness.

TABLE II
AGGREGATE THROUGHPUT FOR DIFFERENT VALUES OF Q

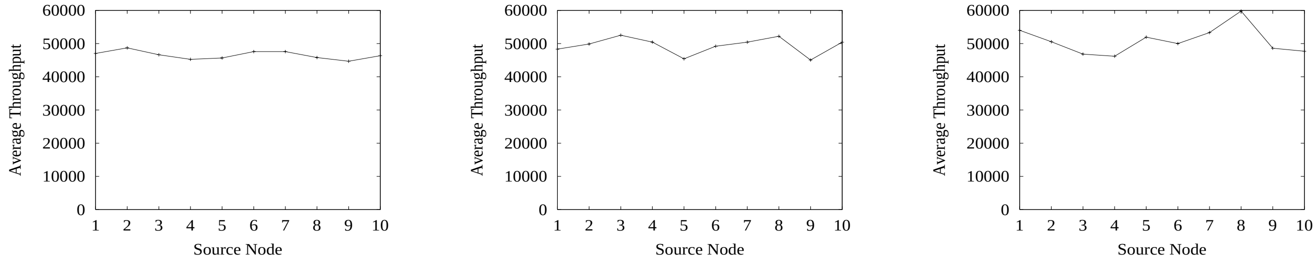
Q	1200	3000	10000
Aggregate throughput	465320	493920	508920

D. Impact of the Number of Stations

We now study the performance of DDC for different number of stations. Figure 6 illustrates the average throughput (in bytes/second) for all the stations, when the number of stations $n = 5, 10$, or 100. All stations have identical weights of 1. It can be seen that the curve oscillates more for larger *n*. The reason for this is that when there are more stations, each individual station will receive lower throughput. Thus, if there is a difference between the throughput of two stations, the relative deviation between the two stations is non-trivial. A larger number of stations may exhibit short-term unfairness and require a longer time scale to converge. However, as seen from the figure, even when $n = 100$, the throughput of all stations still centers around the average value, which shows that the asymptotic behavior of DDC is fair.

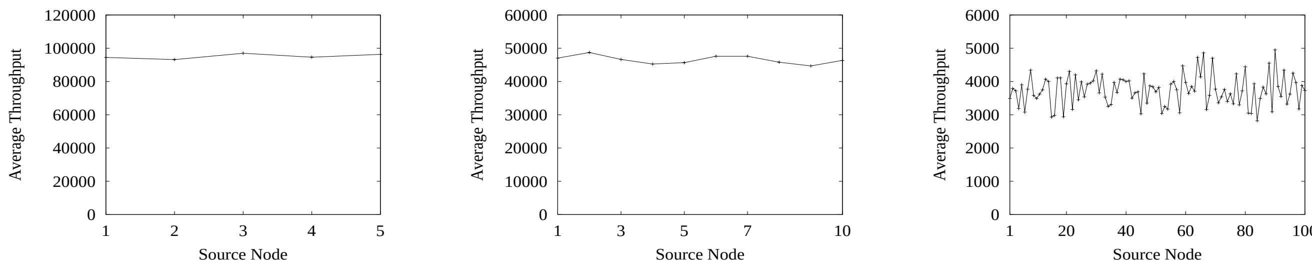
E. Independence of Packet Size

We now investigate the effect of different packet size on fairness. We still use a WLAN of 10 stations and all the stations have identical weights of 1. The packet sizes of the first 2 stations are set to 100 bytes, while for all the other stations, the packet size is 1000 bytes. As seen from Figure 7, the average throughput (in bytes/second) received by the first 2 stations is quite close to that of the other stations. We have also simulated the situation when the packet sizes of a flow exhibit a bi-modal distribution, and have observed similar results (results not shown here for brevity). Therefore, DDC performance is independent of packet sizes.



(a) Q = 1200 (b) Q = 3000 (c) Q = 10000

Fig. 5. Effect of the Quantum Q



(a) n = 5 (b) n = 10 (c) n = 100

Fig. 6. Impact of the number of stations n

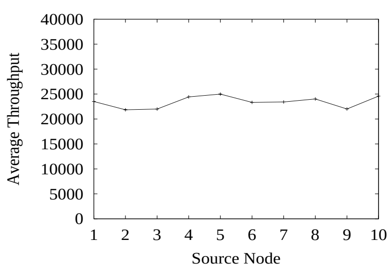


Fig. 7. Independence of packet size

F. Performance of DDC under Bursty Traffic

In all the cases discussed so far, we have assumed that the user flows are backlogged at the MAC layer throughout the simulation period. In practice, the user flows may be bursty at the application (e.g., HTTP) and/or transport (e.g., TCP) levels. Therefore, we study the performance of DDC under bursty traffic sources.

We first study TCP flows. Stations $1, 2, \cdots, n$ are sending data to station 0 using TCP. The weights of stations 1 and 2 are set to 4 and 2 respectively, while the weights of all other stations are set to 1. Figure 8(a) shows the *average throughput/weight* ratio for all the stations. Though the curve exhibits slight oscillations, the *average throughput/weight* ratio for all stations is approximately equal, which shows that DDC still achieves proportional bandwidth allocation for TCP flows.

We then study the case where TCP and UDP flows coexist. In Figure 8(b), stations $1, 2, \cdots, 5$ use TCP and stations $6, 7, \cdots, 10$ use greedy UDP. The weights of stations 1 and 2 are set to 4 and 2 respectively, while the weights of all other stations are set to 1. We observe that UDP flows achieve significantly higher *average throughput/weight* ratio than TCP flows. This can be attributed to the unresponsive nature of the UDP flows. During congestion, the congestion control mechanism will decrease the TCP congestion window sizes, while UDP flows remain unaffected.

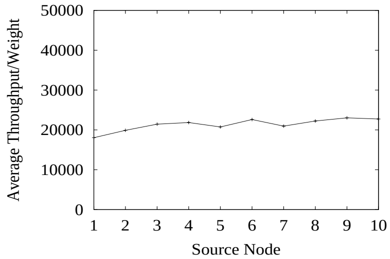
VI. Concluding Remarks

In this paper, we have proposed a new algorithm, Distributed Deficit Credit (DDC), for proportional bandwidth allocation in IEEE 802.11 WLANs. The algorithm is easily implemented as a simple modification of the IEEE 802.11 DCF. Unlike previous work on fair scheduling in 802.11 WLANs (e.g., DFS and DWFQ), DDC uses a credit-based approach to provide long-term throughput fairness. Another appealing feature of DDC is that it does not require any changes to the MAC frame format, which allows legacy 802.11 stations to seamlessly coexist with the DDC-enhanced stations. This makes DDC easily deployable.

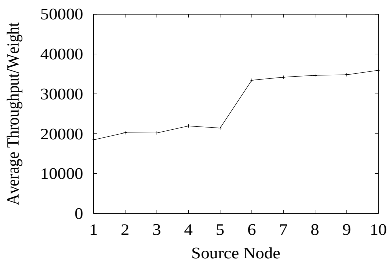
Our analysis and simulation results have shown that DDC indeed allocates bandwidth in proportion to the weights of the flows sharing the channel. The performance is independent of packet sizes. An interesting tradeoff exists between fairness and efficiency, which can be balanced by appropriately tuning the quantum Q.

A number of open issues remain, including:

- Supporting real time services: We have considered the problem of throughput fairness for services without real-time constraints. For real time services, e.g., Voice over



(a) TCP flows



(b) TCP and UDP flows

Fig. 8. Performance with bursty traffic

IP (VoIP), delay and jitter must be considered. The DDC algorithm needs to be extended to take delay and jitter into consideration.

- Multi-rate WLANs: Heusse et al. [28] have observed that in multi-rate WLANs, when certain mobile hosts use a lower bit rate than others, the performance of all hosts is considerably degraded. To address this problem, fairness in channel occupation time is required. We are currently investigating this problem.

ACKNOWLEDGMENTS

The authors would like to thank Professors Ananth Grama, Zhiyuan Li, and Ness Shroff (Purdue University) for their valuable comments on this work.

REFERENCES

[1] "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications," 1997.

[2] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single node case," *IEEE/ACM Transactions on Networking*, pp. 344–357, June 1993.

[3] A. Demers, S. Keshav, and S. Shenker, "Design and analysis of a fair queuing algorithm," in *Proc. of the ACM SIGCOMM*, 1989.

[4] S. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *Proc. of the IEEE INFOCOM*, 1994.

[5] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switching networks," in *Proc. of the ACM SIGCOMM*, 1990.

[6] P. Goyal, H. Vin, and H. Chen, "Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks," in *Proc. of the ACM SIGCOMM*, 1996.

[7] M. Shreedar and G. Varghese, "Efficient fair queuing using deficit round robin," in *Proc. of the ACM SIGCOMM*, 1995.

[8] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," in *Proc. of the ACM SIGCOMM*, 1997.

[9] X. Liu, E. K. P. Chong, and N. B. Shroff, "Transmission scheduling for efficient wireless utilization," in *Proc. of the IEEE INFOCOM*, 2000.

[10] ——, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, pp. 451–474, March 2003.

[11] D.-J. Deng and R.-S. Chang, "A priority scheme for IEEE 802.11 DCF access method," *IEICE Transactions on Communications*, vol. E82-B, no. 1, January 1999.

[12] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proc. of IEEE INFOCOM*, 2001.

[13] A. Veres, A. T. Campbell, M. G. Barry, and L.-H. Sun, "Supporting service differentiation in wireless packet networks using distributed control," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, October 2001.

[14] Y. Xiao, "A simple and effective priority scheme for IEEE 802.11," *IEEE Communication Letters*, vol. 7, no. 2, February 2003.

[15] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over wireless LAN," in *Proc. of IEEE INFOCOM*, 2003.

[16] S. Kim, B. Kim, and Y. Fang, "Downlink and uplink resource allocation in IEEE 802.11 wireless LANs," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 1, January 2005.

[17] J. Dunn, M. Neufeld, A. Sheth, D. Grunwald, and J. Bennett, "A practical cross-layer mechanism for fairness in 802.11 networks," in *Proc. of the First International Conference on Broadband Networks*, 2004.

[18] C. Coutras, S. Gupta, and N. B. Shroff, "Scheduling of real-time traffic in IEEE 802.11 wireless LANs," *Wireless Networks*, vol. 6, no. 6, November 2000.

[19] O. Sharon and E. Altman, "An Efficient Polling MAC for Wireless LANs," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, August 2001.

[20] J. Y. Yeh and C. Chen, "Support of Multimedia Services with the IEEE 802.11 MAC Protocol," in *Proc. of IEEE ICC*, 2002.

[21] T. Suzuki and S. Tasaka, "Performance Evaluation of Priority-based Multimedia Transmission with the PCF in an IEEE 802.11 Standard Wireless LAN," in *Proc. of IEEE Symposium on Personal, Indoor, and Mobile Radio Communications*, 2001.

[22] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Proc. of the ACM Mobicom*, 2000.

[23] A. Banchs and X. Perez, "Distributed weighted fair queuing in 802.11 wireless LANs," in *Proc. of the IEEE ICC*, 2002.

[24] S. Choi, J. del Prado, A. Garg, M. Hoeben, S. Mangold, S. Shankar, and M. Wentink, "Multiple Frame Exchanges during EDCF TXOP, IEEE 802.11-01/566r3," January 2002.

[25] "IEEE 802.11e/D3.0, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," May 2002.

[26] C. Ware, J. Judge, J. Chicharo, and E. Dutkiewicz, "Unfairness and capture behaviour in 802.11 adhoc networks," in *Proc. of the IEEE ICC*, 2000.

[27] "Network Simulator (ns) version 2," http://www-mash.cs.berkeley.edu/ns.

[28] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," in *Proc. of IEEE INFOCOM*, 2003.