Achieving Fairness in IEEE 802.11 DFWMAC with Variable Packet Lengths

Yu Wang
Department of Computer Engineering
University of California
Santa Cruz, CA. USA
ywang@cse.ucsc.edu

Brahim Bensaou
Department of Computer Science
Hong Kong Univ. of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
brahim@cs.ust.hk

Abstract—**The Medium Access Control (MAC) protocol through which mobile stations can share a common broadcast channel is essential in an ad-hoc network. Due to the existence of hidden terminals and partially-connected network topology, contention among stations in an ad-hoc network is not homogeneous. Some stations are at a disadvantage in access to the shared channel and can suffer severe throughput degradation when load to channel is high. This is known as the "fairness problem". Existing MAC protocols like IEEE 802.11 Distributed Foundation Wireless Medium Access Control (DFWMAC) may exacerbate this problem as it uses the binary exponential backoff (BEB) algorithm in contention resolution, which always favors the last succeeding station. This paper reviews the "fairness index", which is a metric to quantify fairness, and proposes a new estimation based backoff algorithm for the IEEE 802.11 DFWMAC protocol. The new algorithm can support the case when packet lengths are variable, which is a typical scenario of IEEE 802.11 compliant implementations that include both the basic CSMA/CA access method and the RTS/CTS access method. Simulation results show that the fairness problem can be very severe with the original BEB algorithm when packet length is variable and our new backoff algorithm can achieve far better fairness without adding much in complexity.**

I. INTRODUCTION

There has been a growing interest in ad-hoc networks in recent years [1] as they can be deployed rapidly without reliance on existing network infrastructure and are self-configurable.

Usually all the stations in an ad-hoc network share a common broadcast channel and an efficient and effective medium access control (MAC) protocol is essential as the shared channel is a scarce resource. "Hidden terminal" problem [2] is prominent in ad-hoc networks and can degrade throughput dramatically if not addressed properly. Therefore, various distributed MAC protocols have been proposed in the literature [3–6] to enhance throughput. Among them, Distributed Foundation Wireless Medium Access Control (DFWMAC) protocol is the MAC protocol used in IEEE 802.11 recommended standard for wireless ad-hoc and infrastructure LANs. DFWMAC is a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol that provides both a mandatory basic access method and an optional RTS/CTS based access method. The basic access method includes only exchange of data packet and acknowledgment packet between a pair of source and destination stations and is suitable for short data packets. RTS/CTS access method is used to combat the hidden terminal problem by requiring that a short Request-to-Send (RTS) and Clear-to-Send (CTS) packet should be exchanged successfully between a pair of sender and receiver before actual data packet transmission begins. To avoid unnecessary protocol overhead, only when data packet length exceeds the value of a parameter called $RTS\_THRESHOLD$ is RTS/CTS access method used. As DFWMAC can alleviate hidden terminal problem and enhance throughput together with its status as a recommended international standard, it is very popular in routing research for ad-hoc networks.

DFWMAC protocol still suffers from the fairness problem which was first investigated by Bharghavan et al. [4]. Fairness problem refers to the situation when some stations may be at a disadvantage in access to the shared channel and will suffer severe throughput degradation when load to channel is high, resulting in unfair share of channel bandwidth. Due to the lack of central administration and partially-connected network topology, fairness problem is prominent in ad-hoc networks as shown in [7].

In this paper, we will provide a simple yet effective approach to address the fairness problem in a DFWMAC system where packets lengths are variable, and where both basic access and RTS/CTS access methods are concurrently activated. The rest of the paper is organized as follows. Section II reviews the related work that has been done to address the fairness problem. Section III advances the work done in [8] and proposes a new estimation based backoff algorithm which requires little modification to the original backoff algorithm used in DFWMAC protocol and can support the case when packet length is variable. Section IV evaluates the performance of the new backoff algorithm via computer simulations and shows that it can achieve far better fairness than the original DFWMAC protocol even when packet length is variable. Section V concludes this paper.

II. RELATED WORK

Bharghavan et al. [4] pioneered the work in addressing the fairness problem. They stated that the commonly used binary exponential backoff (BEB) in multiple access protocols (including DFWMAC) for collision resolution can aggravate fairness problem in a distributed wireless network as it always favors the last succeeding station. This can be explained as follows. According to BEB algorithm, a station's contention window[1] size will be doubled after each unsuccessful transmission until prescribed maximum value and will return to the minimum value if a data packet is successfully transmitted. Therefore a station that has last succeeded in transmitting a data packet will reset its contention window to minimum value, thus has statistically shorter backoff timer than other stations that have failed. When

---

[1] Backoff timer is generated from a uniform distribution which ranges from 0 to the size of the current contention window.

load to channel is high, these other stations may suffer excessive access delay and severe throughput degradation. Therefore, they proposed MACAW (short for Multiple Access with Collision Avoidance for Wireless) protocol to address the problem.

In MACAW, additional control packets and a different backoff algorithm named Multiplicative Increase and Linear Decrease (MILD) with a backoff copy scheme were used to increase throughput and alleviate fairness problem. In addition, "per stream" fairness was introduced in MACAW. It means that each stream that originates from either the same station or different stations should be treated equally and given equal share of the channel capacity. This is different from "per station" fairness which accords channel capacity to individual stations instead of individual streams. For multiple streams that originate from a station, MACAW keeps separate queues for each stream and runs backoff algorithms independently for each stream. In a wireless ad-hoc LAN where all traffic normally goes through an Access Point (AP) which interfaces the whole LAN to the wireline backbone network, "per stream" fairness is useful as the AP can handle more traffic for the surrounding mobile stations. However, the backoff copy scheme used in MACAW only works when congestion is homogeneous which is not necessarily the case in ad-hoc networks and MACAW did not solve all the fairness problems in the scenarios that were investigated in [4].

Ozugur et al. [9] also proposed a p_{ij}-persistent CSMA based backoff algorithm to address the fairness problem. This paper first defined the fairness index to be the ratio of maximum link throughput to minimum link throughput. Then it proposed that each station calculates a link access probability p_{ij} for each of its links based on the number of connections from itself and its neighbors (connection based), or based on the average contention period of its and other stations' individual links (time based). Whenever its backoff period ends, station i will send RTS packet to j with probability p_{ij} or back off again with probability $1 - p_{ij}$. The proposed scheme relies on periodic broadcast packets in the time-based approach or on aperiodic broadcast packets in the connection-based approach whenever the network topology changes. However, none of the schemes can achieve the best results for all the network configurations investigated in [9] and sometimes the best results are achieved when these schemes are in fact not used. In addition, broadcast packets are unreliable to disseminate information to neighbors as the sender cannot elicit acknowledgment packets from its neighbors to ascertain if its packet has been successfully received by all of them.

In the ongoing research work of Vaidya et al. [10], they identified the difficulties in defining fairness itself in multi-hop networks and defined a Generalized Resource Sharing (GPS) algorithm which needs further investigation as it includes sorting flows which requires global information and implementation of a practical approximation is still an open research problem. In addition, a distributed fair scheduling algorithm was also proposed to achieve fairness on local area networks (one hop) and its performance was evaluated.

III. ESTIMATION BASED BACKOFF ALGORITHM

In the preliminary work done in [8], fairness was defined similar to the fair queueing defined in [11]. A fairness index FI was introduced to measure the degree of fairness, which was shown as follows:

$$FI = max\{\forall i, j : max(\frac{W_i}{\phi_i}, \frac{W_j}{\phi_j})/min(\frac{W_i}{\phi_i}, \frac{W_j}{\phi_j})\}$$

where ϕ_i (ϕ_j) stands for a pre-defined fair share that station i (j) should receive and W_i (W_j) is the actual throughput achieved by station i (j). The fairness goal thus becomes to minimize FI.

There comes the problem of how to assign ϕ for individual station in a distributed networks. In fact, this is a problem similar to Call Admission Control (CAC), which is beyond the scope of the paper. In [12] we propose a distributed algorithm to assign fair shares to different links in the network with the objective of achieving max-min fairness. Note that the major problem when assigning fair shares is to determine how many stations are sharing the link in any given neighborhood, and which stations are interfering with which others: this is not always possible due to the hidden terminal problem. A simple workaround to this problem when an ad-hoc network is open to everyone and all stations are trusted and known not to misbehave can be done like this. Each station regards all of its neighbors as a whole entity and has only the notion of "myself" and "others". If a station i wants to get the same fair share of the channel bandwidth with other stations, it can just set $\phi_i = 0.5$. According to its reasoning, "others" gets $\phi_o = 1 - \phi_i = 0.5$ which equals to its own share ϕ_i. This can be interpreted as per-station fairness. If a station has two active links (or streams in MACAW's terminology), as in the case when the station acts as a router, it can set ϕ_i to meet:

$$\frac{\phi_i}{\phi_o} = \frac{\phi_i}{(1-\phi_i)} = \frac{2}{1} \Rightarrow \phi_i \approx 0.67,$$

which simply shows that the station (router) wants each of its streams to be treated equally with other streams. This can be interpreted as MACAW's per-stream fairness. Using per-stream fairness for a station functioning as a router can be beneficial for an ad-hoc network as the router can handle much more packets despite the disadvantage that it has only one network interface to route packets.

In [8], we proposed a first estimation based backoff algorithm that proves to achieve far better fairness than the original BEB algorithm used in IEEE 802.11 DFWMAC protocol. However, it has not considered the cases when RTS/CTS access method is not used (like short data packets transmitted during a Telnet session for instance) and the case where packet length is variable. In fact, previous MAC protocols like MACAW and p_{ij}-persistent CSMA did not consider variable length packet and depended on RTS/CTS packet to carry extra information these protocols needed. Thus the approach of both algorithms would fail to provide fair access in a DFWMAC based system when the optional RTS/CTS access method is not implemented. To solve the problem in a practicable approach, in this section we propose a modified backoff algorithm for DFWMAC protocol where each station estimates its share and other stations' share of the channel and then adjusts its contention window accordingly. The new algorithm supports variable length packet and works as well when basic access method is used. We introduce the following notation to facilitate description of the algorithm:

W_{ei} : The estimated share of the estimating station itself;
W_{eo} : The estimated share of other stations;
T_{type} : Time to transmit a packet of type *type*;
T_{edata} : When receiving an ACK packet, the estimated transmission time of the corresponding data packet.

Algorithm 1 shows how the estimation works, through packet sensing.

Algorithm 1 Fair share estimation

```
switch (received packet type) {
case RTS:
    if (destID != localID) {
        $W_{eo} += T_{rts}$; update $T_{edata}$
    } else {
        send CTS packet; $W_{eo} += (T_{rts} + T_{cts})$
    }
case CTS:
    if (destID != localID) {
        $W_{eo} += (T_{rts} + T_{cts})$; update $T_{edata}$
    } else {
        send DATA packet;
        $W_{ei} += (T_{rts} + T_{cts} + T_{data})$
    }
case DATA:
    if (destID != localID) {
    if ($T_{data} > RTS\_THRESHOLD$) {
            $W_{eo} += (T_{rts} + T_{cts} + T_{data})$
        } else {
            $W_{eo} += T_{data}$
        }
        $T_{edata} = T_{data}$
    } else {
        send ACK packet;
        if ($T_{data} > RTS\_THRESHOLD$) {
            $W_{eo} += (T_{rts} + T_{cts} + T_{data} + T_{ack})$
        } else {
            $W_{eo} += (T_{data} + T_{ack})$
        }
    }
case ACK:
    if (destID != localID) {
        if ($T_{edata} > RTS\_THRESHOLD$) {
            $W_{eo} += (T_{rts} + T_{cts} + T_{edata} + T_{ack})$
        } else {
            $W_{eo} += (T_{edata} + T_{ack})$
        }
    } else {
        if ($T_{data} > RTS\_THRESHOLD$) {
            $W_{ei} += (T_{rts} + T_{cts} + T_{data} + T_{ack})$}
        } else {
            $W_{ei} += (T_{data} + T_{ack})$
        }
    }
}

When an RTS packet is sent, $W_{ei} += T_{rts}$.
When a DATA packet is sent without
RTS/CTS, $W_{ei} += T_{data}$.
```

The basic idea is that whenever a station receives a packet, it will update its estimation of either its own share or others' share based on type and role of the packet. For example, if a station receives a CTS packet destined to itself, the station sends a data packet and updates estimation of its own share as the data packet transmission request was originated by itself. Additionally, RTS and CTS packets' transmission time is also counted toward its estimated share because they are used as a channel reservation scheme and consume channel resource as well.

In fact, it is relatively easy for a station to adjust its estimation whenever it receives an RTS, CTS or DATA packet as these packets carry sufficient information for the station to judge. However, the main difficulty lies in how a station should do when it receives an ACK packet, as the ACK packet does not carry any information about the length of the corresponding DATA packet and thus this station does not know if RTS/CTS access method is used for the DATA packet if its length is unknown. To solve this problem, we use a simple approach based on the reasoning of packets' time affinity, which simply means that the newly received ACK packet destined to others may be closely related with the previous received RTS/CTS/DATA packet destined to others. In DFWMAC, an RTS/CTS packet includes a field in its packet header named Network Allocation Vector (NAV) which notifies other neighboring stations how long they should defer access to the channel such that the following DATA and ACK packet transmissions can be completed without interference from these stations. If a station receives an RTS/CTS packet destined to other stations, it can derive the length of the corresponding DATA packet and set its transmission time to T_{edata}. If a station receives a DATA packet destined to other stations, it already knows the length of the DATA packet and update T_{edata}. When the station receives an ACK packet destined to other stations, it can then adjust its estimation based on the value of T_{edata}.

Based on the estimation algorithm, we modify the binary exponential backoff algorithm used in DFWMAC. We define the estimated fairness index to be: $FI_e = (\frac{W_{ei}}{\phi_i})/(\frac{W_{eo}}{\phi_o})$ and the adjustment of contention window is shown in Algorithm 2.

Algorithm 2 shows that if a station estimates that it has got more share than it should get, it will double its contention window size until it reaches the maximum value ($CWMAX$) so that its neighbors can have more chances to recover earlier from backoff procedure and win access to the channel and vice versa. If a station estimates that it has got only its fair share, it will hold onto its current contention window size. In this algorithm, C is a constant used to adjust the adaptability of the algorithm. The

Algorithm 2 Contention window adjustment

```
switch $(FI_e)$ {
case $> C$:
    $CW_{new} = min(CW_{new} \times 2, CWMAX)$
case $(1/C, C)$:
    $CW_{new} = CW_{old}$
case $< 1/C$:
    $CW_{new} = max(CW_{old} \;/\; 2, CWMIN)$
}
```

RTS	CTS	ACK
25-byte	20-byte	20-byte

CWMIN	CWMAX	C	backoff unit time
31	1023	1	6μsec

DIFS	SIFS	EIFS
12μsec	0μsec	1.3msec

TABLE I

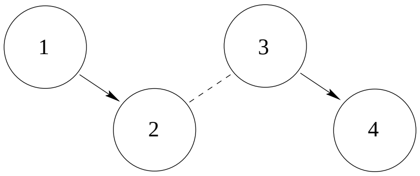
PROTOCOL CONFIGURATION PARAMETERS

smaller the value of C, the more aggressively is the contention window size adjusted and vice versa. However, the choice of C is rather limited. For example, if we choose $C = 2$, stations would not change their contention windows when estimated FI is between $(0.5, 2)$ and the probability of collision may be high when the number of competing stations is large and/or the load offered to the channel is high. Influence of C will be investigated later.

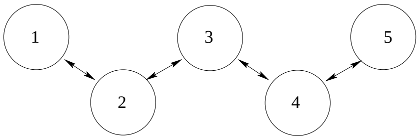
IV. SIMULATION RESULTS AND DISCUSSIONS

In our experiments, we investigate some configurations of wireless ad-hoc networks used in MACAW, p_{ij}-persistent CSMA and [8]. These are the 4-station, 5-station and 6-station scenarios. They are shown in Fig. 1, where arrow lines indicate that there is traffic between stations and dashed lines indicate that the stations are within communication reach of each other but no traffic flows between them.

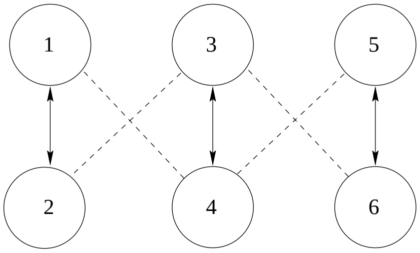
We assume a 1Mbps ideal channel with zero preamble and processing overhead and a propagation delay of about 6μ seconds. We have performed different sets of simulations with OPNET Modeler/Radio and we compare our results with the original IEEE 802.11 DFWMAC protocol[2]. Table I lists the parameters used to generate the simulations results. As we ignore the extra time incurred by hardware and software, the different InterFrame Spaces (IFSs) in IEEE 802.11 are reduced accordingly and they are also shown in Table I. Unless otherwise specified, all stations will use a fair share $\phi = 0.5$. As to the size of data packet, we choose two types. One is short data packet with total length of 50-byte. The other is long data packet with total length of 500-byte. For the short data packet, RTS/CTS access method is not used. These two types may represent Telnet and FTP traffic commonly carried in IP based networks. The data packet generator at each station generates a short packet and a long one with equal probability.



(a) 4-station

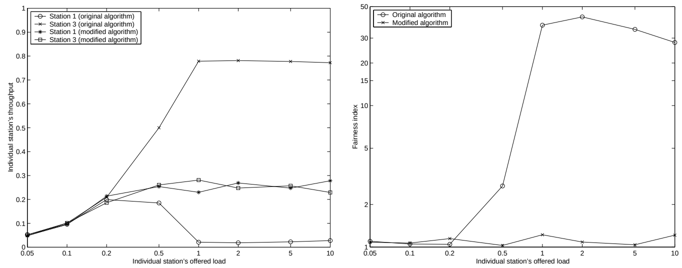


(b) 5-station



(c) 6-station

Fig. 1. Network configurations



(a)

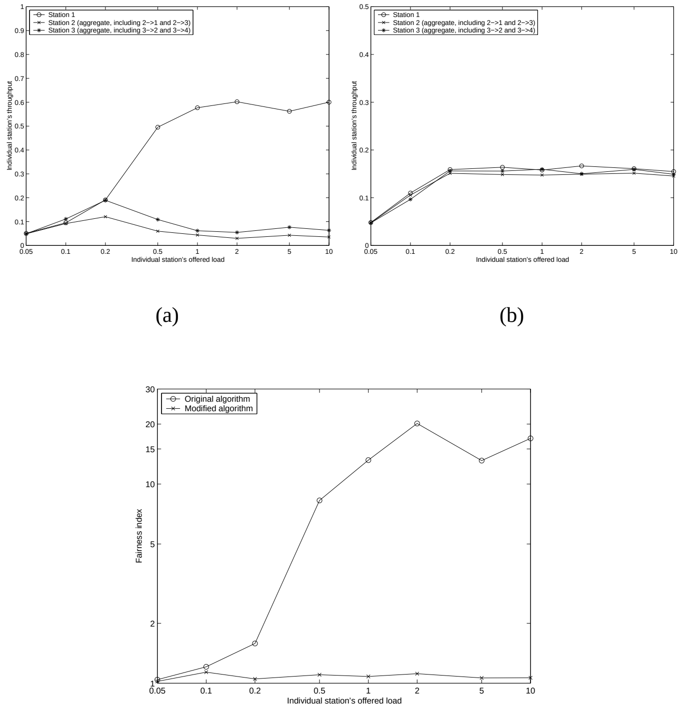
(b)

Fig. 2. (a) Station throughput, (b) fairness index versus station's offered load for the 4-station scenario – variable packet length.

In the 4-station scenario, station 1 and 3 generate Poisson traffic with the same mean rate, and results are shown in Fig. 2. The figure shows that DFWMAC will have serious fairness problem when the offered load is high enough, which can be explained as follows. Most of the time station 1's transmission may coincide with station 3's transmission as they are hidden from each other. Station 2 will not be able to receive station 1's packet due to station 3's concurrent transmission. However, station 4 can

---

[2] We use its specification for Direct Sequence Spread Spectrum when applicable.



(a)

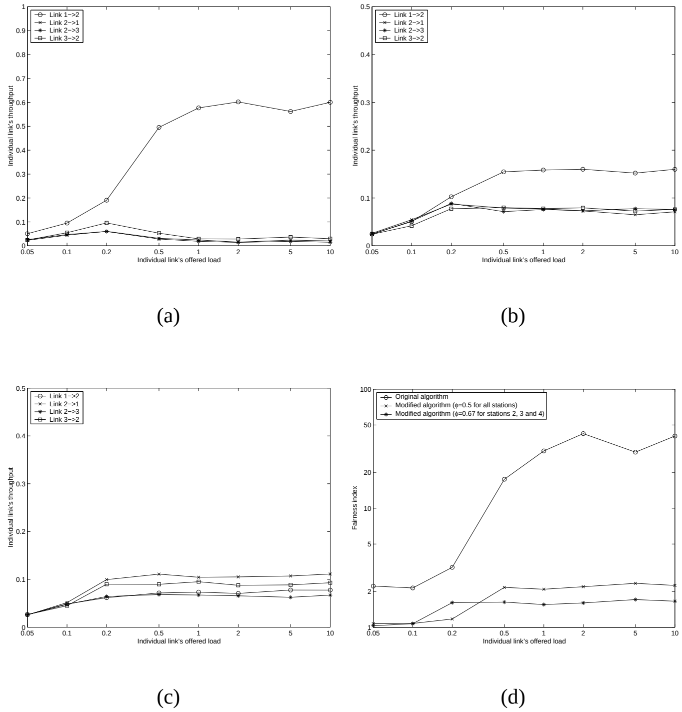
(b)

(c)

Fig. 3. (a) Station throughput (original algorithm), (b) station throughput (modified algorithm), (c) fairness index versus station offered load for the 5-station scenario – variable packet length.

still receive station 3's transmission successfully and reply to station 3 thereafter. According to BEB algorithm in DFWMAC which always favors the last succeeding station, station 3 usually enjoys a much smaller contention window, and thus has statistically shorter backoff timer than station 1. When the load is high, station 3 will *capture* the channel eventually. In our backoff algorithm, if station 3 overhears a few packets transmitted from station 2 (in this case, either CTS or ACK packet), its estimation will show that it has obtained more bandwidth share than what it should and will increase its contention window size[3] accordingly. With the ever increase of station 3's contention window, station 1 will get more chances to transmit packets to station 2. In the end, station 1's throughput can be balanced with station 3's throughput, so this algorithm can achieve far better fairness than BEB.

In the 5-station scenario, we investigate two cases. In the first case, each station generates Poisson traffic with the same mean rate. For station 2, 3 and 4, each has two active links to its neighbors. For each packet that is generated, these stations will randomly choose a neighbor as destination. For this case, we consider per-station fairness only and aggregate the two links' throughput as the corresponding station's throughput. The results are shown in Fig. 3. Due to symmetry, we show results for station 1, 2 and 3 only. In this case, edge stations (1 and 5) face less congestion and their packets get through "easier" than



(a)

(b)

(c)

(d)

Fig. 4. (a) Link throughput (original algorithm), (b) link throughput (modified algorithm, $\phi = 0.5$ for all), (c) link throughput (modified algorithm, $\phi = 0.67$ for station 2, 3 and 4), (d) fairness index versus station offered load for the 5-station scenario – variable packet length.

other stations' packets. As the BEB algorithm always favors the last succeeding station, the edge stations will get much higher throughput than other stations. Our algorithm works much better to achieve fairness because station 1 and 5 will yield the channel to other stations when they estimate that they have obtained extra share than what they should get.

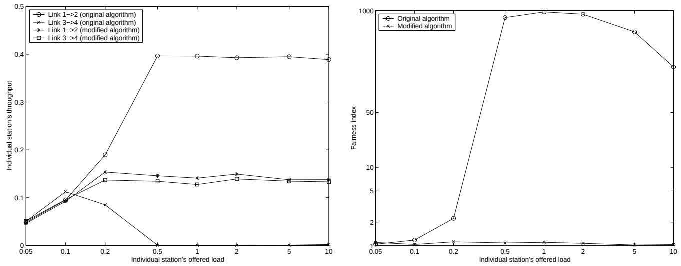
In the second case, each station generates Poisson traffic for each link with the same mean rate. Therefore, stations 2, 3 and 4 may require equal fair share for each of their links. We experiment with two situations. One is that station 2, 3 and 4 still set $\phi = 0.5$, the other is that these stations set $\phi = 0.67$ which indicates that they require two times the share of other stations as they have two active links. The results are shown in Fig. 4. It shows that even if station 2, 3 and 4 do not increase their fair share ϕ, the modified algorithm can still achieve much better fairness than DFWMAC. When they do increase ϕ, the fairness can be further improved.

In the 6-station scenario, each station generates Poisson traffic with the same mean rate and results are shown in Fig. 5. Due to symmetry, we show the results for station 1 and 3 only. It shows that station 3 is virtually denied access to the channel when load to channel is high and has nearly zero throughput when BEB is used. Due to the effective estimation of our algorithm, it can achieve far better fairness than BEB.

To show the effect of C on the fair index and throughput, due to limited space, we reproduce the results only for the 6 stations scenario in Fig 6. Results for other scenarios can be obtained from the authors upon request. As a general rule, we have observed that smaller values of Cs are usually favored as a station

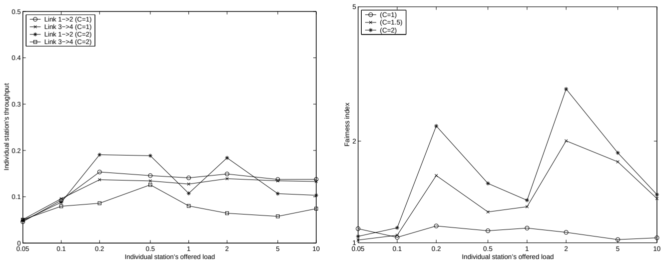
---

[3] Here station 3 does not even need not know the fact that station 1 has packets for station 2.



(a) (b)

Fig. 5. (a) Station throughput, (b) fairness index versus station's offered load for the 6-station scenario – variable packet length.



(a) Throughput (b) Fair index

Fig. 6. C's influence on 6-station scenario

can act more adaptively to changes in its estimated fair index and it is unlikely that a station will be stuck in a small contention window which can have negative effects on throughput. Generally speaking, C can be chosen very close to 1 or a little above it to achieve optimum results. Values close to 2 have shown instability in the case of 5 and 6 stations scenarios.

V. CONCLUSION

In this paper, we have proposed a new estimation based backoff algorithm for IEEE 802.11 DFWMAC protocol. The new backoff algorithm can support variable packet length as well as both basic access methods and RTS CTS access method. Simulation results have shown that fairness problem can be very severe in wireless ad-hoc networks when packet length is variable and our new algorithm can achieve far better fairness at the expense of some throughput. As the new algorithm does not require any extra information to be carried in packets and all stations run the algorithm distributedly, it is very simple and can be easily overlaid on the existing DFWMAC protocol.

REFERENCES

[1] "Mobile Ad-hoc Networks (manet) Charter." `http://www.ietf.org/html.charters/manet-charter.html`.

[2] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II - the Hidden Terminal Problem in Carrier Sense Multiple-access Modes and the Busy-tone Solution," *IEEE Trans. on Comm.*, vol. 23, no. 12, pp. 1417–33, 1975.

[3] P. Karn, "MACA - a New Channel Access Method for Packet Radio," in *ARRL/CRRL Amateur Radio 9th Comp. Networking Conf.*, pp. 134–40, ARRL, 1990.

[4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," in *Proc. of ACM SIGCOMM '94*, 1994.

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

[6] J. Garcia-Luna-Aceves and C. L. Fullmer, "Performance of Floor Acquisition Multiple Access in Ad-Hoc Networks," in *Proc. of 3rd IEEE Symposium on Computers and Communications*, 1998.

[7] K. Tang and M. Gerla, "Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks," in *Proceedings of IEEE MMT'99, Venice, Italy*, Oct. 1999.

[8] B. Bensaou, Y. Wang, and C. C. Ko, "Fair Medium Access in 802.11 Based Wireless Ad-Hoc Networks," in *IEEE/ACM MobiHoc 2000*, Aug. 2000.

[9] T. Ozugur, M. Naghshineh, P. Kermani, and J. A. Copeland, "Fair Media Access for Wireless LANs," in *Proc. of IEEE GLOBECOM '99*, 1999.

[10] N. H. Vaidya and P. Bahl, "Fair Scheduling in Broadcast Environments," Tech. Rep. MSR-TR-99-61, Microsoft Research, Dec. 1999.

[11] 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 Trans. on Networking*, vol. 1, pp. 344–357, 1993.

[12] X. L. Huang and B. Bensaou, "On Max-min fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation." To appear in ACM MobiHOC 2001.