CONTI: Constant-Time Contention Resolution for WLAN Access

Zakhia G. Abichar and J. Morris Chang

Dept. of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011, USA
{abicharz, morris}@iastate.edu

Abstract. Designing an efficient and fair access control protocol is a challenging task in the field of wireless networks. Often in the known schemes, one of the important performance metrics is enhanced at the expense of another. In this paper, we present a distributed access scheme that is based on the binary countdown mechanism. The proposed scheme has two main features: 1) the ability to resolve the contention in a constant number of time slots, hence the name constant-time (CONTI) and 2) a very low collision rate even at large network sizes. The simulation results show that CONTI outperforms the IEEE 802.11 DCF scheme in all the essential performance metrics: CONTI achieves a higher throughput by up to 55%, reduces the collision rate by up to 84%, renders the delay less variant and exhibits high fairness.

1 Introduction

The area of wireless access protocols has been an active research arena in the last decade. The main goal of shared medium access protocols is to achieve high efficiency and fairness under various network configurations. The wireless protocol standard that is dominantly in use today is the IEEE 802.11. The standard describes two modes of operation: the distributed coordination function (DCF) and the point coordination function (PCF). DCF is a distributed scheme that is designed to support best-effort traffic. On the other hand, PCF is a centralized scheme where stations are polled to ensure their service requirements are met.

In the IEEE 802.11e draft [1], which is proposed for quality-of-service (QoS) support, the DCF is enhanced through a scheme where each station runs up to four instances of DCF; this scheme is known as enhanced distributed channel access (EDCA). In addition, the PCF scheme is enhanced through a scheme where the access point is given increased priority to allow a better service for QoS traffic; this scheme is known as the hybrid coordination function (HCF). All of the aforementioned schemes are built on top of DCF, which increases the importance of the basic scheme.

Performance evaluation has shown that DCF performs reasonably well for transmitting best-effort packets in small-size networks. Its medium utilization has been shown to approach 68% in [2]. However, DCF leads to significant throughput degradation for networks of large size [2]-[3]. The main reason for the throughput drop under DCF is contributed to the high collision rate which reaches 40% (presented in section 6). Making use of this observation, we develop a scheme that is based on an efficient collision resolution mechanism called the binary countdown mechanism [4]. The main goal of the scheme is to allow for a low collision rate even when the network size grows.

Before a channel access occurs, CONTI eliminates a portion of the contending stations after the elapse of one contention slot. A powerful elimination mechanism is presented which is capable of reducing the number of stations from 100 stations to 5 stations in just one contention time slot. This elimination is achieved based on a randomly chosen value of a local Boolean variable. The value of the Boolean variable is then refreshed and the process is repeated several times until one station remains in the set of contending stations. As a result, the number of time slots that precedes a channel access is of logarithmic complexity.

According to CONTI, the collision rate is directly dependent on the number of contention slots: the longer the contention period, the lower the collision rate. Thus, a low collision rate can be achieved independently of the number of contending stations. Moreover, the resulting scheme exhibits a desired behavior: for a wide range of scenarios, CONTI can be efficiently run with the same number of contention slots. This would make the channel access time constant while keeping the collision rate minimal. Finally, the simulation studies show that CONTI outperforms DCF in all the essential performance metrics: throughput, delay, fairness and rate of collisions.

The rest of this paper is organized as follows. Section 2 discusses related work and section 3 briefly describes the operation of the IEEE 802.11 DCF. The proposed scheme is presented in section 4 and a mathematical analysis to determine the parameters of CONTI is presented in section 5. The simulation results are presented in section 6 and, finally, the paper is concluded by summarizing remarks in section 7.

2 Related Work

A brief history on the milestones in wireless networking protocols is included in the complete form of this paper, publicly available as a technical report [5]. This part has been omitted from this paper due to space limitation. We limit the discussion in this section to schemes that are closely related to CONTI.

The scheme we propose is based on a mechanism that is efficient in collision resolution called the binary countdown mechanism. In this section we give a description of the binary countdown mechanism and clarify the contribution of our work. In addition, we elucidate the difference between the binary countdown mechanism and a class of access protocols known as binary tree protocols. The idea behind the binary tree protocols is close to that of the binary countdown mechanism. However, there are significant hurdles in applying binary tree protocols to wireless networks [6]. On the other hand, the binary countdown mechanism can be easily applied to wireless networks.

The binary countdown mechanism has been described in [4] for wired networks access. However, this mechanism had a major problem in the form it was presented which made limited its practical use. To access the channel, stations run through a contention period whose length (in time slots) is equal to the number of bits representing the stations' addresses. Each time slot corresponds to a bit position, with the earliest time slot corresponding to the most significant bit. During each time slot, a station that has a value of one in the corresponding address bit jams the medium. Stations having a value of zero for that bit sense the medium. If the medium has been jammed, the listening stations retire from the current contention. Through this procedure, the station with the highest address is guaranteed to win the contention.

The main limitation in the binary countdown mechanism is letting the station with the highest address win. On the other hand, the main advantage of the binary countdown mechanism is collision-free communication since addresses are unique identifiers. In this paper, we re-discover the mechanism of binary countdown and consider the key parameters that allow to obtain the best of this idea. Our main contribution lies in addressing the two critical parameters that control the operation of the binary countdown mechanism. Those parameters have not been mathematically analyzed before:

- The binary countdown session should be run for a certain number of time slots. The number of time slots should be kept minimal so as not to waste time. On the other hand, the contention time should be high enough to allow for an efficient contention resolution procedure. We develop the necessary mathematical analysis in section 5 to address this issue.
- For each time slot a station needs to decide whether to jam or sense the medium. Making this decision based on the address of the stations is not at an optimal solution. We present a mechanism that chooses those bits in such a way to quickly reduce the number of contending stations. The resultant mechanism is remarkably powerful and allows reducing the number of stations from 100 to 5 in just one time slots.

In the recent literature, the binary countdown mechanism has been employed to underlie access schemes for wireless networks [7]-[8]. However, the mechanism was used in a sub-optimal manner. The probability of selecting bit values was chosen purely randomly [8]. In addition, the number of contention slots was not kept at a minimum and many components have been added to it to maintain fairness.

A class of algorithms that is close to the binary countdown mechanism has been proposed in the 1970s by Tsybakov [9], Capetanakis [10] and others. Those algorithms are known as binary tree protocols. In the binary tree schemes, stations are allowed to collide and then colliding stations are divided into groups. Then, the groups are processed in a certain order and collisions within these groups are resolved by recursively dividing the concerned groups until a group contains one stations.

The shared aspect between binary tree protocols and the binary countdown mechanism is that, in both cases, the stations are divided into groups. However, binary tree protocols allow collisions to occur since, in wired networks, collisions can be detected and an ongoing transmission is halted, if need be. In wireless networks, it is not possible to detect collisions while transmitting.

A scheme has been proposed [11] to apply binary tree protocols to wireless networks. The scheme uses a separate feedback channel that informs the stations whether the transmission has collided or not. The feedback is provided by an access point. This solution, however, has few limitations: 1) there is a need for a separate channel, 2) even though the stations are informed of the results of a collision, the feedback is received after the whole frame is transmitted and 3) there is a need for an access point which renders the scheme inapplicable to ad-hoc networks. On the other hand, the proposed scheme CONTI is free of those limitations.

3 Operation of the IEEE 802.11 DCF

The contention-based access mechanism that is adopted in the 802.11 standard is the DCF. A successful frame transmission cycle under DCF consists of the transmission of a data frame and the reception of the corresponding ACK frame by the source station (Fig.1). In addition, a transmission cycle may include one or more collisions.

Before transmitting a data frame, a station senses the medium. If the medium is idle, the station initiates the transmission. Otherwise, the station proceeds to follow a two-step procedure. First, the station waits until the medium becomes idle and then defers for an interframe space (IFS) that is defined for each class of frames (DIFS for data frames). If the medium remains idle throughout the IFS, the station proceeds to the second step which consists of setting a backoff timer to a value that is uniformly chosen from the contention window (CW) interval. The purpose of the second step is to reduce the chance of a collision between stations that are transmitting data frames. After that, the station reduces its backoff timer by one unit (slot time) following the elapse of an idle time slot. If the medium ceases to be idle during any of the time slots, the station pauses its backoff timer and resumes it after detecting the medium to be idle for another IFS time frame.

In the event where the backoff timer reaches zero, the station transmits its packet. Should a collision occur, the involved stations defer and each of those stations doubles its CW if it were less than the defined maximum CW size. In the case of a successful packet transmission, the destination station defers for a short IFS (SIFS) and then sends back an ACK packet. Moreover, the transmitting station resets its contention window to the minimum defined value (CW_{min}). More elaborate details on the operation of the DCF scheme can be found in the IEEE 802.11 standard document [12].

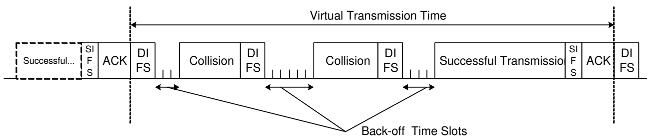

Fig. 1. DCF transmission cycle

4 Proposed Scheme

To provide contention access to the medium, CONTI runs a certain number of contention slots to resolve the contention before a transmission can be initiated. This period will be referred to as the contention resolution period (CRP) henceforth (Fig.2). During each contention slot, a portion of the contending stations is eliminated. When a station is eliminated, it is meant that it will cease contention to the medium during the current CRP. Thus, it can contend again after the occurrence of one transmission trial.

The elimination of stations is achieved through the use of a local Boolean variable called the $try-bit$. Each station randomly chooses a value for its try-bit and either jams ($try - bit = 1$) or senses the medium accordingly. A station retires only if its try-bit is equal to zero and the medium was jammed. The stations with try-bit equal to one do not retire at this contention slot. After the elapse of one time slot, the stations that were not eliminated refresh their try-bit variables and repeat the same process. This process is repeated until one station remains in the set of contending stations. Therefore, this station will initiate a transmission.

This procedure proves to be capable of quickly reducing the number of contending stations if the relevant parameters are carefully chosen. The two relevant parameters are: 1) the number of time slots during which the contention resolution runs and 2) the probability of jamming or sensing the medium during a given time slot. Those two issues are addressed in section 5.

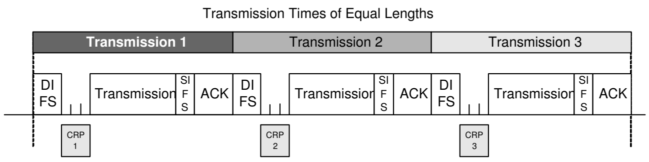

Fig. 2. Operation of CONTI

In general, the length of the CRP should increase with an increase in the number of operating stations. However, we show in the mathematical analysis of section 5 that the collision resolution mechanism is flexible enough to let CONTI run with a constant number of contention slots for a wide range of scenarios.

We finally mention the following notes: during a contention slot, if all the stations choose a value of one for the try-bit, then all of them would jam the medium and thus move to the subsequent contention slot. On the other hand, if all the stations choose a value of zero for the try-bit, the medium remains idle and, again, all the stations proceed through the CRP.

It is noteworthy to mention that it is not possible for all the stations to retire from the contention. This is because a station retires, only if there is another one who jammed the medium. Thus, the station that has jammed the medium does not retire.

The proposed scheme does not totally eliminate the chance of the occurrence of a collision. In the event where two (or more) stations choose the same value for their try-bits in the last contention time slot, those stations will collide.

5 Mathematical Analysis

We present in this section a mathematical analysis that allows for choosing the two parameters for CONTI: the number of contention time slots and the probability for assigning a value of one for the try-bit during a given time slot.

5.1 Choosing a Value for the *try-bit* with Probability p

In the process of contention, a station chooses a value for its try-bit variable during each time slot. The chosen value is equal to one with probability p. This value determines whether the station will jam or sense the medium.

We highlight the following two factors. *Factor A:* If the value of p is high, most of the stations would choose a try-bit of value one and thus move to the next contention slot. In this case, the collision will be resolved in a slow manner. On the other hand *(Factor B)*, an extremely low value of p is not desirable. In that case, all stations may get a value of zero and thus no stations would be eliminated. This is especially true for a low number of contending stations, which occurs during the last couple of the contention slots.

At the early time slots of the contention, the number of competing stations is large and is reduced as the contention proceeds. Thus, the value of p is made small at the early contention slots *(due to Factor A)* and then elevated as the contention proceeds *(due to Factor B).* To avoid elevating the value of p in the last time slot more than needed, we make use of the *Lemma* that is presented later in this section, where the optimal value of p is found for the last contention slot.

After the elapse of a contention slot, the probability that i stations, out of n contending stations move to the subsequent contention slot is given by the following equation:

$$pr(i) = \begin{cases} \binom{n}{i}.p^i.(1-p)^{n-i} & 1 \leq i \leq n-1, \\ p^n + (1-p)^n & i = n. \end{cases} \tag{1}$$

It follows that the expected number of stations, E, that will move to the subsequent contention slot is given by the following equation:

$$E = \left(\sum_{i=1}^{n} i.\binom{n}{i}.p^i.(1-p)^{n-i} \right) + n.(1-p)^n. \tag{2}$$

Solving this equation numerically, we can determine the optimal values of p that allow for the fastest reduction in the number of stations between two consecutive contention slots.

For a given number of stations, there is an interval for the values of p that allows for the steepest reduction in the number of stations per time slot. Fig.3 plots those intervals as the number of stations varies up to 100. The digit that is placed over the interval lines represents the minimum number of stations that could remain in contention after the elapse of one slot time. For example, starting with 20 stations in a given contention slot, a value of p in the interval [0.12 – 0.18] reduces the number of stations to an expected value of 3, the steepest possible reduction.

Fig.3 also shows a desirable property of this mechanism. We can see that the intervals for several network sizes share many points in common. The dotted line has been drawn to pass through those intervals in a smooth way. It could be noticed that the dotted line is not far from being a horizontal line. This fact allows us to choose a value of p that is close to the optimal value even if the network size is not known. Thus, it makes it feasible to run CONTI under the same tuning for a wide range of scenarios.

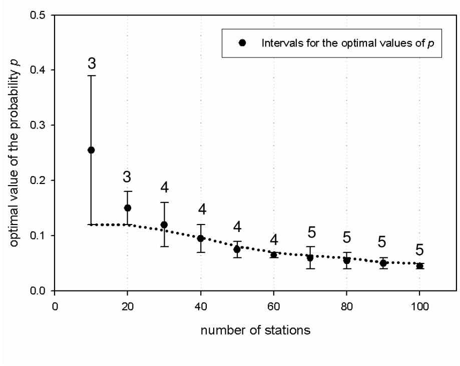

Fig. 3. Intervals of p for the fastest reduction

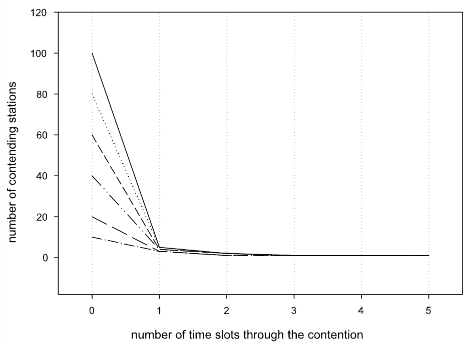

Fig. 4. Fastest contention resolution

Fig.4 shows how fast the number of competing stations can be reduced when the optimal values of p (based on Fig.3) are employed, thus using a minimum number of contention slots. From the data underlying this figure, it takes two contention slots before the number of expected stations is reduced to one when starting from 20 stations or less. As for the other cases, 20 to 100 stations, it takes three contention slots. In practice, it is beneficial to run the contention for one or two additional slots if we need to ensure a low collision rate.

At the last contention slot, a successful transmission occurs in one of the following two cases. The first case: only one station has remained from the previous contention slot. This station will successfully transmit independently of the chosen value for its try-bit. The second case: there is more than one remaining station and the try-bits are chosen in such a way that only one station has a value of one for the try-bit. For the latter case, the optimal value of p is given by *Lemma 1.*

Lemma 1: Given n remaining stations at the last contention slot, the optimal solution demands that each station chooses a value of one for its try-bit with probability $p = 1/n$.

Proof: At the last stage of the contention, a successful transmission occurs if only one station chooses a value of one for its try-bit. It follows that the probability of a successful transmission is given by:

$$p_s = n.p.(1-p)^{n-1}. \tag{3}$$

The value of p that maximizes p_s is obtained by solving the following equation (4) which leads to $p = 1/n$.

$$\frac{dp_s}{dp} = n.(1-p)^{n-1} - n.p.(n-1).(1-p)^{n-2} = 0.\blacksquare \tag{4}$$

In practice, the value of p will be set to $1/2$ in the last contention slot. This is because the number of expected stations that remain in the last contention slot is equal to two, based on the information from Fig.4.

6 Performance Evaluation

This section presents the simulation studies that were carried to evaluate CONTI and compare its performance to the IEEE 802.11 DCF. We developed a packet-level simulator to conduct the simulation studies. The IEEE 802.11 standard was run using the DSSS specifications. The parameters used for the simulation are summarized in Table 1. It was assumed that best-effort data packets are available at stations at all times. Separate simulation runs were carried for various packet sizes ranging from 50 bytes to 1250 bytes at 2 Mbps. In addition, simulation studies were conducted at the rate of 11 Mbps for packet sizes ranging from 250 bytes to 2346 bytes. The results for 11 Mbps were omitted due to space limitation and due to the fact that they exhibit similar trends to those shown at 2 Mbps. Those results are included in the technical report version of this paper [5].

For the presented simulation results, CONTI was run for six time slots. This number of the time slots has resulted in the best values of the throughput when jointly considering all the presented scenarios. The values of p that were used during each of the six time slots are shown in Table 1.

Table 1. Simulation Parameters

Parameters for DCF and CONTI	Parameters for CONTI
DIFS duration: $50\mu s$	#contention slots: $k = 6$
SIFS duration: $10\mu s$	$p_i : i = [1 : 6]$
Slot time: $20\mu s$	$[0.07, 0.2, 0.25$
Channel rate: $2Mbps$	$0.33, 0.4, 0.5]$

Figures 5, 6 and 7 show a throughput comparison between CONTI and DCF when the channel rate is 2 Mbps and the network size is 10, 50 and 100 stations, respectively. The measured normalized throughput is defined as the data rate (including packet headers) divided by the channel rate. In addition, the ideal throughput of CONTI is represented by the dotted line. The ideal throughput can be attained if the number of competing stations is known to all the stations. The packet sizes for those simulation runs vary from 50 bytes to 1250 bytes.

For a network size of 10 stations, CONTI attains a throughput of 92.4% while DCF reaches a throughput value of 82.2%. However, as the the network size grows larger, the disparity in performance between CONTI and the DCF is widened. For 50 stations, CONTI reaches a throughput of 91.5% while the standard's throughput attains 66.5%. Finally, for the case of 100 operating stations, CONTI achieve a throughput of 90.4% while the standard's throughput is 58.5%.

In the case where the network size is 10 nodes (Fig.5), DCF allows for a better performance than CONTI for packet sizes that are less than 75 bytes. The reason for this trend is twofold. Firstly, at a low network size, the collision rate of DCF is low even though it is still four time higher than that of CONTI (measurements of collision rates are presented in Fig.8). Thus, the low collision rate of CONTI is not much advantageous over DCF. Secondly, the average access time of DCF for a 10-node network is less than six time slots (as measured: about 3 slots). Then, the combination of those two factors allows DCF a better performance for the considered specific scenario. However, as the packet size grows larger, CONTI outperforms DCF. This is because the cost of a collision increases for large packet sizes. We note that the chances of transmitting a packet of size smaller than 75 bytes at 2 Mbps is very small. Thus CONTI retains the advantage in the majority of cases.

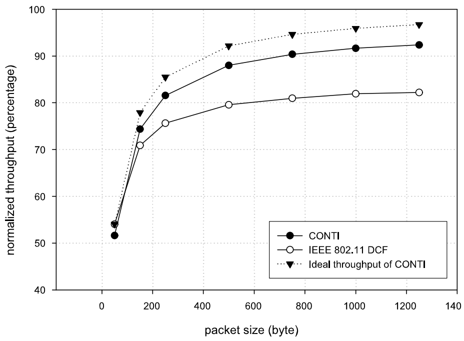

Fig. 5. Throughput for 10 stations at 2 Mbps

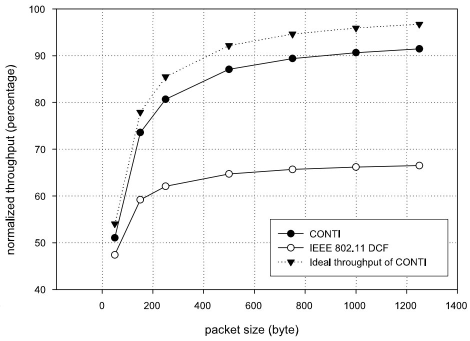

Fig. 6. Throughput for 50 stations at 2 Mbps

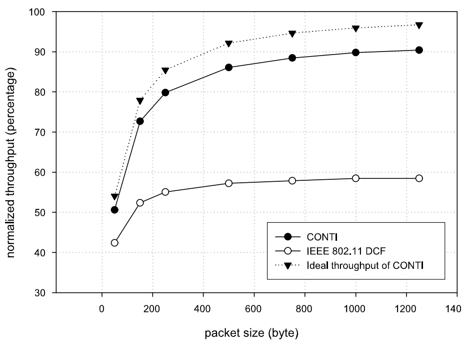

Fig. 7. Throughput for 100 stations at 2 Mbps

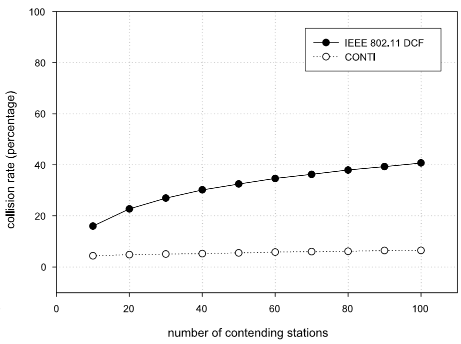

Fig. 8. Collision rates

As we mentioned in the introductory section, the main motivation of CONTI was to allow for a low collision rate. The measurement of this factor is presented in Fig.8.

This figure shows the collision rates for CONTI and DCF as the number of contending stations varies from 10 to 100 stations. The packet sizes in this case are set to 1000 bytes. It could be noticed that there is a great disparity in the collision rates between the two schemes. For the shown cases, the collision rate of CONTI varies from 4.37% to 6.37% while that of DCF varies from 16.00% to 40.75%. This observation is significant in giving insight about why CONTI outperforms DCF: the high collision rate of DCF impedes other essential performance metrics such as the throughput, delay and fairness.

Figures 9 and 10 show a comparison of the delay distribution when the network consists of 10 and 100 stations, respectively. The delay metric that is presented in the following plots is measured from the moment the packet reaches the head-of-line in the queue to the instant it is successfully received at the other party. For the following simulation runs, the packet sizes were set to 1000 bytes.

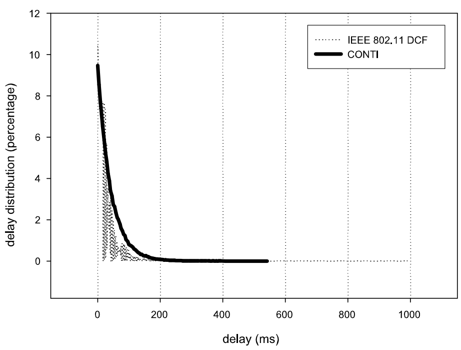

Fig. 9. Delay distribution for 10 stations

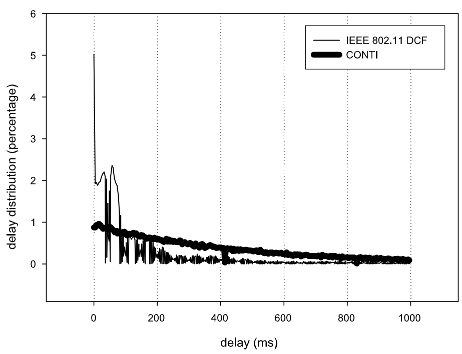

Fig. 10. Delay distribution for 100 stations

In Fig.9, the delay distribution of CONTI is represented by the thick line. The worst case delay for CONTI in this case is 542 ms while that of DCF is 5.07 seconds. It could also be noticed that the delays of the packets that correspond to DCF are much fluctuating. This fluctuation is attributed to the high collision rate of DCF. Since the instant in which a collision can occur is highly random, this renders the delay that is experienced by a packet hard to anticipate. However, for the case of CONTI the delay distribution is much smoother.

For the case of 100 operating stations, the worst case delay for CONTI is 5.2 seconds whereas the corresponding one for DCF is 25.93 seconds. However, from the data underlying Fig.10 (also shown in the figure), it could be noted that about 5.02% of the packets under DCF are transmitted with an infinitesimal delay while only 0.87% of the packets under CONTI are transmitted with that minimal delay, which is close to zero. This trend is attributed to the following reason: the average access time of DCF is smaller than that of CONTI. Thus, in a given time interval, more transmission trials are initiated under DCF. This gives the chance of having more occurrences of the infinitesimal delay case. However, due to the high collision rate of DCF, a fairly large delay can occur at any time.

As an overall result, DCF allows for more frequencies of the infinitesimal delay at the expense of many fluctuations in its delay distribution. On the other hand, CONTI favors regularity in the expected delay at the expense that its minimal delay may be higher than that of DCF. This observation renders CONTI a more suitable scheme to be extended for support of real-time traffic where delay anticipation is a precious asset. We also note that the worst-case delay of DCF is much larger than the corresponding one of CONTI.

Figures 11 and 12 present the measurement of the fairness metric for DCF and CONTI. For those simulation runs, the packet sizes were set to 1000 bytes. The simulation was run until 100,000 transmission trials have occurred.

The plots in the figures show the percentage amount of the fair share that each of the contending stations has attained. The fair share is obtained by dividing the number of successfully transmitted packets by the number of operating stations.

For the case of 10 operating stations, the fair share values under CONTI range from 98.34% to 101.31% whereas the counterpart values that correspond to DCF range from 95.00% to 105.83%. At first sight this might seem a bit unexpected since the DCF scheme is well credited in regard to the fairness metric. Taking a closer look at the binary exponential backoff algorithm, it could be realized that a station doubles its contention window size every time it undergoes a collision, thus imposing a penalty on colliding stations. In the short following time range, the colliding stations are disadvantaged with respect to other stations. On another count, all stations are equally likely to collide in the long-run. We then deduce that the time range that takes DCF to be fair is greater than the corresponding range of CONTI. This could be interpreted by the fact that CONTI is a memoryless protocol.

The same trend that was described in the previous paragraph is also exhibited for the case of 50 contending stations. In this scenario, the fair share values under CONTI range from 96.88% to 105.67% whereas the corresponding values to DCF range from 77.81% to 116.67%. Finally, we mention the following observation: the fair share values at 50 stations exhibit a wider deviation than those measured at 10 stations. This is due to two reasons: 1) for an undiscriminating scheme, the longer the simulation time is, the closer the share values are to 100% and 2) the number of transmission trials has remained the same for both of the simulation scenarios.

7 Conclusion

In this paper, we propose a contention-based MAC scheme that is based on the binary countdown mechanism. In CONTI, an efficient mechanism is employed to allow for a fast contention resolution and a low collision rate. We also present a mathematical analysis that allows to choosing proper values for the parameters of CONTI. Finally, the presented simulation studies show that CONTI outperforms DCF in all of the essential performance metrics: 1) CONTI allows for a higher throughput for the vast majority of the scenarios. 2) A given value of the fairness metric could be achieved in a shorter time range over CONTI as compared to DCF. 3) The delay distribution under CONTI is

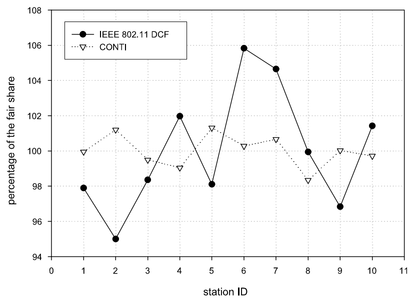

Fig. 11. Fairness measure for 10 stations

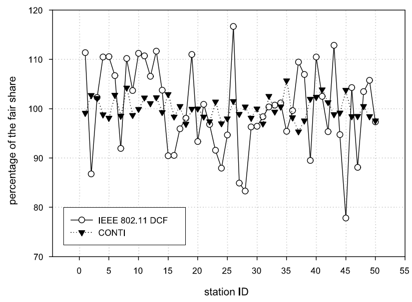

Fig. 12. Fairness measure for 50 stations

much smoother than that of DCF and the worst-case delay of CONTI is much smaller. 4) The collision rate of CONTI is, by a large margin, smaller than that of DCF. In summary, CONTI proves to be an efficient wireless MAC scheme that is capable of keeping a balance between all the essential performance metrics.

References

1. IEEE 802.11 WG, 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), IEEE 802.11e/Draft 8.0, February 2004.
2. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications, Vol.18, No.3, pp.353-547, March 2000.
3. Golestani, S.J: A Self-Clocked Fair Queuing for Broadband Applications. IEEE INFOCOM, 1994, pp.636-646.
4. Tanenbaum, A.S: Computer Networks, 4th edn. Prentice-Hall, Redwood City, CA (2003)
5. Abichar, Z., Chang, M.: CCR: A novel MAC scheme with constant-time contention resolution for WLAN. Iowa State University Technical Report. TR-2004-07-0. August 2004.
6. Bertsekas, D., Gallager, R.: Data Networks, 2nd edn. Prentice-Hall, Englewood Cliffs, NJ (1992)
7. Yeh, C.H.: Distributed Collision-Free/Collision-Controlled MAC Protocols for Mobile Ad Hoc Networks with Hidden Terminals. International Conference on Information Networking(ICOIN), August 2004.
8. Wu H., Utgikar A. Tzeng, N.F.: SYN-MAC: A Distributed Medium Access Control Proto-col for Synchronized Wireless Networks, ACM Mobile Networks, to appear.
9. Tsybakov, B.S., Beloyarov, A.N.: Random multiple access in binary feedback channel. Problemy Peredachi Informatsii. Vol26 (4). October-December (1990) 83-98.
10. Capetanakis, J.I.: Tree algorithms for packet broadcasting channel. IEEE Transactions on Information Theory. Vol25. September (1979) 505-515.
11. Garcs, R., Garcia-Luna-Aceves, J.J.: Collision avoidance and resolution multiple access for multichannel wireless networks. IEEE INFOCOM, 2000.
12. IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11, 1997.