Comparison of Distributed Fair QoS Mechanisms in Wireless LANs

Wasan Pattara-Atikom†, Prashant Krishnamurthy †, and Sujata Banerjee* †
† Telecommunications Program, University of Pittsburgh, Pittsburgh, PA 15260
*Mobile and Media Systems Laboratory, Hewlett-Packard Laboratories, Palo Alto, CA 94304
{wapst7,prashk,sujata}@pitt.edu

***Abstract*— It is well known that the distributed coordination function (DCF) of the IEEE 802.11 MAC protocol is not suitable for supporting multimedia and QoS-sensitive applications because of its inherent lack of QoS support and fairness. Recently, several distributed QoS mechanisms have been proposed which translate user QoS requirements into typically a single parameter of the DCF protocol. In this paper, we compare the pros and cons of the major distributed QoS mechanisms, and propose a new mechanism provides superior performance and supports two different QoS models. The proposed mechanism is based on deficit round robin scheduling and translates the user throughput requirements into the 802.11 MAC InterFrame Space and Backoff Interval parameters. We show via simulations that the proposed mechanism provides low variability of throughput and delay and has the advantage of low complexity.**

Symposium: Wireless Communications

I. INTRODUCTION

Wireless Local Area Networks (WLANs) have become the networks of choice with an increasingly high demand for Quality of Service (QoS)-sensitive applications. These applications are time sensitive and require specific throughput and delay bounds. The current medium access protocol in IEEE 802.11 WLANs is suitable for QoS-insensitive and bursty data traffic only, and does not have any QoS support. Several QoS-enabling mechanisms have been proposed as extensions to DCF over the last few years. These proposals provide service differentiation among different traffic classes, but they also suffer from high throughput variability and lack of fairness, especially for traffic from a low data rate class.

Recently, fair queuing mechanisms have been investigated to overcome these problems by partitioning the bandwidth fairly between different mobile stations (MSs). In this work, we proposed a technique called *distributed deficit round robin* (DDRR) that is similar in approach to these fair queuing mechanisms, but is simple to implement with excellent performance in terms of reducing throughput and delay variability. In Section II, we describe related work in this area followed by a brief description of the IEEE 802.11 MAC and its problems in Section III. We describe DDRR in Section IV, the simulation set up and numerical results in Section V and conclude the paper in Section VI.

II. RELATED WORK

Recent research activities have investigated different ways to support QoS, by providing different treatment for different traffic types in IEEE 802.11 WLANs. The primary focus in these efforts is to provide service differentiation between low priority traffic and high priority traffic. The various proposals provide performance differentiation, typically by tuning one of three WLAN parameters: 1) length of Inter-Frame Space (IFS) [1], [2], [3], 2) length of Contention Window (CW) [4], [5], [6], and 3) length of Backoff Interval (BI) [7], [6], [8]. In each case, a smaller value of the parameter (IFS, CW or BI) is assigned to traffic with higher priority enabling them to access the medium before traffic with a lower priority. A comprehensive review of QoS support in IEEE 802.11 WLANs can be found in [9].

In the DCF mode of operation, MSs operate independently making it harder to employ any scheduling mechanism. The principle of fair queuing can be used to regulate the wait time of traffic according to its priority so that traffic in each class has an equal opportunity to be sent and the bandwidth is fairly apportioned between different traffic classes. Vaidya et al. [7] proposed a mechanism called Distributed Fair Scheduling (DFS) and Banchs et al. [6] proposed Distributed Weighted Fair Queuing (DWFQ). Both mechanisms are based on fair queue scheduling and provide fair access to shared bandwidth, in proportion to *flow weights*.

In DFS, a packet with the smallest ratio between its length and weight receives an opportunity to transmit first. The weight represents a value associated with the traffic class. A higher data rate traffic class has an associated higher weight. The main idea of this mechanism is to pick a backoff interval proportional to a finish tag. The finish tag is the ratio between the packet length and the weight of a frame given by: $B_i = \left\lfloor \left\lfloor Scaling\_Factor * \frac{L_i}{\phi_i} \right\rfloor * \rho \right\rfloor$, where B_i is the backoff interval, L_i is the packet length, ϕ_i is the weight, and ρ is a random variable uniformly distributed in the range of [0.9, 1.1].

In DWFQ, all flows of all MSs are constrained to have the same ratio (TW_i) between throughput (R_i) and a weight (W_i), i.e. $TWi = R_i/W_i$. By comparing its own TW_i to those of other MSs, a given MS can adjust its CW. The CW is decreased if its TW_i is smaller than those of other MSs and it is increased. The weights are used to differentiate between traffic classes and apportion the bandwidth between them. However, the randomness associated with using the CW increases the variability of throughput and delay. Moreover, this mechanism requires an additional field in the frame header in the MAC layer to exchange the values of TW_i among MSs.

Although our proposed scheme has some similarities with DFS and DWFQ, our mechanism is different in the following aspects: 1) the choice of the fair queuing mechanism, 2) translation of user requirements 3) the WLAN parameters used to differentiate service and 4) type of QoS support. First, the choice of the fair queuing mechanism will affect protocol complexity and computational cost. For example, the mechanism in [7] is based on Self-Clocked Fair Queuing (SCFQ) [10] which has

$O(\log(n))$ complexity where n is the number of flows. The proposed DDRR mechanism is based on the Deficit Round Robin (DRR) [11] which is of complexity $O(1)$. Second, the method used to translate users' requirements (such as throughput or delay) into the parameter of the fair queuing mechanisms, i.e. weight, can be complex and impractical. In our work, we have established a simple relation between the desired throughput of a user and the relevant DRR parameter. Third, our scheme allows using combinations of the IFS, BI, and CW parameters to differentiate between traffic classes at different levels. Instead of modifying only one parameter (the BI in [7] and the CW in [6]) we modify both the IFS and BI values since each parameter has its own differentiation ability and effectiveness that vary under different scenarios. Lastly, rather than supporting only QoS differentiation, our mechanism has the ability to provide both absolute throughput levels and relative throughput (or throughput differentiation) in the same system. This is attractive for future applications that are expected to consist of both types of QoS. Applications such as voice, video, and medical information require absolute QoS. Applications such as email, web, and file transfer require only QoS differentiation.

III. THE IEEE 802.11 MAC

The DCF mode of IEEE 802.11 is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). A MS first senses the medium to determine whether the channel is busy. If the channel is idle for a time equal to DCF Inter-frame Space (DIFS), the MS can transmit a packet. If the medium is busy, the MS 1) waits until the medium becomes idle, 2) waits for DIFS and 3) enters a deferral period to reduce the chance of collisions with other MSs that may have also initially sensed the medium as busy and proceeded likewise. In the deferral period, the MS selects a backoff interval - BI, that is uniformly distributed between a minimum period (CW_{min}) and a maximum period (CW_{max}). The difference between CW_{max} and CW_{min} is called the contention window (CW). The deferral period is divided into time slots, each a duration of $SlotTime$ seconds. The length of the backoff interval is calculated as a multiple of $SlotTime$ as shown in (1).

$$BI = Random(CW_{min}, CW_{max}) * SlotTime \tag{1}$$

As long as the medium is idle, a backoff timer counts down from the BI value every $SlotTime$. When the timer reaches zero and the medium is still idle, the MS can transmit. The backoff timer stops when a transmission is detected and continues to elapse when the channel becomes idle again. After receiving a frame, the receiving MS waits for a Short Inter-frame Space (SIFS) and responds with an acknowledgement (ACK) to confirm a successful transmission. The SIFS parameter is smaller than DIFS to allow acknowledgements to have the highest priority. The standard also defines a *point-coordination function IFS* (PIFS) that is in between the SIFS and DIFS values and is used for polling. A collision occurs when the backoff timer of two or more MSs reach zero at the same time. To reduce the probability of collision, the CW is doubled every time a collision repeatedly occurs until the maximum value of CW is reached, i.e. 1023 timeslots.

Currently, there is an on-going effort in the IEEE standard called 802.11e which intends to support QoS in WLAN by assigning different values of IFS, CW_{min}, and Persistence Factor (PF) to different traffic category. The higher priority traffic receives small IFS, CW_{min} and PF. Our proposed mechanism also used the same set of parameters, i.e. IFS and BI, to provide QoS in WLAN. However, 802.11e does not consider fairness or using fair queuing scheduling in provide QoS the same way previous studies [1], [2], [3], [5] the priority of traffic is statically combined with the right to access the medium leading to unfair occupation of the bandwidth.

IV. DISTRIBUTED DEFICIT ROUND ROBIN (DDRR) MECHANISM

In [12], we presented an initial version of the DDRR mechanism that translated user throughput requirements into the 802.11 IFS parameter. In this paper, we build on that work to design a generalized scheme (also called DDRR) that is an enhancement of the previously proposed scheme. The generalized DDRR is able to support two different QoS models - an absolute throughput model and a relative throughput model. The QoS guarantees are soft guarantees and are contingent upon the total offered load sharing the 802.11 channel capacity. In our ongoing work, we are considering the design of a distributed admission control strategy which will be able to indicate what throughput levels can indeed be given hard guarantees.

We first describe the concepts behind DDRR and then describe how the generalized DDRR works. Suppose the traffic at each MS is categorized into classes with different QoS requirements. A traffic class i at MS j (TC_i^j) with throughput requirement (λ_i^j) is allotted a *service quantum* of Q bits every T_i^j seconds. Traffic classes requiring a higher throughput receive the service quantum at a faster rate. The "Deficit Counter" of TC_i^j at any given time is denoted by (DC_i^j). The value of DC_i^j is increased continuously with time at a rate of Q bits every T_i seconds as shown in Eq. 2, and is decreased by the size of the frame whenever a frame is transmitted as shown in Eq. 3.

$$DC_i^j(t) = DC_i^j(t') + \frac{Q}{T_i} * (t - t') \tag{2}$$

$$DC_i^j(t) = DC_i^j(t) - Frame\_Size(t) \tag{3}$$

The simplicity of this scheme is that we can simply choose Q and T_i such that the quantum rate $Qr_i^j = \frac{Q}{T_i^j} = \lambda_i^j$, the required throughput (or expected load) for the traffic class. If DC_i^j is higher than the minimum DC_i^j required for transmission, a frame that belongs to TC_i^j can be transmitted by entering a transmission process. The transmission process could simply be DCF as described previously. Alternatively, it could be the mechanism described in [12][1] or that described for the generalized DDRR below. If DC_i^j is less than the minimum DC_i^j requirement, TC_i^j is no longer eligible to transmit until DC_i^j becomes higher than the minimum DC_i^j required.

Because of this, the basic DDRR followed by a transmission process can be used to regulate traffic class TC_i^j to provide fair

---

[1] In [11], no backoff is used. Instead, the DC_i^j value is translated into an IFS multiplied by a random number to prevent collisions

bandwidth allocation according to the quantum rate as long as the total quantum rate ($\sum Qr_i^j$) of all classes in all stations is less than the maximum possible throughput of the network (C). If $\sum Qr_i^j$ is higher than C, the basic DDRR will deteriorate to the basic transmission process if no other changes are made to the protocol for the following reasons. When $\sum Qr_i^j$ is higher than C, there will always be a left-over quantum accumulated in DC_i^j. As the time passes, the DC_i^j will always be higher than the minimum DC_i^j requirement and ultimately reach the maximum value. If the maximum value of DC_i^j is the same for all i classes and all j stations, every traffic class will be equally eligible to transmit data. In order to allow fair allocation, a transmission process needs to be in place that further produces fair allocation of bandwidth when the DC_i^j of all stations is filled up due to lack of channel capacity. That is the goal of generalized DDRR.

In generalized DDRR, we make use of the allocated quantum rate to traffic classes to regulate access to the medium in case the DC_i^j reaches its maximum value for all MSs i.e., we modify the transmission process. We achieve this by mapping the DC_i^j value and the quantum rate to the parameters in the DCF mechanism - the IFS value and the BI as follows: Let R be the raw data rate of the channel. Then we define the following quantities:

$$W_i^j = \frac{Qr_i^j}{R} \tag{4}$$

$$BI = \frac{BI_{802.11}}{W_i^j} \tag{5}$$

$$IFS_i^j(t) = DIFS - \alpha * \frac{DC_i^j(t)}{Q} + \frac{\delta}{W_i^j} \tag{6}$$

The quantity W_i^j (called the *weight*) is equal to the fraction of the channel bandwidth demanded by TC_i^j. For example, if $Qr_i^j = 200$ kbps and $R = 2$ Mbps, $W_i^j = 0.1$. The IFS value for traffic class TC_i^j comprises of three components - the DIFS from DCF, a negative value derived from DC_i^j and a positive value dependent on W_i^j. Let us consider two cases: (1) underload when $DC_i^j < DC_{max}$ (i.e., $\sum Qr_i^j < C$) and (2) overload when $DC_i^j = DC_{max}$ (i.e., $\sum Qr_i^j > C$). DC_{max} is the maximum number of DC_i^j allowed for TC_i^j to accumulate quantum and used to avoid unfairness due to excessive accumulation of DC_i^j.

As long as the DC_i^j value is less than the maximum, we let $W_i^j = 1$. In that case, the last component in (6) remains a constant. In such a case, we see that IFS_i^j is the difference between $DIFS$ and the ratio between the size of the quantum Q and the value of deficit counter DC_i^j. The parameter α is a scaling factor used to translate the ratio between Q and DC_i^j into an appropriate length of IFS_i^j that complies with IEEE 802.11. The value of DC_i^j is bounded between zero and $\frac{DIFS-PIFS}{\alpha} * Q$ so that the value of IFS_i^j is bounded between PIFS and DIFS. The longest possible IFS, corresponding to the smallest value of the deficit counter is equal to DIFS + δ. The smallest possible IFS value is PIFS + δ. The channel is allocated to traffic classes TC_i^j fairly as accumulating DC_i^j allows TC_i^j to access the medium earlier.

When all TC_i^j have reached maximum DC_i^j values, the weight W_i^j becomes important. In this case, the quantity $\alpha * \frac{DC_i^j(t)}{Q}$ is a constant for all TC_i^j. The larger the weight, the smaller will be the factor $\frac{\delta}{W_i^j}$ so that the class TC_i^j with largest throughput requirement has the smallest IFS. However, to ensure fairness, the backoff interval is also scaled as shown in (5).

The operation of the transmission process in this way enables us to define two types of QoS. As long as the channel resources are available, we keep the weight equal to 1 for all traffic classes. When channel resources are no longer available, the weight defaults to its definition in (4). This process can be implemented in a different way as well. For example, we could split a 2 Mbps channel into a 1 Mbps *absolute* part and a 1 Mbps *differential* part. An admission control process (that is under investigation) will allocate the 1 Mbps absolute throughput part to traffic classes such that the $\sum Qr_i^j$ of these classes is less than say 0.8 Mbps. For all of these traffic classes, the weight will be 1. The rest of the traffic will make use of the remaining capacity by using weights and obtain differentiated throughputs based on their allocated quantum rates. In the next section, we illustrate these mechanisms with numerical results.

V. PERFORMANCE COMPARISON AND EVALUATION

We performed our simulations using OPNET 9.0[2] . We used a 2 Mbps WLAN in our simulations to reuse some of the available models in OPNET rather than the more common 11 Mbps 802.11 system. Neither the AP nor the RTS/CTS mechanism were implemented. In this section, the performance of DDRR is compared to that of DCF, DFS and DWFQ with respect to the aggregate throughput achievable, the variability of the throughput, delay and fairness. We also demonstrate the ability of DDRR to support absolute throughput as well as relative throughput.

The simulation comprised of a network ranging from 10 to 50 MSs and the aggregated offered load ranging from 1.2 Mbps to 1.8 Mbps. There are two data rate classes. The high data rate stations have twice the offered load, weight, quantum rate as the low data rate stations as shown in Eq. 7. λ_l is the offered load of low data rate MS and λ_h is the offered load of high data rate MS, where $\lambda_h = \lambda_l * 2$. O is the total offered load ranging from 1.2 Mbps to 1.8 Mbps. N_l is the number of low data rate MSs and N_h is the number of high data rate MSs, where $N_h = N_l * 0.2.$, M is the total number of MSs ranging from 10 to 50 where $N_h + N_l = M$. For example, a network with 10 MSs with total offered load of 1.2 Mbps, 2 MSs are high data rate with offered load at constant rate of 200 Kbps. The rest 8 MSs are low data rate with offered load at constant rate of 100 Kbps. The packet size is constant at 1000 Bytes. The buffer size of sender is 256 Kbits. The value of δ is 3. The value of α is 1.25e-5. The value of Q is 100,000. In this simulation, each MS has only one class of traffic, however, our mechanism works for different classes of traffic per MS as well. Due to

---

[2] The authors would like to express their gratitude to OPNET for providing the simulation tool for this study.

space limitations, only a small subset of our results and figures are shown here.

$$\lambda_l N_l + \lambda_h N_h \quad = \quad O \tag{7}$$

A. Aggregate Throughput

The aggregate throughput of DDRR, DCF, DWFQ, and DFS as the number of MSs increases is shown in Fig 1. The total offered load is kept constant at 1.8 Mbps. We can see that DDRR can provide the highest and almost constant aggregate throughput, followed by DFS. The higher aggregate throughput of DDRR over DFS can be explained by the better handling of collisions by DDRR as well as the differences in queuing delay. The queuing delay of DDRR is smaller than that of DFS as shown in Figure 2. In this scenario where the buffers are always full, the longer the queuing delay, the average service rate of the mechanism. The lower average service time resulted in lower aggregated throughput. The service rate of DDRR mechanism is higher than that of DFS. The aggregate throughput of DCF and DWFQ are the lowest and decrease as the number of MSs increases.

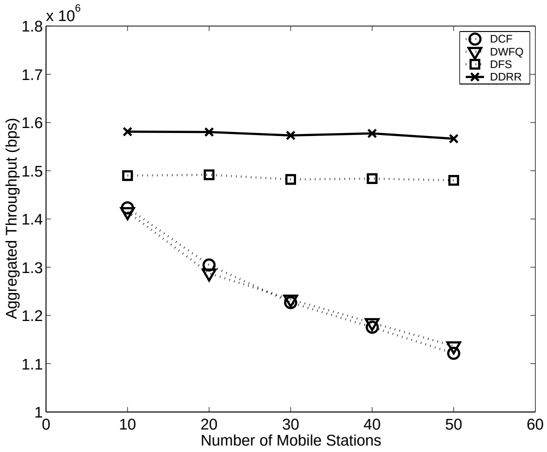

Fig. 1. Aggregate Throughput

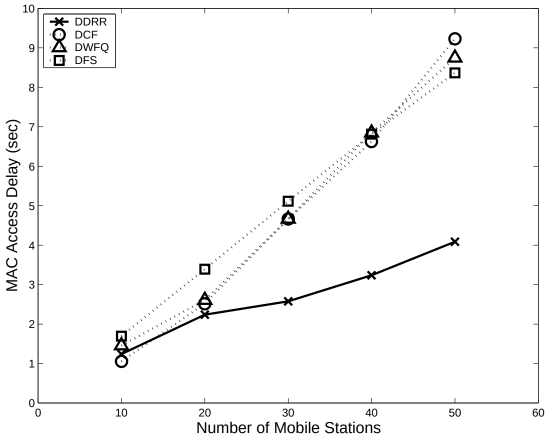

Fig. 2. MAC Queuing Delay

B. Throughput Variability

Figure 3 shows the instantaneous throughput of DDRR, DWFQ, and DFS for both the low and high data rate traffic when the total offered load is 1.8 Mbps (which is higher than the maximum achievable channel capacity). According to figure, the average throughput of low data rate MSs of DDRR, DWFQ, and DFS are 130 kbps, 122 Kbps, and 124 Kbps, respectively. The average throughput of high data rate MSs of DDRR, DWFQ, and DFS are 270 Kbps, 248 Kbps, and 249 Kbps, respectively. Although, the average throughput of each mechanism is similar, we can see that the instantaneous throughput is very different. The throughput variation of DFS is minimum, followed by that of DDRR, while the instantaneous throughput of DWFQ is wildly fluctuating.

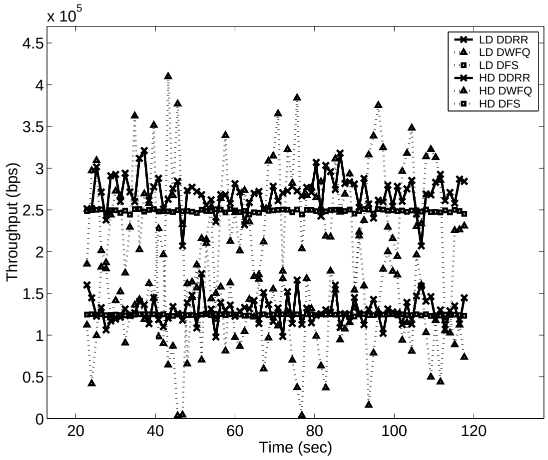

Fig. 3. Throughput Variability

C. Fairness

Figure 4 shows the ratio of the average throughput of high data rate traffic classes to that of the low data rate traffic classes for different schemes as a function of the number of stations. In the simulations, the quantum rate for DDRR and the flow weights of the high data rate traffic class was chosen to be twice that of the low data rate traffic class. With a mechanism that provides fair bandwidth allocation, the ratio of the average throughputs should also be equal to two. We can see from Figure 4 that DDRR and DWFQ are very fair whereas DFS exhibits very small fluctuations as the number of stations is increased. The results here are shown for an offered load of 1.2 Mbps.

In a configuration without an access point (AP) and no RTS/CTS signalling, the fairness of DWFQ degraded as the offered load and the number of MS increased (See Figure 5). We believe that the reason for the degradation of fairness is due to the significant variability of throughput that reduces the separation between high data rate and low data rate traffic, especially when the difference in absolute value between high data rate and low data rate throughput decreased as the number of MS increased. However, as presented in [6], in a configuration with an access point and using RTS/CTS signalling, the throughput variability for DWFQ decreases significantly and the fairness improves to the same levels as the DFS and DDRR schemes.

D. Relative Throughput and Absolute Throughput Support

In addition to the above performance comparison, Figure 6 shows a unique feature of DDRR that it can provide both absolute throughput and relative throughput support. In the experiment with 16 MSs, there were 4 high data rate MSs. Two of

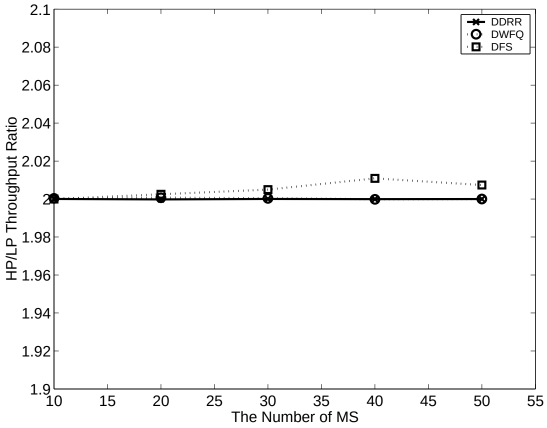

Fig. 4. Ratio of Throughput between High Priority and Low Priority at 1.2 Mbps Aggregated Offered

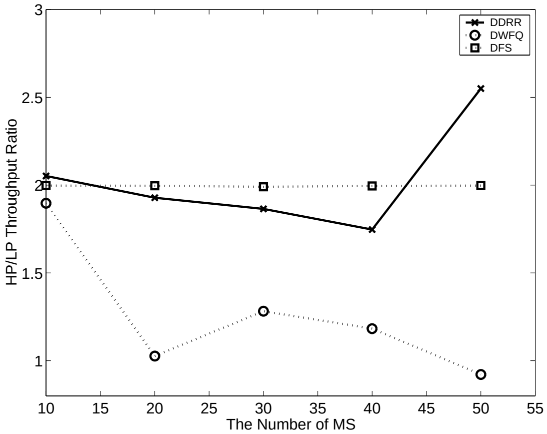

Fig. 5. Ratio of Throughput between High Priority and Low Priority at 1.8 Mbps Aggregated Offered

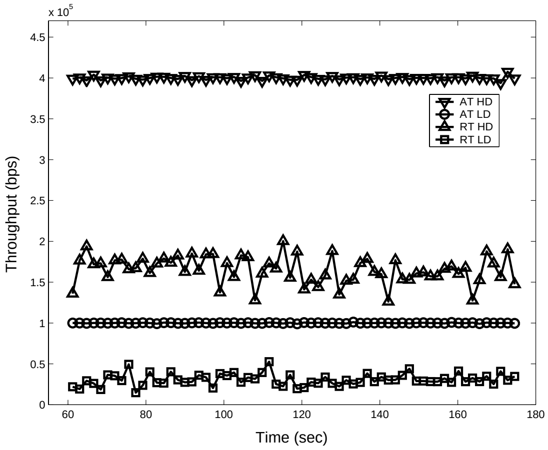

Fig. 6. Absolute Throughput vs. Relative Throughput

these required absolute throughput at 400 Kbps (AT HD) while the others required only relative throughput at 400 Kbps (RT HD). Among 12 low data rate MSs, two of which required absolute throughput at 100 Kbps (AT LD) while the rest required only relative throughput at 100 Kbps (RT LD). The parameter δ is set to one. We can see that the absolute throughput MS received exactly the desired throughput while the relative throughput MS fairly shared the rest of the available bandwidth based on the calculated weight from quantum rate. Thus, the throughputs achieved by the high data rate MSs requiring relative throughputs is 4.3 times that of the low data rate MSs requiring relative throughputs. The perfectly fair ratio is 4.0. We are currently studying the effect of δ parameter on fairness, aggregated throughput, and throughput variability in a greater detail.

VI. CONCLUSION

In this paper, we proposed an enhancement to the 802.11 MAC protocol that can provide both absolute throughput and relative throughput in a highly distributed fashion without compromising fairness. The proposed mechanism called Distributed Deficit Round Robin (DDRR) is based on deficit round robin scheduling. Absolute throughput and relative throughput are supported by translating the user throughput requirements into the 802.11 MAC InterFrame Space and Backoff Interval parameters. We show via simulations that the proposed mechanism provides low variability of throughput and delay and has the advantage of low complexity.

This work is similar to recent fair queue scheduling mechanisms [7] and [6]. We believe it has the advantages over these schemes in terms of complexity, ease of translating user requirements into a tunable MAC parameter, and, more importantly, dual QoS support, i.e. absolute throughput and relative throughput. Our simulation results demonstrated that the performance of DDRR is better than DWFQ in terms of higher aggregate throughput, lower throughput variability, dual QoS support, and no change in frame header requirement. Our scheme also performs better than DFS in terms of higher aggregate throughput, dual QoS support, and lower complexity.

REFERENCES

[1] 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, pp. 96–102, 1999.
[2] I. Aad and C. Castelluccia, “Differentiation mechanisms for IEEE,” in *INFOCOM*, 2001.
[3] G. Chesson, M. Benvenisite, W. Diepstraten, M. Hoeben, A. Singla, H. Teunissen, and M. Wentink, “EDCF Proposed Draft Text,” Tech. Rep., IEEE working document 802.11-01/131r1, 2001.
[4] A. Veres, A.T. Campbell, M. Barry, and Li-Hsiang Sun, “Supporting service differentiation in wireless packet networks using distributed control,” *IEEE Journal on Selected Areas in Communications*, 2001.
[5] K. Kim, S. Shin, and K. Kim, “A Novel MAC Scheme for Prioritized Services in IEEE 802.11a Wireless LAN,” in *International Conference on ATM and high speed intelligent Internet (ICATM)*, 2001.
[6] A. Banchs and X. Perez, “Distributed weighted Fair queuing in 802.11 wireless LAN,” in *ICC*, 2002.
[7] N. H. Vaidya, P. Bahl, and S. Gupta, “Distributed fair scheduling in a wireless LAN,” in *Proceedings of Mobile Computing and Networking Conference*, 2000, pp. 167–178.
[8] A. Banchs and X. Perez, “Providing throughput guarantees in IEEE 802.11 wireless LAN,” in *WCNC2002*, 2002.
[9] W. Pattara-atikom, P. Krishnamurthy, and S. Banerjee, “Distributed Mechanisms for Quality of Service in Wireless LANs,” *IEEE Wireless Communications Magazine*, June 2003.
[10] S.J. Golestani, “A self-clocked fair queueing scheme for broadband applications,” in *INFOCOM*, 1994.
[11] M. Shreedhar and G. Varghese, “Efficient fair queuing using deficit round-robin,” *IEEE/ACM Transactions on Networking*, 1996.
[12] W. Pattara-atikom, S. Banerjee, and P. Krishnamurthy, “Starvation Prevention and Quality of Service in Wireless LANs,” in *WPMC*, 2002.