DAT: An AP Scheduler using Dynamically Adjusted Time Windows for Crowded WLANs

Yi Yao
Northeastern University
Email: yyao@ece.neu.edu

Bo Sheng
University of Massachusetts Boston
Email: shengbo@cs.umb.edu

Ningfang Mi
Northeastern University
Email: ningfang@ece.neu.edu

***Abstract*—This paper proposes a new packet scheduling algorithm for access points in a crowded 802.11 WLAN. Our goal is to improve the performance of efficiency (measured by packet response time or throughput) and fairness which often conflict with each other. Our solution aggregates both metrics and leverages the balance between them. The basic idea is to let the AP allocate different time windows for serving each client. According to the observed traffic, our algorithm dynamically shifts the weight between efficiency and fairness and strikes to improve the preferred metric without excessively degrading the other one. A valid queuing model is developed to evaluate the new algorithm's performance. Using trace-driven simulations, we show that our algorithm successfully balances the trade off between the efficiency and the fairness in a busy WLAN.**

I. Introduction

Wireless LANs (WLANs) have been well deployed nowadays providing the last mile delivery of Internet access to mobile clients. For example, wigle.net has reported more than 33 million observed WiFi networks, and SkyHook a WiFi-based localization service, has claimed to have "tens of millions" access points (APs) in its database. With the dense deployment of WiFi infrastructure and the evolution of 802.11 family, WLANs keep playing an extremely important role in serving mobile clients. When more and more people carry WiFi-enabled devices, such as laptop, smartphone, and iPad, WLANs are often crowded, especially at particular locations with special events, e.g., a meeting room for a large conference or a stadium hosting a sports game. Under a heavy traffic load, WLAN clients may encounter serious performance degradation due to channel contention and interference.

In this paper, we aim to improve the performance of a busy WLAN by designing a new scheduling algorithm for APs. We focus on the packet scheduling of the downlink traffic, i.e., from the AP to clients, as it carries the majority of data. Typically, an AP applies First-In-First-Out (FIFO) scheduling discipline, i.e., the first packet arriving from the Internet will be first sent via the wireless channel. The FIFO strategy works well with idle traffic, where each downlink packet can be immediately sent to a client with little delay at the AP. However, under heavy traffic, downlink packets may not be delivered right after arrivals at APs. Instead, each AP maintains a queue to buffer the incoming packets from the Internet. Every packet in the queue has different attributes such as packet size and transmitting rate (according to the destination client). The simple FIFO policy, however, ignores all these characteristics and can barely yield the optimal performance. For example, when the AP delivers a packet with low rate, the response times of all other packets in the queue are increased by the transmission time. Thus, it would be better to deliver the packets with high rates first.

When designing a new scheduling algorithm, we mainly consider two metrics, *link efficiency* measured by packet response time or link throughput and *fairness* among all the clients. When an AP serves multiple clients, the wireless link quality between each client and the AP is different. In order to improve the efficiency, the AP prefers to first send the packets through the fastest link. However, some other clients with slow links may suffer from starvation causing unfairness. In this paper, we propose a new scheme, named DAT, that dynamically adjusts the time windows allocated to each client and strikes the balance between efficiency and fairness.

Our basic idea is to combine Round-Robin, which achieves the best fairness, with an adaptive time window for service. The AP rotates all active clients and delivers the packets in the buffer for them one client after another. Each client is assigned a time window for serving its packets, i.e., during a time window, the AP continuously sends the packets to a particular client. Our DAT scheme dynamically adjusts the time window for each client according to the observed efficiency and fairness values. Our goal is to aggregate these two metrics and balance the performance of them.

In summary, our major contributions in this paper are: 1) We propose a novel AP scheduling algorithm that aggregates the efficiency and the fairness considerations. 2) We build a queuing model that captures the behaviors of WLAN clients and the AP for the performance evaluation. 3) We conduct comprehensive trace-driven simulations to evaluate our proposed scheme and compare to two classic policies. The simulation results show that our algorithm is superior for a crowded WLAN.

The rest of the paper is organized as follows. Section II summarizes the prior work and Section III presents our new scheduling algorithm. In Section IV, we introduce the queuing model for evaluation. The simulation results are reported in Section V. Finally, we conclude in Section VI.

II. Related Work

Throughput and fairness are traditional metrics for network packet scheduling. They have also been well studied

978-1-4673-0012-4/11/$26.00 ©2011 IEEE

in WLANs' literature. One direction particularly works on the TCP flows [1]–[5]. The key problem is to handle head-of-line blocking and the competition between TCP data packets and TCP ACKs, especially when considering the channel errors. While this paper focuses on the MAC layer scheduling, the prior work on TCP flows can certainly be combined with our solution to form a cross-layer scheduling scheme.

Another direction in the prior work is to improve throughput and fairness by managing the whole WLAN such as strategically associating clients to APs [6]–[11] and assigning channels to APs [12], [13]. Most of these work requires central control and coordination. The prior work here is also complementary to our work in this paper which handles the packet level scheduling on a single AP.

Furthermore, some previous work [14], [15] proposed to achieve time-based fairness in WLANs via rate adaptation with modifications on 802.11 standards. Their solutions focus on the rate adaptation algorithms and require modifications on 802.11 standards. Our work targets on MAC layer scheduling and we consider the fairness of throughput.

Also, some previous work focuses on the improvement of efficiency. For example, [16] proposed an embedded round robin policy which improves the efficiency by reducing the time for polling idle stations. Kar., K. etc. [17] presented a throughput-optimal scheduling policy when the time-varying channel rate can be measured only infrequently. However, the metric of fairness is not considered in these papers.

Finally, wireless packet scheduling is often studied to provide QoS represented by the IEEE 802.11e standard [18]. In 802.11e EDCA mode, high priority traffic is given shorter delay-related parameters such as contention window (CW) and arbitration inter-frame space (AIFS) so that it has better chance to be sent than low priority traffic. Our proposed scheduler is also complementary to QoS provision scheme. While 802.11e distinguishes traffics with different priorities, our scheme can be used to schedule the traffics with the same priority.

III. New AP Scheduling Algorithm: DAT

A. System Model and Overview

In this work, we consider an AP serving n clients, $\{c_1, c_2, \ldots, c_n\}$, in a busy wireless LAN. We assume that the clients are ordered based on their effective downlink rates, considering the MAC layer transmitting rates, acknowledgment, retransmission, and other per-packet overheads. It follows that client c_1 has the slowest link to the AP and c_n is connected with the fastest link. When the AP is over-loaded, the downlink packets may be buffered in a queue at the AP before being sent out. Normally, the queue has a capacity limit indicated by a maximum number of packets that can be held in the buffer. Later in Section V, we show the evaluation under infinite queue capacity as well.

In this paper, we consider two metrics as the performance objective of an AP scheduling algorithm: efficiency and fairness. The first metric is measured by the packet response time or alternatively by the throughput. The second metric of fairness is measured by the Jain's index. Despite that both metrics are critical in the scheduler evaluation, it is often difficult to improve them simultaneously under a particular AP scheduling policy. For example, one can use Round Robin (RR) to achieve the best fairness. By rotating among all active clients, RR always delivers one packet for each client in a round. This policy can certainly avoid starvation, but the efficiency under RR is very poor. In contrast, the other extreme of scheduling policy (MaxTP) is to always give higher priority to faster clients, i.e., keep sending the available packets to a client which has the highest downlink rate. As a result, the optimal efficiency in terms of packet throughput or packet transmit delay time is achieved under MaxTP while poor fairness often becomes a big problem under this policy because it unfairly treats the clients with slow downlink rates.

How to balance the trade-off between efficiency and fairness is imminently important and challenging in the AP scheduler design. In this paper, we propose a new scheduling algorithm, named DAT, which takes into account both metrics in scheduling the downlink packets at the AP and strikes to obtain the fairness and the efficiency close to the optimal results provided by RR and MaxTP respectively.

B. Algorithm Description

Our new algorithm adopts the basic idea of rotating clients to serve from the RR scheme. However, we assign *different* service time to each client.

In our scheme, the AP selects a client c to serve and allocates a time window for service. The AP keeps sending the packets for client c till the assigned time window is elapsed or there is no more packets available for client c. Then the AP will select another client and start delivering its packets. Different from the RR scheme, the duration of time window for each client is dynamically adjusted across time by considering the trade-off between efficiency and fairness. We consider that the AP selects a window size from a set of discrete values to serve a client. Let w denote the minimum time slice (finest granularity), e.g., $w = 0.01$ seconds. Then a pool of k candidate values for the window size is represented by $\{1w, 2w, 3w, ..., kw\}$. In our DAT scheme, the AP chooses the best window size for each client from these k values based on the following two target functions.

"Relative Efficiency" function: This function expresses the relative efficiency gain for a particular choice of the window size. Intuitively, if efficiency is the only concern, then a good policy (e.g., MaxTP) should always consider large windows for clients with fast link rates in order to empty the AP buffer as soon as possible, resulting in short packet response times, high system throughput and high system availability. To characterize the effect of the window size, we define the *Relative Efficiency* function as follows:

$$\alpha_i = i \cdot w \cdot \frac{Rate_c - \overline{Rate}}{\overline{PackSize}}, \forall i \in [1, k], \qquad (1)$$

where $Rate_c$ represents the link rate of the selected client, $\overline{Rate}$ represents the average link rate of all the remaining active clients that have packets in the AP buffer, and $\overline{PackSize}$

is the mean size of the packets. A higher value of α_i indicates more packets expelled from the queue, thus a better efficiency is achieved. Based on Eq.(1), if $Rate_c > \overline{Rate}$, then the AP prefers to allocate a large window (or the largest window if α_i is the only metric) to client c . Otherwise, if α_i becomes negative, the AP would reduce the window size as much as possible. We remark that α_i provides a good indication of efficiency that the policy can achieve when the AP assigns a particular time window to a client.

"Expected Fairness" function: Our second target function is designed to quantify the fairness. As mentioned earlier, we use the Jain's fairness index as the metric to measure the fairness of a given scheduling policy. Eq.(2) gives the definition of Jain's fairness index I:

$$I = \frac{(\sum_{j=1}^{n} TP_j)^2}{n \sum_{j=1}^{n} {TP_j}^2}, \tag{2}$$

where n is the number of active clients and TP_j represents the throughput of client j in a predefined time period. The range of I is between 0 and 1, and a higher value of I indicates a better fairness. The main goal of our second target function, named *Expected Fairness*, is to estimate the packet throughput among all active clients and thus express the performance of fairness. To accomplish it, DAT online tracks each client's throughput in the past monitoring window t and uses this information to decide the duration of the time window for the current client. The intuition is that if the client has already received a higher throughput in the previous monitoring window, then a smaller time window should be chosen for that particular client in the next round, and vise versa. Note that the size of monitoring window t is a user-specific parameter and we will describe its setting in Section V.

Given a client c and a candidate time window $i \cdot w$, we have the following equations to calculate the expected throughput for all n clients if DAT decides to send the packets to client c during the $i \cdot w$ time period:

$$TP_j = \begin{cases} \frac{S_j}{t+i \cdot w}, & \text{for } \ j \neq c, \\ \\ \frac{S_c+s}{t+i \cdot w}, & \text{for } \ j = c, \end{cases} \tag{3}$$

where S_j represents the total amount of data transmitted to client j during the previous monitoring window t, and s equals to the estimated amount of data that can be transmitted to client c within the $i \cdot w$ time window, i.e., $s = Rate_c \cdot i \cdot w$. Let $\widehat{Sum} = \sum_{j=1}^{n} S_j$ and $\widehat{Sum}_2 = \sum_{j=1}^{n} {S_j}^2$. We then express the *Expected Fairness* target function as follows:

$$\beta_i = \frac{(s + \widehat{Sum})^2}{n(\widehat{Sum}_2 + 2 \cdot s \cdot S_c + s^2)}. \tag{4}$$

Given the above two target functions, DAT further uses the 0-1 scaling technique to scale α_i and β_i for all k candidate time windows (i.e., $1 \leq i \leq k$) as follows.

$$\alpha'_i = \frac{\alpha_i - \alpha_{min}}{\alpha_{max} - \alpha_{min}}. \tag{5}$$

$$\beta'_i = \frac{\beta_i - \beta_{min}}{\beta_{max} - \beta_{min}}. \tag{6}$$

Then, DAT selects the best window size for the current client based on the following equation:

$$P_i = w_1 \cdot \alpha'_i + w_2 \cdot \beta'_i, \tag{7}$$

where w_1 and w_2 are the user-defined weights for α_i and β_i, respectively. We expect that higher P_i will achieve better efficiency/fairness balance. Therefore, the time window which can get the highest value of P_i will then be assigned to the current client. The major steps of DAT are presented in Fig. 1.

Algorithm: DAT
begin
1. choose an active client c based on round-robin rotation and set current time as t_0;
2. assign time window size for the chosen client c;
  a. for $i = 1$ to k
    I. calculate α_i using Eq.(1);
    II. calculate β_i using Eq.(4);
  b. $P' \leftarrow 0$, $i' \leftarrow 0$;
  c. for $i = 1$ to k
    I. scale α_i and β_i to range [0,1] by 0-1 scaling method;
    II. calculate P_i using Eq. (7);
    III.if $P_i > P'$
      then $P' \leftarrow P_i$ and $i' \leftarrow i$;
  d. assign window size $i' \cdot w$ to client c;
3. send packets to client c;
  a. if current time $t < t_0 + i' \cdot w$ and number of packets from client $c > 0$
  then
    I. send a packet to client c;
    II. update history information S_c, $\widehat{Sum}$, and $\widehat{Sum}_2$;
  else go to **step 1**;
end

Fig. 1. The high level description of DAT.

IV. Simulation Model

In this section, we present a queuing model built for evaluation. We consider the circumstance where requests from multiple clients and the corresponding reply packets from the AP are all sent through the same wireless channel. Fig. 2 illustrates the model that captures the behavior observed in the single AP situation and evaluates the performance of different scheduling algorithms under heavy load conditions.

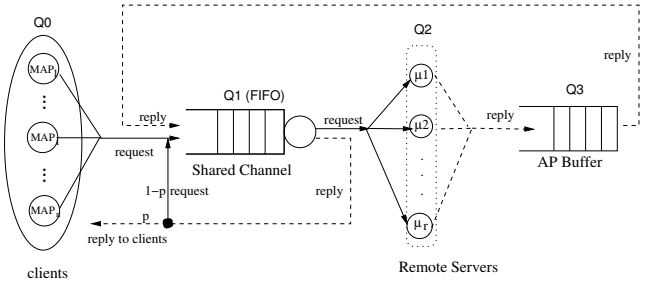

Fig. 2. A queuing model of the single AP wireless network.

In this model, an infinite queue (Q_0) with n servers is used to emulate n clients' activity in the system, where each server represents a single client, independently sending requests to the AP. We model the request inter-arrival times as a Markovian Arrival Process (MAP) [19] for each server in Q_0 such that each client can have different arrival rates

and different arrival distributions. The specifications of each request include request arrival times, request sizes in bytes, client index, and effective uplink/downlink rates. All requests generated by Q_0 are then enqueued to the queue Q_1, waiting for the transmission in the shared wireless channel. We remark that in the real wireless network, clients usually trigger a retry mechanism if there are simultaneous contentions for the shared channel. Since our focus is on AP's scheduling for downlink packets, we here simply ignore such a retry mechanism and instead transmit these requests in the order of their arrival times using the FIFO discipline at channel queue Q_1.

We further introduce a delay center Q_2 to model the processing times at remote servers, which process a received client request with a fixed time $1/\mu_i$ and then send the corresponding reply data back to the AP. As transmitted packets usually have a limited size, e.g. 1.5K bytes, if a reply data set is larger than that particular limit, then the original one will be partitioned into several packets and transmitted at the packet level. It follows that instead of having one to one relationship between requests and reply packets, remote servers in the delay center Q_2 might generate m reply packets for each arrived request, where m is determined by the original reply data size and the size limit.

These reply packets received from the remote servers are then queued in the AP buffer, shown as Q_3 in Fig. 2, waiting for the service or transmission at the shared channel Q_1. When detecting no reply packet waiting or serving at Q_1, the AP chooses a reply packet from the buffer and enqueues it to Q_1 immediately. The selection of next transmitted reply packet is done according to different scheduling disciplines. For example, if the MAXTP policy is considered, then the AP buffer can be implemented as a priority queue based on transmit rates. Consequently, the AP always selects a reply packet with fastest downlink rates. In real wireless network, the transmitted reply packets might trigger one or several client requests after some delay time. We capture this behavior by adding a branch probability for reply packets at Q_1: with probability p a completed packet at Q_1 is simply forwarded to its associated client and with probability $1-p$, a batch of client requests (≥ 1) are sent back to the channel queue.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our DAT and compare the results to the other two classic policies, RR and MAXTP.

A. Evaluation Settings

Workload: We conduct simulations with the following three synthetic trace sets which represent different scenarios.

- *Base case*: The requests from each client follow a random arrival pattern with the same mean request arrival rate.
- *Burst case*: In this case, we choose the top two fastest clients and set idle and burst periods in the request arrival trace of these two clients. The requests from other clients are the same as in the base case.
- *Uneven case*: In this case, we set the request rates of the top two fastest clients to be five times higher than those of other clients.

We use the trace from SIGCOMM 2008 [20] to generate requests and reply packets. The mean size of requests is 322 bytes and the mean size of reply packets is 1004 bytes.

Default parameters: In our default setting, there are totally $n = 20$ clients. Each client has the fixed uplink and downlink rates throughout the whole simulation, and the link rate ranges from 100KB to 1MB. Without loss of generality, clients with larger index have faster link rates, such that client 20 has the highest uplink and downlink rates among all clients. By default, we set the minimum time slice $w = 0.01$sec, the number of candidate window sizes $k = 10$ and the branch probability $p = 1$. The user-specific parameters are set as follows: $t = 0.5$sec (size of the monitoring window), $w_1 = 1$ and $w_2 = 2$ (weights in Eq.(7)).

For each trace set, we further consider both infinite buffer size situation and finite buffer size situation when measuring performance. With the setting of infinite buffer size, we measure the average response time for efficiency and average Jain's index for fairness. When considering finite buffer size, we additionally measure the number of dropped packets and the corresponding drop ratios.

Fairness measurement: In our simulation, the fairness index is measured across time within a 0.25-second time window. We also tried other time window lengths (e.g., 0.5 sec, 1 sec), which result qualitatively the similar results. In each time window, we only consider the active clients for calculating the fairness index. We define a client to be active in a certain time window if it has sent requests during this time window or if it has a pending request (sent in past time windows) that has not been replied. In addition, we ignore the time windows with no active clients or only one active client since there is no fairness issue in such a situation.

Scale rate: To clearly illustrate how well our DAT balances between the efficiency and the fairness, we further present the relative scale rate using the 0-1 scaling technique as follows: among all the polices (e.g., RR, DAT and MAXTP), we scale the best performance to 1 and the worst performance to 0 and then normalize our DAT's performance between 0 and 1, see Eq.(8).

$$Scale\ rate = \frac{|\text{DAT} - Worst|}{|Best - Worst|}. \tag{8}$$

Therefore, a larger-than-0.5 relative scale rate indicates that DAT performs closely to the best policy, e.g., with shorter response time or larger fairness index. If the relative scale rate is smaller than 0.5, then it implies the opposite. If the values with respect to both response time and fairness index are greater than 0.5, then our DAT obtains a well balance between the efficiency and the fairness.

B. Performance Improvement

1) Base Case: In this case, each client's mean request arrival rate was generated independently through a 2-state

Markov-Modulated Poisson Process (MMPP), which is a special case of the Markovian Arrival Process (MAP) [19]. We stress that distributions of modern network traffic, e.g., packet and connection arrivals are no longer Poisson distributed [21]. Therefore, we investigate heavy-tailed WLAN packet arrival processes where the mean arrival rate of each client c_i is set to $\lambda_i = 1.5$ *per sec*, and the coefficient of variation (CV) at the arrival process is equal to 5.

The simulation results under the three scheduling policies are shown in Table I and Table II with infinite and finite buffer size (maximum 800 packets in the buffer), respectively. The numbers in parentheses are the relative scale rates of DAT. As we can see, in both infinite buffer and finite buffer settings, mean response time of our DAT is close to MAXTP and mean fairness index is close to RR. For example, compared to RR, when using infinite buffer, DAT improves the efficiency (e.g., response time) by 50 percent yet only degrades the fairness by 18 percent. The corresponding scale rates in terms of response time and fairness index are both more than 0.5.

	RR	DAT	MAXTP
RespTime(s)	2.301	1.135(0.82)	0.885
FairIndex	0.766	0.626(0.56)	0.450

TABLE I
PERFORMANCE UNDER BASE CASE WITH INFINITE BUFFER SIZE.

	RR	DAT	MAXTP
RespTime(s)	0.866	0.714(0.69)	0.647
FairIndex	0.758	0.641(0.70)	0.369
DropRatio(%)	2.010	1.012(0.81)	0.773

TABLE II
PERFORMANCE UNDER BASE CASE WITH FINITE BUFFER SIZE.

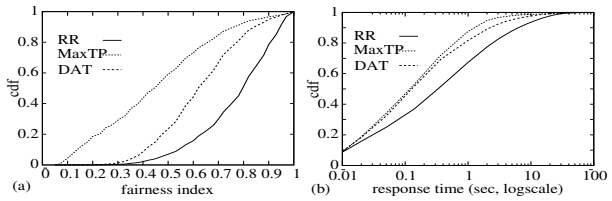

Fig. 3. Illustrating (a) CDFs of fairness index, and (b) CDFs of response time for RR, MAXTP, and DAT, where $k = 10$ and buffer size is infinite under base case.

Fig. 3 further illustrate more details about the performance comparison of the three different scheduling policies. According to the cumulative distribution functions (CDFs) of fairness index and response time per packet, it is apparent that DAT policy achieves the fairness close to RR and the efficiency performance close to MAXTP. The minimum fairness index under DAT is about 0.3, which is close to the minimum one under RR, while the minimum index in MAXTP is less than 0.1 indicating that most clients are starved during that time period.

2) Burst Case: In our burst case workload, the mean request arrival rate of each client is the same as in the base case. Yet, a bursty access pattern is introduced into the arrivals of two clients with fastest link rates, i.e., client 19 and 20, such that the SCV of their inter-arrival times is equal to 20 and the autocorrelation function (ACF) at lag 1 is equal to 0.47. Therefore, high variability and strong temporal dependence are injected to the workload of these two clients.

As shown in Table III, all three policies encounter significant performance degradation in terms of response time and drop ratio. The efficiency of DAT seems not as good as we observed in the base case. The relative improvement of average response time is only 22 percent compared to RR. However, if we examine the average response times of those clients without bursty patterns, i.e., client 1 to 18, then the average response times of both RR and DAT decrease while the average response time of MAXTP on the contrary increases. Consequently, DAT becomes the most efficient policy for those non-bursty clients. Since the bursty patterns are injected into the arrivals of the top two fastest clients, MAXTP achieves good efficiency by giving them high priority, but on the other hand, sacrifices the performance of other clients, resulting in extremely serious unfairness. Fig. 4 (b) further demonstrates such unfairness under MAXTP where it always degrades the performance of clients which have slow link rates.

Another interesting point in this case is that the mean fairness index value of MAXTP is counter-intuitively better than that in the base case. In fact, this is caused by the property of Jain's fairness index, which is sensitive to the number of active clients. For example, considering there are now only two active clients, even under the extreme scenario where one client is starved during the whole period, the index value will be equal to 0.5 which is still relatively good. In the burst case, the AP also experiences more idle periods where the number of active clients is small. In such an idle time period, the difference of fairness index values among the three policies is thus reduced.

3) Uneven Case: Now, we turn to investigate different request arrival rates. In order to keep the overall request arrival rate similar as the previous two cases, we scale the arrival rates of selected clients (e.g., 19, 20) to 6 *per sec* and decrease the rates of other clients (e.g., 1$\sim$18) to 1.2 *per sec*. In addition, the SCV of all request arrival traces is equal to 5 and no clients have bursty patterns in their arrival flows. The results shown in Table IV validate that our DAT still works well in this case, consistently achieving a good balance between efficiency and fairness.

Fig. 4 (c) further presents the average response time of each client's reply packets. Recall that in our evaluation setting, the link rate linearly increases as the client index increases and clients with larger index always have faster link rates. Thus, one can clearly observe that MAXTP always degrades the performance of one or two clients with slowest link rates. While under RR, clients which are responsible for the buffer congestion suffer significant performance degradation and the other clients have almost the same response times despite of their varying link rates. Also, observe that our DAT always punishes the clients that cause the buffer congestion in order to improve the efficiency of other clients. On the other hand, DAT strikes to give clients with faster link rates better performance, which fortunately is not too aggressive to degrade the fairness as MAXTP does.

	InfiniteBuffer			FiniteBuffer			
	Resp(s)	Resp*(s)	FairIndex	Resp(s)	Resp*(s)	FairIndex	DropRatio(%)
RR	5.921	1.490	0.759	0.734	0.658	0.750	3.329
DAT	4.594 (0.27)	0.849 (1.00)	0.662 (0.72)	0.623 (0.95)	0.538 (1.00)	0.675 (0.69)	2.866 (0.28)
MAXTP	1.052	1.152	0.417	0.617	0.662	0.507	1.682

TABLE III
PERFORMANCE UNDER BURST CASE (RESP* REPRESENTS THE MEAN RESPONSE TIME OF CLIENTS WITHOUT BURSTINESS).

	InfiniteBuffer		FiniteBuffer		
	Resp(s)	FairIndex	Resp(s)	FairIndex	DropRatio(%)
RR	1.118	0.729	0.476	0.725	1.091
DAT	0.500 (0.74)	0.625 (0.58)	0.360 (0.62)	0.650 (0.61)	0.425 (0.68)
MAXTP	0.287	0.480	0.289	0.535	0.108

TABLE IV
PERFORMANCE UNDER UNEVEN CASE

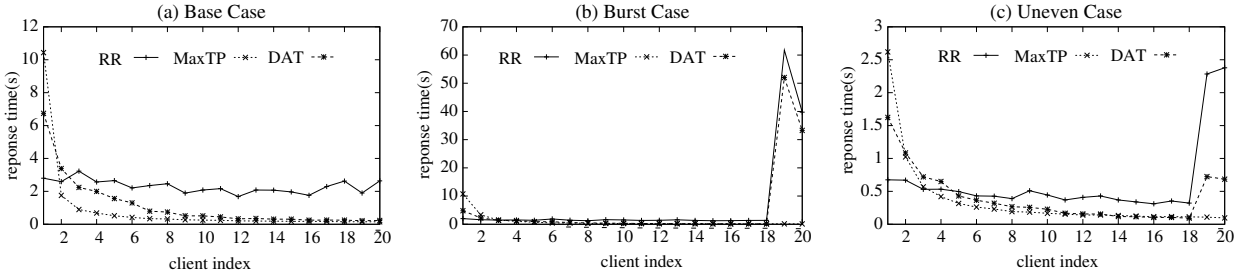

Fig. 4. Average response times of each client for RR, MAXTP, and DAT with $k = 10$ under (a) base case, (b) burst case, and (c) uneven case and infinite buffer size scenario.

VI. CONCLUSION AND FUTURE WORK

When WLANs become more and more popular, they are often crowded and the clients experience significant performance degradation. To solve the issue, we proposed a new packet scheduling algorithm DAT for heavily-loaded APs, which uses the basic Round-Robin scheme to select a client for service, but allocates each client with a different time window. The representitive case studies carried out in this paper have revealed that DAT significantly improves the performance in terms of efficiency and fairness in a crowded WLAN, reducing the packet response times and avoiding the starvation especially for clients with poor link quality. We also show that by dynamically adjusting the window size, DAT leverages the balance between the efficiency and the fairness.

Our future work mainly includes two directions. First, we would like to implement the scheduling algorithm on commercial wireless routers and conduct experiments for evaluation. Second, we plan to explore a new algorithm as well as a new model in a setting of multiple APs with possible coordination.

REFERENCES

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

[2] M. Bottigleliengo, C. Casetti, C.-F. Chiasserini, and M. Meo, "Short-term fairness for TCP flows in 802.11b WLANs," in *INFOCOM 2004*.

[3] G. Urvoy Keller and A.-L. Beylot, "Improving flow level fairness and interactivity in WLANs using size-based scheduling policies," in *MSWiM'08*.

[4] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathit, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling," in *INFOCOM'96*.

[5] P. Bhagwat, P. Bhattacharya, A. Krishma, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve tcp throughput over wireless lans," *Wirel. Netw.*, vol. 3, pp. 91–102, March 1997.

[6] A. Balachandran, P. Bahl, and G. M. Voelker, "Hot-Spot congestion relief in public-area wireless networks," in *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, ser. WMCSA '02, 2002, pp. 70–.

[7] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," in *MobiCom '04*.

[8] N. Ahmed and S. Keshav, "SMARTA: a self-managing architecture for thin access points," in *CoNext '06*.

[9] A. P. Jardosh, K. Mittal, K. N. Ramachandran, E. M. Belding, and K. C. Almeroth, "IQU: practical queue-based user association management for WLANs," in *MobiCom '06*.

[10] H. Lee, S. Kim, O. Lee, S. Choi, and S.-J. Lee, "Available bandwidth-based association in IEEE 802.11 wireless LANs," in *MSWiM '08*.

[11] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley, "Facilitating access point selection in IEEE 802.11 wireless networks," in *IMC '05*.

[12] S. Manitpornsut, B. Landfeldt, and A. Boukerche, "Efficient channel assignment algorithms for infrastructure WLANs under dense deployment," in *MSWiM '09*.

[13] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *MobiCom '06*.

[14] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *MobiCom '02*.

[15] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate WLANs," in *Proceedings of the annual conference on USENIX Annual Technical Conference*, 2004.

[16] D. H. D. E. R.S. Ranasinghe, L.L.H. Andrew, "Scheduling disciplines for multimedia wlans: embedded round robin and wireless dual queue," in *ICC '01*.

[17] X. L. S. S. Kar., K., "Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements," in *INFOCOM '07*.

[18] "802.11e amendment." [Online]. Available: http://standards.ieee.org/getieee802/download/802.11e-2005.pdf

[19] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Philadelphia PA: SIAM, 1999, aSA-SIAM Series on Statistics and Applied Probability.

[20] "Sigcomm 2008 trace." [Online]. Available: http://www.cs.umd.edu/projects/wifidelity/sigcomm08_traces/

[21] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 6 1995.