MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Achieving Fair Scheduling Over Error-Prone Channels in Multirate WLANs

Yuan Yuan, Daqing Gu, William Arbaugh, Jinyun Zhang

TR2005-082   December 2005

Abstract

Wireless packet scheduling has been a popular paradigm to provide packet-level quality of services (QoS), in terms of throughput, delay and fair sharing, over error-prone channels. However, the state-of-the-art scheduling solutions are designed for single-rate environments. They cannot exploit the multi-rate capacity offered by the physical layer of the current WLANs. In this paper, we propose Adaptive Wireless Fair Scheduling (AWFS), which oppotunistically exploits the multirate feature and provides packet-level QoS in the presence of channel errors. AWFS departs from the throughput-oriented fairness and compensation, and adopts temporal fair sharing and fair compensation defined in virtual time. It offers a packet-level solution that ensures virtual temporal share, works with variable packet size and occaasionally idle flows, and operates in the infrastructure mode. AWFS improves overall channel utilization while nsuring fair sharing. It can operate with current 802.11b/e MAC design. through both analysis and simulations, we evaluate the effectiveness of AWFS. The simulations show that AWFS can achieve signficant throughput gain by improving overall throughput up to 159% compared with state-to-the-art scheduling algorithms in the simulated scenarios.

International Conference on Wireless Networks, Communications and Mobile Computing

Achieving Fair Scheduling Over Error-Prone Channels in Multirate WLANs

Yuan Yuan*, Daqing Gu†, William Arbaugh*, Jinyun Zhang†
* Computer Science Department
University of Maryland, College Park, MD 20740
Email: yuanyuan, waa@cs.umd.edu
†Mitsubishi Electric Research Laboratories, Cambridge, MA 02139
Email: dgu, jzhang@merl.com

***Abstract*— Wireless packet scheduling has been a popular paradigm to provide packet-level quality of services (QoS), in terms of throughput, delay and fair sharing, over error-prone channels. However, the state-of-the-art scheduling solutions are designed for single-rate environments. They cannot exploit the multi-rate capability offered by the physical layer of the current WLANs. In this paper, we propose Adaptive Wireless Fair Scheduling (AWFS), which opportunistically exploits the multirate feature and provides packet-level QoS in the presence of channel errors. AWFS departs from the throughput-oriented fairness and compensation, and adopts temporal fair sharing and fair compensation defined in *virtual* time. It offers a packet-level solution that ensures *virtual* temporal share, works with variable packet size and occasionally idle flows, and operates in the infrastructure mode. AWFS improves overall channel utilization while ensuring fair sharing. It can operate with current 802.11b/e MAC design. Through both analysis and simulations, we evaluate the effectiveness of AWFS. The simulations show that AWFS can achieve significant throughput gain by improving overall throughput up to 159% compared with state-to-the-art scheduling algorithms in the simulated scenarios.**

I. Introduction

Wireless LAN technology based on the IEEE 802.11 standard has become increasingly popular in corporate and campus environments to provide users untethered Internet access. In order to improve radio spectrum utilization, the IEEE 802.11a/b/g/n specifications offer a physical-layer multi-rate capability [1]. Specifically, in IEEE 802.11b, users can transmit at 1, 2, 5.5, or 11Mbps, whereas in 802.11a, eight rate options are allowed at 6, 9, 12, . . . , and 54Mbps, and the upcoming standard IEEE 802.11n expects to deliver much higher and more diverse transmission rates. The multi-rate feature allows a wireless host to choose the best transmission rate depending on its perceived channel quality measured by the signal-to-noise ratio (SNR), and the better channel quality promises the higher transmission rate.

The multirate option poses new challenges for network protocol design in the context of wireless packet scheduling [3]. If used properly, this new option can greatly improve the system throughput and effectively support communication-intensive multimedia and data applications in WLANs.

Packet scheduling, notably fair queueing, has long been a popular paradigm [10], [11], [14] to provide packet-level quality of services (QoS) in terms of throughput, delay and fair sharing, thus enabling both delay-sensitive and throughput-sensitive applications. Wireless packet scheduling [3], [8], [9], [12], [13] achieves wireless QoS by further addressing the issue of location-dependent channel error and shielding short-term error bursts from packet flows. However, the state-of-the-art wireless fair scheduling typically assumes a single, fixed transmission rate for *all* users and normalizes the fair share among users in terms of throughput. It does not anticipate multiple rate options adopted by various users in practice where the channel quality of wireless hosts can vary significantly, both for mobile and stationary nodes [15]. As a result, fair scheduling algorithms designed for single-rate environment, oblivious to rate diversity among users, suffer from significant throughput reduction in 802.11a/b/g/n WLANs.

In this paper, we propose AWFS, a wireless fair scheduler that adapts wireless packet scheduling to the multirate WLANs based on 802.11b/a/g. AWFS exploits the multirate physical-layer capability and supports both data and multimedia applications. It renovates fairness and compensation in virtual *temporal* shares (i.e., temporal shares in virtual time) and departs from throughput-based fairness and compensation. In AWFS, each backlogged flow will receive a fair share in terms of transmission time slices. As a result, AWFS may dramatically improve system throughput by multiplexing better channel conditions across users, thus leveraging the rate heterogeneities among wireless hosts. At the same time, all flows will achieve rational time shares in proportion to their weights governed by the QoS requirements of applications. In summary, AWFS concerns the rate diversity among users and offers *virtual* temporal fairness model to handle channel error and exploit channel dynamics. Thus, it is able to opportunistically utilize high quality channels via transmitting packets in proportion to their high data rates. Through both analysis and simulations, we confirm the effectiveness of AWFS design. The results show that AWFS is able to improve overall throughput up to 159% over the current single-rate scheduling algorithm and individual throughput by up to 550% in simulated scenarios (error, mobility, etc.).

The paper is organized as follows. Section 2 introduces background and identifies the limitations of current fair scheduling algorithms in the multi-rate scenario. Section 3 describes the design, performance analysis, and implementation of AWFS. Section 4 evaluates AWFS through simulations. Section 5 compares it with the related work, and Section 6 concludes the paper.

II. Background

A. Network Model

We consider a wireless LAN based on 802.11b/a/g, operating in the infrastructure mode. Each access point (AP) coordinates all packet transmissions for its hosts. A single channel is shared for both uplink (from a host to an AP) and downlink (from an AP to a host) flows, and for both data and signaling. Every host in a cell can communicate with the AP, though it is not required for any two hosts to be within range of each other. Each flow of packets is identified by a "host, uplink/downlink flag, protocol-id" triple.The underlying MAC protocol follows the 802.11b/e standard [1]. Even though all the hosts and the AP share the same channel, channel errors are location-dependent due to channel fading, interferences, etc. In our design, we do not make any assumptions about the exact error model, though we use two-state Markov chain to generate error patterns in simulations.

B. Motivation

The unique characteristic of location-dependent transmission rate at each receiver renders a wireless fair queuing model, proposed for single-rate scenario, inapplicable. In fact, we find problems with both the fair sharing model among competing flows and the compensation model for error-prone flows.

In wireless fair scheduling, packet flows in the presence of channel errors seek to approximate the services they should receive under idealistic, error-free channel conditions. To this end, error-prone packet flows temporarily defer their transmissions and let error-free flows transmit in advance. This way, channel throughput can be greatly improved because only flows that perceive clean channels are granted transmissions at any given time. Compared with their error-free services (where all flows perceive clean channels all the time), error-prone flows may temporally lag behind and error-free flows lead ahead. However, these leading flows have to give up their future transmissions in order to *compensate* for the lagging flows (that deferred earlier) when the lagging flows perceive clean channels. This way, both leading and lagging flows still receive their contracted rates over longer term, and QoS in terms of throughput, delay and fairness, is still preserved over a larger time scale. In the design, both fair sharing among competing flows and compensation for error-prone flows are defined with respect to throughput. Roughly speaking, each backlogged flow will receive a fair share of throughput (defined in bytes/second) in its error-free service and the compensation in the presence of channel errors is also performed based on throughput. If a flow f's assigned weight is r_f, then flow f receives services during $[t, t+\Delta T]$ in proportion to its weight r_f, given by $S_f(t, t+\Delta T) \approx \frac{r_f}{\sum_{i\in B} r_i} C\Delta T$, where C denotes the channel capacity that is the same perceived by *all* flows, and B represents the set of backlogged flows. While such models work fine in the single-rate scenario, it does not work in multi-rate wireless networks. There are several reasons contributing this. First, each flow perceives different rate, and there is no single system-wide C anymore. Second, normalizing flow throughput in a multirate network will lead to significant inefficiency and mitigate the gains offered by the multirate physical layer, because poor-channel flows will consume disproportionately more time and channel resources. Last, the proposed models cannot leverage the channel dynamics across users, leading to severe throughput degradation. In all, the fundamental problem is, if each host may adopt a different transmission rate, the fairness and compensation models have to be adapted to accommodate such multirate options.

III. AWFS: Adaptive Wireless Fair Scheduling

In this section, we describe the design and analysis of AWFS. AWFS departs from the throughput-oriented fairness and compensation models, originally proposed for the single-rate scenario, and adopts fairness and compensation in temporal shares, defined in virtual time slots. Hence, our proposed design ensures fairness while improving overall channel throughput in the presence of channel errors and leveraging channel dynamics. AWFS has three main components:

- Error-Free Service Model, which defines the ideal fair service for flows that transmit at different rates.
- Lead and Lag Model, which determines which flows are leading or lagging their error free service, and by how much.
- Compensation Model, which compensates for lagging flows at the expense of leading flows, thus addressing the issue of location-dependent channel error.

The specific algorithm described here is by adapting the algorithm of [8]. However, similar adaptations can be readily performed to transform other algorithms [3], [9], [12], [13].

An additional benefit of AWFS is its backward compatibility with WFS [8] in the single-rate scenario. If all flows perceive identical transmission rates, then AWFS degenerates to WFS.

A. Error-Free Service Model

We now describe the algorithm that achieves error-free service for each flow. Our proposed model is adapted from the popular Start-time Fair Queueing (SFQ) [10] algorithm. In SFQ, each arriving packet is assigned two tags: a start tag and a finish tag. Specifically, a packet with sequence number k of flow f arriving at time $A(t_k^f)$ is assigned two tags: a start tag S_k^f and a finish tag F_k^f, defined as follows:

$$S_k^f = \max\{V(A(t_k^f)), F_{k-1}^f\}; \quad F_k^f = S_k^f + L_p/r_f \qquad (1)$$

where L_p denotes the packet size in bits, and $V(\cdot)$ is the system virtual time, taken to be the start tag of the packet currently being served in the scheduler. Then, SFQ selects the flow with the minimum service tag (i.e., the start tag) and transmits its head-of-line packet.

Now we adapt SFQ to the multirate WLAN environment. For flow f, let its transmission rate at t be $C_f(t)$[1], then its tagging is modified as:

$$S_k^f = \max\{V(A(t_k^f)), F_{k-1}^f\};$$
$$F_k^f = S_k^f + L_p/(r_f \cdot C_f(t)) \tag{2}$$

Therefore, the finish tag, as well as the start tag of a packet, is normalized with respect to its current transmission rate. This is to allow for a high-rate flow (that perceives good channel quality and transmits at higher rate) to receive service in proportion to its current rate. Equivalently, the service seeks to provide *temporal* fairness, defined in virtual times, for each backlogged flow only but not for temporally idle flows. It also allows for variable packet size. Once the tags are assigned to each packet, the scheduling decision is still to let the flow f_{min} with the *smallest* start tag transmit first.

B. Lead/Lag Model

The lead and lag model specifies how much compensation is needed for a lagging flow at the cost of a leading flow. In AWFS, this is defined in terms of virtual time units to ensure fair temporal compensation. Each flow f has a credit counter $E_f(t)$ to tag the current flow status, e.g. $E_f(t) > 0$ indicates f is leading flow. It is designed based on two principles. (1) Only when the virtual time units given up by a lagging flow g are used by another leading flow l, we update the flow's credit counter. (2) The credit counter is in virtual time, i.e., transmitted bytes normalized with respect to each flow's transmission rate. That is, $E_l(t) = E_l(t) + L_g/(r_l C_l(t))$, and $E_g(t) = E_g(t) - L_g/(r_l C_l(t))$, where L_p is the packet size being transmitted from the leading flow, and $C_l(t)$ is the current transmission rate of the leading flow.

In essence, the lead and lag model specifies the temporal share (i.e., how many virtual time units) each leading flow has to give up in the future, and how many virtual time units a lagging flow will receive compensation.

C. Compensation Model

We further address two issues: (a) How does a leading backlogged flow decide whether to transmit its data packet or relinquish the current scheduling opportunity for compensation? and (b) Which among several lagging backlogged flows gets to transmit in case that has been relinquished by a leading flow? We now consider each question in turn.

We adapt the graceful service relinquishing model for leading flows used by WFS. Consider a leading flow i with a lead of $E(i)$, a rate r_i, and a maximum lead of $E_{\max}(i)$. Flow i hierarchically decomposes itself into two flows, i^c and i^t, with rates of $r_i E(i)/E_{\max}(i)$ and $r_i(1 - E(i)/E_{\max}(i))$. Flow i^c is designated to be the compensation flow, while i^t is designated to be the transmission flow. When flow i is allocated a transmission, it hierarchically schedules it among flows i^c and i^t. All time units belonging to flow i^c are relinquished. Note that as the lead decreases, the rate for flow i^c decreases linearly. This has the property of graceful degradation of service for a leading flow.

Once a leading flow gives up the transmission of its head-of-line packet, we need to select a lagging flow for transmission in the given time slice. If the packet size is L_l, and the transmission rate of the leading flow is $C_l(t)$, then the available time for compensation is $\Delta T_l = \frac{L_l}{C_l(t)}$. In the multirate scenario, how to select a lagging flow to fill in ΔT_l poses new challenges. In the single-rate scenario, WFS assumes fixed slot and identical packet size for all flows. Therefore, any lagging flow, once selected, can be transmitted in the relinquished slot by the leading flow. In the multirate case, this is not true anymore. In fact, the transmission of the head-of-line packet from the selected lagging flow may fill a portion of ΔT_l, or may far exceed the length of ΔT_l. Moreover, the simple weighted round robin (WRR) based compensation used in WFS is not applicable because the weights may not be even integers.

We devise a new solution to the compensation model, which still preserves the feature of graceful compensation among lagging flows but works with any transmission rate and arbitrary packet size. This is achieved by adopting a special SFQ for the compensation process. Given the credit counter of a lagging flow i as $G_i(t)$, we normalize it with respect to all lagging flows $g_i(t) = \frac{G_i(t)}{\sum_{j \in F(t)} G_j(t)}$, where $F(t)$ denotes all lagging flows at t. Then, each lagging flow receives compensation (in terms of time slices) in proportion to $g_i(t)$. This can be realized via the following operations. The head-of-line packet from lagging flow g is assigned a compensation tag $\frac{L_g}{C_g(t)} \cdot \frac{1}{g_g(t)}$, given that L_g is the head-of-the-line packet size, $C_g(t)$ is the current transmission rate. Then, the lagging flow with the smallest compensation tag is selected to receive the compensation time slice. In short, the compensation model seeks to allocate compensation time slices fairly among lagging flows.

D. Performance Analysis

a) Throughput bound: We can establish the throughput bound for AWFS in the real-time domain. The proofs are by adapting the argument of SFQ [10]; we omit them due to lack of space.

Theorem 3.1: (Per-Flow Throughput Bound) If a flow i is continually backlogged over a real-time interval $[t_1, t_2]$, then its aggregate service (in bits) $W_i(t_1, t_2)$ is bounded by

$$W_i(t_1, t_2) \geq r_i \int_{t_1}^{t_2} C_i(t)dt - kL_{\max} \tag{3}$$

where k is a constant, and r_i is the normalized weight factor.

Theorem 3.2: (Throughput Improvement) Consider the same set of backlogged flows, and each flow i uses transmission rate C_i, then the overall channel throughput gain achieved by AWFS over WFS during steady state is approximately given by $\sum_i r_i C_i \cdot \sum_i \frac{r_i}{C_i}$, where weights satisfy $\sum_i r_i = 1$.

Corollary 3.1: The overall throughput of AWFS is always greater than WFS in the presence of multiple rates. That is, $\sum_i r_i C_i \cdot \sum_i \frac{r_i}{C_i} > 1$ if at least two rates are different.

[1]This rate can be normalized with respect to the base rate.

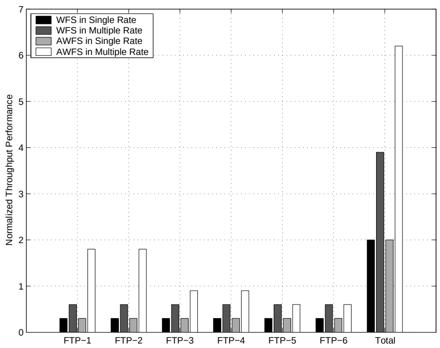

Fig. 1. Throughput Gain of AWFS over WFS

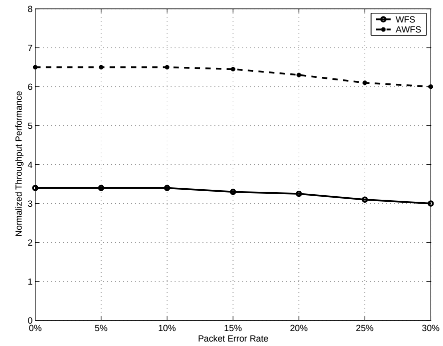

Fig. 2. WFS vs. AWFS Throughput

b) Fairness: The following fairness holds:

Theorem 3.3: (Long-term fairness index) Consider all flows are backlogged. For a continually backlogged flow i, it achieves the following long-term proportional fairness for its throughput $S_i(0, v)$ during virtual time interval [0,v]:

$$\lim_{v \to \infty} \frac{S_i(0, v)}{v} = r_i C_i. \tag{4}$$

E. Implementation

We address several implementation issues of AWFS within the current 802.11 MAC framework.

- *Channel state estimation and propagation*: Since each receiver has the most accurate information on packet transmissions, we estimate channel state (i.e., clean or dirty) at each receiver's side. Therefore, each mobile host estimates the channel state. For a host which is not the intended receiver at the moment, we may use the polling procedure provide in current 802.11b/e MAC design to estimate channel states. Once estimated, each mobile host will pass this channel state information back to the AP through piggybacked QoS Data or QoS Null packet.
- *Handling uplink flows*: Our design works for both downlink and uplink flows. For uplink flows, the AP will periodically poll each host to collect the following information: the size of the HoL packet, and the arrival time of the HoL packet if the flow becomes backlogged again after an idle period.
- *Transmission rate at each host*: The AP will collect the transmission rate at each host. The mechanism to obtain current transmission rate can follow ARF [5] in the current 802.11.

IV. Simulation Evaluation

We now present simulation results to evaluate AWFS in various scenarios. We compare its performance with WFS [8]. Four type of traffics are considered in the simulations, i.e., FTP, CBR, Poisson and Markov-modulated Poisson Process (MMPP) sources. Packet size for each flow may vary. We use one-step prediction [12] to estimate the immediate future channel based on the current channel state (i.e., clean/dirty). This exploits the feature that channel errors are highly correlated over short time. Each simulation run lasts for 100000 units unless otherwise explicitly stated, and the results are averaged over 50 simulation runs.

A. Throughput gain in error-free channel

We consider six FTP flows in the error-free scenario to show the throughput improvement of AWFS over WFS. The comparison base is that each flow uses 2 Mbps transmission rate. In the multi-rate scenario, flows 1 and 2 transmit at 11 Mbps, flows 3 and 4 use 5.5 Mbps, and flows 5 and 6 still use 2 Mbps. We also vary the packet size of each flow in simulation runs. Figure 1 shows the per-flow throughput, as well as the overall throughput.

The figure shows that, AWFS achieves aggregate throughput 159% of WFS in the multi-rate scenario, while achieving throughput identical to WFS if all flows use the same rate. Significant throughput gain is also achieved on a per-flow basis, particularly for flows that use higher transmission rates. Throughput increases by 550% for flows 1 and 2, and increases by 270% for flows 3 and 4. Compared with the base case, WFS only increases 94% for each flow.

B. Throughput and fairness in error-prone channel

In this set of experiments, we study the effectiveness of the compensation model of AWFS in the presence of channel errors. The popular two-state discrete Markov Chain is employed to simulate channel errors. Four FTP source are used, and two flows (FTP-3,4) use base transmission rate 2.0 Mbps and the other two (FTP-1,2) transmit at 11 Mbps.

The throughput results for WFS and AWFS are depicted in Figure 2, where channel error varies from 0% up to 30%. We observe that as channel error increases to 20%, the throughput for both algorithms begins to suffer moderately because the probability that all flows are simultaneously error-prone increases. However, the overall throughput of AWFS remains approximately 87.5% to 92% higher than WFS.

To study the temporal fairness and effectiveness of compensation model, we record the normalized time share acquired by each flow in Table I. When the channel is clean, each flow obtains an equal temporal share 1 unit. Note that as the channel

Error Rate	FTP-1	FTP-2	FTP-3	FTP-4
0%	1.0000	1.0000	1.0000	1.0000
2%	0.9991	0.9991	1.0015	1.0005
5%	0.9982	0.9980	1.0019	1.0019
10%	0.9972	0.9979	1.0056	0.9977
15%	0.9915	0.9920	1.0094	0.9928
20%	0.9770	0.9779	1.0110	0.9940
25%	0.9513	0.9518	0.9604	0.9522
30%	0.9261	0.9254	0.9456	0.9294

TABLE I

NORMALIZED TEMPORAL SHARE STATISTICS

error increases, the time units obtained by flows decrease, and this subsequently leads to throughput reduction, shown in Figure 2. However, the long-term temporal shares of all flows are still roughly preserved, thus showing that the compensation model is working. Even when channel error rate increases to 20%, the system throughput and time share gained by each flow only reduce slightly. This demonstrates that AWFS is still able to shield errors from flows and retain good overall throughput.

C. Packet delay in error-prone channel

We study the impact of error-prone channel on packet delay. Flows 1 and 2 are MMPP sources with packets arriving at the rate 1.0, and flows 3 and 4 are Poisson sources with packet arrival rate of 0.5. Two CBR flows with rate 1.0 are to emulate the background traffic. The transmission rates for MMPP-1, Poisson-1 and CBR-1 are set as 2 Mbps, and the other three sources use 11 Mbps. The error patterns are the same as in Section 4.2.

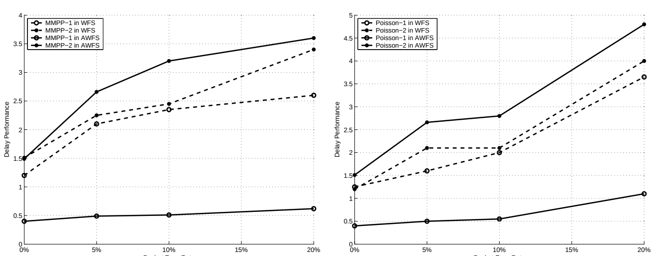

Fig. 3. Packet delay for MMPP and Poisson flows

The packet delay for MMPP and Poisson flows is depicted in Figure 3. It shows that, the delay experienced by each flow increases as channel error becomes severe. Because WFS ensures throughput fairness, the delays experienced by high-rate flows and low-rate flows are approximately the same. However, delay is different for low- and high-rate flows in AWFS. High-rate flows, MMPP-2 and Poisson-2, experience noticeably less delay than those low-rate flows. However, low-rate flows still have delay performance comparable to WFS. This shows that AWFS is able to provide certain degree of flow separation among high-rate and low-rate flows, such that high-rate flows will not be penalized or even paralyzed by low-rate flows.

D. Throughput Gain over Changing Channel Conditions

In the above simulations, we let each flow transmit at the same rate all the time. In reality, the transmission rate of a flow is varying dependent on the changing channel condition; this can be due to channel fading, obstacles, host mobility, and etc. To this end, we adopt the Rayleigh fading model and threshold rate adaptation mechanisms to emulate the effect of varying channel quality. The Rayleigh fading is base on the well-known Jakes simulator. The variation of wireless signal is induced at a rate that depends, in part, on the speed along the line-of-sight between the AP and the mobile station. The station's mobility further affects the average channel coherence time, and higher velocity will result in smaller channel coherence time. To observe the impact of dynamic channel conditions, we conduct the experiments in IEEE 802.11b, which allow four transmission rates in Rayleigh fading channels. We consider six flows, where the mobile station travels back and forth from the AP with different velocities in an oscillatory fashion as described in *Configuration 1* of [6]. The simulation time is set long enough to ensure the average time spent at each distance is independent of the station velocity, and thus only the speed of channel variation is the deciding factor.

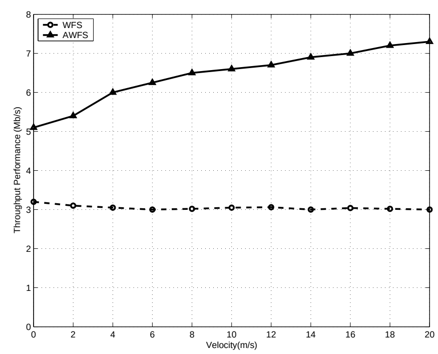

Fig. 4. Throughput over Changing Channel Conditions

The throughput for AWFS and WFS is depicted in Figure 4 for speed up to 20m/s (72km/hr). WFS has a throughput that is nearly independent of velocity, while AWFS improves the overall throughput by 22.8% with the mobility speed increasing from 1m/s to 20m/s. Moreover, AWFS remains appropriately 61% to 120% greater throughput than WFS. The key reason is that AWFS is able to opportunistically utilize the good channel quality even within small coherence time because both the S_f and F_f are updated based on the current channel condition of each flow. AWFS implicitly favors the flow with highest transmission rate since such a flow tends to have the smallest start tag according to tagging method (2).

V. RELATED WORK

Packet scheduling has been a very popular paradigm to provide packet-level quality of services for packet flows. Numerous algorithms have been proposed, such as WFQ [11] and SFQ [10]. In recent years, much research effort has been made to adapt fair packet scheduling to cellular wireless networks, notably IWFQ [12], CIF-Q [9], SBFA [13], and WFS [8]. The goal of these wireless fair scheduling algorithms has been to hide short bursts of location-dependent channel errors from well-behaved flows by dynamically swapping channel allocations between backlogged flows that perceive channel errors and backlogged flows that do not. All the proposed fairness models are throughput based, and will suffer in the multirate scenario.

There have been several recent efforts on new MAC designs to exploit the multirate physical-layer capability. In Auto Rate Fallback (ARF) [5], senders seek to use higher transmission rates after consecutive transmission successes (that indicate high channel quality) and revert to lower rates after failures. In Receiver Based Auto Rate (BRAR) [6], receivers set the transmission rate for each packet according to the highest feasible value allowed by the channel condition, which is measured by physical-layer analysis of the RTS message at the receiver. In Opportunistic Auto Rate (OAR) [7] protocol, the sender opportunistically transmits multiple back-to-back data packets whenever the channel quality is good. It also seeks to provide each host same time-shares as achieved by the single-rate IEEE 802.11. AWFS is fundamentally different from recent MAC-layer solutions that exploit multirate capability [6], [7]. These proposals modify the current 802.11 MAC, whereas AWFS works with the existing MAC. Though [7] ensures *fine-grain* temporal fairness at MAC layer, it works with fixed frame size, assumes that flows are always backlogged, and targets the *ad-hoc* mode. AWFS is a packet-level solution that ensures *virtual* temporal share, works with variable packet size and occasionally idle flows, and operates in the infrastructure mode. In fact, the packet-level and MAC-layer solution should work in concert to both improve the channel efficiency and satisfy QoS requirements of various applications. In addition, AWFS differs the previous work [16] by introducing the *virtual* temporal fairness concept and the fortified component models. Theoretical performance analysis and implementation issues are also supplemented.

VI. Conclusion

The state-of-the-art 802.11 WLAN technology offers a multi-rate physical-layer feature, opening the door for significant throughput improvement via such adaptive modulation techniques. However, the current wireless fair scheduling design is unable to take advantage of such features. In this paper, we propose AWFS, which is able to significantly improve both aggregate and per-flow throughput while still preserving a notion of fair sharing defined in virtual temporal shares. Thus, AWFS enables flows to opportunistically exploit the good channel quality to transmit packets at their high rates. It also works with variable packet size, and allows for bursty transmission by flows that switch between idle and backlogged modes over time. AWFS can effectively operates with the current 802.11b/e MAC design or other improved MAC scheme [7] [6], and can be implemented as scheduler inside Access Point or Hybrid Coordinator to satisfy the QoS requirements posed by various applications. Both simulations and analysis show significant performance gains, thus enabling communication-intensive multimedia and data applications in the 802.11 WLAN. Overall, AWFS offers an effective solution that provides packet-level quality of service, in terms of throughput, delay, and fairness, to diverse applications.

References

[1] B. O'Hara and A. Petrick. IEEE 802.11 Handbook, A Designer's Companion. IEEE Press, 1999.

[2] IEEE 802.11e/D5.0, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS).

[3] T. Nandagopal, S. Lu, and V. Bharghavan. A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms. In ACM MOBICOM'99.

[4] G. Bianchi. Performance analysis of the IEEE 802.11 media access protocol for wireless LANs. In ACM SIGCOMM'94.

[5] A. Kamerman and L. Monteban. WaveLAN II: A high-performance wireless LAN for the unlicensed band. Bell Labs Technical Journal, pages 118-133, Summer 1997.

[6] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In ACM MOBICOM'01.

[7] B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly. OAR: A Multi-rate Media Access Protocol for Wireless Ad Hoc Networks. ACM Mobile Networking and Applications, 2003.

[8] S. Lu, T. Nandagopal and V. Bharghavan. A Wireless Fair Service Algorithm for Packet Cellular Networks. In ACM MOBICOM'98.

[9] T. Ng, I. Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location dependent errors. In IEEE INFOCOM'98.

[10] Goyal, P., Vin, H. M., and Cheng H. Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks. IEEE/ACM Transactions on Networking, October 1997.

[11] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing alorithm. In ACM SIGCOMM, September 1989.

[12] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. IEEE/ACM Transactions on Networking, Aug. 1999.

[13] Parameswaran Ramanathan , Prathima Agrawal. Adapting packet fair queueing algorithms to wireless networks. In ACM MOBICOM'98.

[14] A. Parekh. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks. PhD Thesis, MIT Laboratory for Information and Decision Systems. 1992.

[15] T. S. Rappaport. Wireless Communications: Principles and Practice. Prentice Hall, 1999.

[16] Yuan Yuan, Daqing Gu, William Arbaugh, Jinyun Zhang, "Achieving packet-level quality of service through scheduling in multirate WLANs", IEEE VTC 2004

[17] N. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a wireless LAN. In ACM MOBICOM'00.