AQM for Weighted Fairness in Wireless LANs

Do-hyung Kim, Jong-hwan Kim, Hyun-soo Yoon and Ikjun Yeom *Member, IEEE*

***Abstract*—It is not possible to achieve fairness in IEEE 802.11-based WLANs using only previous fair medium access schemes, because of the impact of higher-layer features such as TCP dynamics and AP congestion. To overcome the limitations of MAC layer approaches, we propose a very simple but effective queue management scheme, which can be applied to the AP. The simulation result shows that the proposed scheme efficiently achieves weighted fairness and can be used to preserve the effect of MAC layer scheduling strategies.**

I. Introduction

In the past, the fairness issue in IEEE 802.11 WLANs is considered as the problem how to provide an equal access probability to each node. As the amount of traffic increases, however, it turns out that congestion losses at an Access Point (AP) impact on fairness, and the equal access probability itself does not guarantee the fair throughput to each individual flows [1]. In TCP flows, especially, since they respond differently to data packet losses and ACK losses, uplink flows observing ACK losses at an AP achieve higher throughput than downlink flows observing data packet losses at the same AP.

To deal with this unfairness problem due to congestion losses at an AP, several queue management schemes have been suggested in [2]–[4]. In [2], separate queues are used to enable data packets and ACK packets to be handled differently. This scheme effectively balances uplink/downlink traffic, but cannot achieve per-flow fairness. In [3] and [4], fairness among TCP flows is provided by per-flow state management. However, the schemes require a high complexity to manage the per-flow state for the arrival of every packet, and even a modification of the TCP protocol stack. Considering that an AP is a cheap and light-weight type of equipment, the complexity and overhead of these schemes could be a critical impediment to their implementation.

In addition to the max-min fairness problem, the weighted fairness problem in IEEE 802.11 WLANs also needs to be considered. Many scheduling policies at the MAC layer have been studied for service differentiation, whereupon various weighted medium access schemes such as the IEEE 802.11e standard have been introduced. However, the effect of weighted medium access schemes at the MAC layer is dissipated when flows pass through the AP. Nevertheless, there has been no previous research on this weighted fairness problem.

In this letter, we introduce a simple but effective queue management scheme, which can be applied to the AP for supporting weighted fairness in IEEE 802.11 WLANs. The proposed scheme is based on stateless AQM technology such as CHOKe [5] and E-AQM [6], which had been originally devised to achieve max-min fairness in wired networks. In spite of the low overhead, CHOKe and E-AQM cannot be directly applied to wireless networks because: 1) they cannot support weighted fairness, which is an important requirement in IEEE 802.11 WLANs; 2) they do not work properly when a few dominant flows (traffic intensive flows) occupy the whole bandwidth, which is a matter of frequent occurrence in wireless networks; and 3) they degrade the performance of wireless networks because they waste expensive channel resources by discarding packets. The proposed scheme overcomes these limitations by introducing novel cache management methods and exploiting ECN bit. Via ns-2 simulations, it is verified that the proposed scheme efficiently implements weighted fairness.

II. The proposed scheme

A. Basic operation

The proposed scheme is equipped with a fixed-size cache containing flow identifiers (IDs) of successfully relayed packets. When a new packet arrives at the AP, it is compared with one randomly selected from the cache. If the packets match, then the arrived packet is discarded in order to penalize high bandwidth flows. This comparison operation is performed with a probability p_d in order to realize the desired drop rate of packets, p, which is intended for a stable queue size in order to prevent an empty queue and burst packet drop. For the determination of p, any prior active queue management scheme can be used. Once p is determined, p_d is set to $\frac{p}{p_m}$ based on (1).

$$p = p_d p_m \tag{1}$$

where p_m is the probability that the compared packet is successfully matched with the one selected from the cache.

Whenever a packet is admitted into the queue, an additional comparison between this successfully enqueued packet and the one randomly selected from the cache is performed with a probability, p_u, for updating the cache. If the packets match, the oldest ID in the cache is replaced by the ID that has been compared. This cache update process causes an increase in the portion of unfair high throughput flows in the cache and consequently those flows suffer from high drop probability.

If we let τ_i be the obtained throughput(enqueuing rate) of flow i and c_i as the proportion of flow i in the cache, the cache update rate for flow i can be represented by $\tau_i p_u c_i$. In each comparison for updating a cache, the matching probability of flow i becomes $\frac{\tau_i c_i}{\sum_{j=1}^{N} \tau_j c_j}$ when N number of flows exist. And using the probability c_i, the ID of flow i can be replaced as the oldest one in the cache. Therefore, the variation of c_i, Δc_i, on each cache update is as follows

$$\Delta c_i = \frac{\tau_i c_i}{\sum_{j=1}^{N} \tau_j c_j} - c_i \tag{2}$$

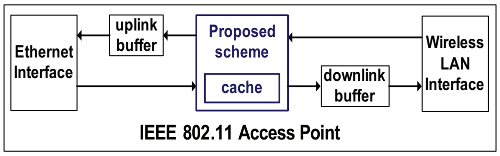


Fig. 1. Implementation of AQM

In [6], it was proved that 1)the cache converges to the equilibrium state, $\Delta c_i = 0$ and 2)max-min fairness is achieved in the equilibrium state, thus for all flows, the achieved throughput becomes $\tau_i = \sum_{j=1}^{N} \tau_j c_j$.

The basic operation effectively achieves max-min fairness in wired networks. However, in wireless networks where a few dominant flows can occupy the whole bandwidth of the wireless link, the basic operation malfunctions due to excessive cache occupation by the dominant flows. Besides, discarding packets in wireless networks wastes expensive wireless channel resources, which leads to the degradation of throughput performance. More seriously, the basic operation results in max-min fair throughput even when weighted access schemes are used at the MAC layer for service differentiation. In the following section, we will present how the proposed scheme handles these limitations and supports weighted fairness in IEEE 802.11 WLANs.

B. Application to IEEE 802.11 WLANs

Because the IEEE 802.11 WLAN is a half-duplex system, we control both uplink and downlink traffic with a common cache, as shown in Fig. 1. When data packets arrive at the AP from any direction, the proposed scheme inserts them into the queue or discards them through comparison with the ID in the cache. Here we note that TCP ACK packets and other control packets are ignored during the process and are always inserted into the queue unless it is full.

C. Generalization for weighted fairness

For weighted fairness, the flow with a larger weight should observe a lower discarding rate. The differentiation of the discarding rate can be manipulated by using different per-flow cache update probabilities. For max-min fairness, the cache update process is performed by p_u for every packet admitted into the queue. For weighted fairness, we introduced the per-flow cache update probability, u_i, instead of p_u. Then, Δc_i in (2) is rewritten as follows

$$\Delta c_i = \frac{\tau_i u_i c_i}{\sum_{j=1}^{N} \tau_j u_j c_j} - c_i \tag{3}$$

In an equilibrium state, $\Delta c_i = 0$, (3) implies that $\tau_i u_i = \sum_{j=1}^{N} \tau_j u_j c_j$ and $\tau_i u_i$ has the same value for all i. Therefore, τ_i can be differentiated by controlling u_i. For example, the proposed scheme doubles the throughput of flow i by halving the cache update probability of flow i. By applying different u_i for the traffic of each priority class, the proposed scheme can realize weighted fairness simply and effectively.

This strategy is also utilized to handle different-sized packets. The flow with a larger packet size should observe a higher discarding rate for fair throughput. Therefore, u_i, which should be proportional to the packet size, is set by $\frac{k_i}{1000}$, where k_i is the packet size of flow i.

D. Dealing with traffic intensive flows

Unlike in large-bandwidth wired networks, a single UDP flow can occupy a large portion of the whole bandwidth in IEEE 802.11 WLANs. Then, the majority of the cache is occupied by the ID of this traffic intensive UDP flow for penalization of unfair high bandwidth. The problem is that the remainder of the cache is too small to reflect the traffic log of the other flows, which makes the proposed scheme dysfunctional. To solve this problem, the portion of normal TCP flows in the cache should increase while maintaining the same discarding rate. For this, we introduce different comparison probabilities for TCP packets and UDP packets respectively, instead of using the same p_d.

The discarding rate of flow i, p_i, is represented by (4).

$$p_i = \lambda_i p_d c_i = \lambda_i (\frac{p_d}{n})(n c_i) \tag{4}$$

where λ_i is the sending rate of flow i and $n \geq 1$.

To increase the portion of TCP flows in the cache, the proposed scheme uses a smaller comparison probability for TCP packets. Even if the comparison probability changes from p_d into $\frac{p_d}{n}$, fair throughput is still guaranteed by the convergence of Δc_i in (3). The maintenance of fair throughput implies that p_i remains constant. According to (4), the portion of flow i in the cache converges from c_i to nc_i. TCP congestion control mechanisms regulate the sending rate of a TCP flow, which generally results in a packet drop of 0.1$\sim$10%. On the other hand, a UDP flow can observe a packet drop of 0$\sim$99%. Because $\frac{p_i}{\lambda_i}$ in (4) represents the ratio of dropped packets to sending packets of flow i, the portion of a UDP flow in the cache can be more than ten times larger than that of a TCP flow under the same p_d. Based on this observation, 10-20 is an appropriate value of n for TCP flows for equal per-flow occupation in the cache.

TCP flows belonging to different service classes have the same problem, because the flow with a smaller weight must occupy a larger portion of a cache in order to realize a lower throughput. Based on the observation that the throughput is inversely proportional to the $\sqrt{p_i}$ in TCP, the comparison probability for the flow with a desired throughput that is α times larger should be $\frac{p_d}{\alpha^2}$ for equal per-flow occupation in the cache.

E. Minimizing waste of channel resources

Uplink packets arrive at the AP after consuming the wireless channel resources of a bottleneck link. Therefore, discarding uplink packets is simply a waste of channel resources and cannot increase the throughput of other contending flows. To avoid this waste of channel resource, the proposed scheme uses ECN bit for uplink TCP packets instead of discarding them. Here we note that the proposed scheme does not regulate uplink UDP flows for the same reason.

III. SIMULATION

The proposed scheme is evaluated using the ns-2 simulator and compared with the IEEE 802.11 standard, CHOKe, E-AQM and the latest research (ACC-AF) [3] for fairness in IEEE 802.11. The simulation topology is shown in Fig. 2. The

AP queue can contain a maximum of 100 packets. The packet size is 1KB and the value of n for securing the portion of TCP flows in the cache is set to 20. The parameters for E-AQM are set as follows: $q_t = 50$, $d = 20$, $p_s = 0.005$ and $C = 600$ packets based on the IEEE 802.11b link capacity. The AQM used to determine p and the cache size in the proposed scheme are the same as in E-AQM.

In the first simulation, ten TCP flows belonging to three different Access Classes (ACs) pass through the AP. Five flows generate uplink TCP packets and the other five flows generate downlink TCP packets. For each direction, one flow belongs to AC1 and AC2 respectively and the other three flows belong to AC3. To achieve the 4:2:1 desired throughput ratio among different ACs, we used values of 7, 15 and 31 for the CW size of AC1, AC2 and AC3 respectively, according to a weighted access scheme presented in [7]. Interframe spaces are set to 50us for all ACs in common. The propagation delay of the wired link varies by 5ms for each flow(from 10-55ms).

The throughput result is shown in Fig. 4. Flow 0 and flow 1 in Fig. 4 correspond to AC1 and AC2 respectively. In the standard, there is a big difference in throughput between uplink and downlink flows belonging to the same ACs. And, downlink flows with the lowest priority even observe starvation. In the proposed scheme, however, the access ratio achieved by the scheduling algorithm at the MAC layer is almost preserved. This result indicates that the proposed scheme can enhance the usability of IEEE 802.11e and support other various MAC layer scheduling schemes.

Fig. 3 shows the simulation result where the proposed scheme is applied without any additional weighted access scheme at the MAC layer. The result shows that the proposed scheme can provide weighted fairness on its own, even without any support at the MAC layer. Considering the difficulty of MAC layer modification, the proposed scheme is a good alternative option to achieve weighted fairness in IEEE 802.11 WLANs.

The second simulation is performed with TCP and UDP

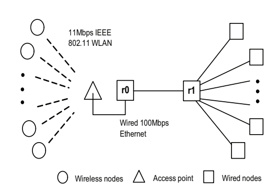


Fig. 2. Simulation topology

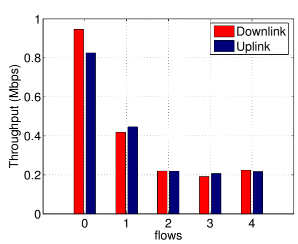
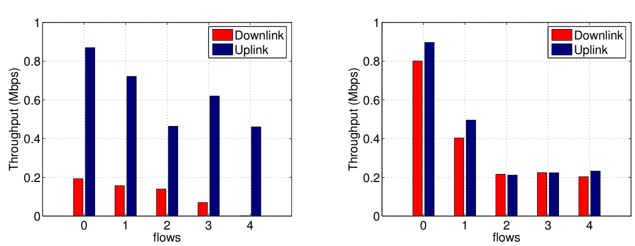


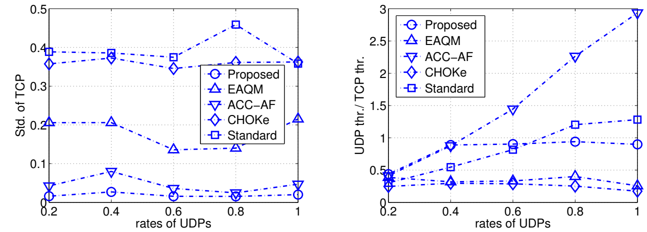
Fig. 3. Achieving weighted access in IEEE 802.11 WLAN



(a) IEEE 802.11e

(b) The proposed scheme

Fig. 4. Preserving weighted access in IEEE 802.11 WLAN



(a) standard dev. of TCP flows

(b) UDP thr./TCP thr.

Fig. 5. Performance comparison

flows. Two UDP flows and three TCP flows generate downlink traffic, and the other five TCP flows generate uplink traffic in the same topology as the first simulation. The sending rate of UDP flows ranges from 0.2-1Mbps. In Fig. 5(a), the standard deviations of TCPs' throughput in the proposed scheme are very close to 0, which indicates that the proposed scheme achieves throughput fairness among TCP flows without regard to the sending rate of UDP flows. In Fig. 5(b), the ratio of averaged throughput between TCP and UDP flows is shown in order to verify how the proposed scheme handles traffic intensive UDP flows. Unlike the other schemes, the proposed scheme shows a value close to 1 for traffic intensive UDP flows ($\geq$ 0.4Mbps). When the sending rate of UDP flows is 0.2Mbps, the proposed scheme preserves UDP throughput almost perfectly (0.199Mbps) while achieving a TCP throughput of 0.457Mbps.

The averaged overall throughput of the proposed scheme is 0.41Mbps, which is 10% lower than that of the 802.11 standard, 0.45Mbps. TCP flows require more access to the bottleneck wireless link than UDP flows to complete packet delivery because of ACK mechanism. Therefore, the increased TCP throughput in the proposed scheme for achieving fairness naturally results in overall throughput degradation.

IV. Conclusion

In this letter, we proposed a simple and light-weighted queue management scheme to realize weighted fairness in the IEEE 802.11 WLAN environments. The performance evaluation shows that the proposed scheme preserves the effect of the scheduling schemes in the MAC layer effectively, and even achieves weighted fairness on its own using only limited resources.

References

[1] S. Pilosof *et al.*, "Understanding TCP fairness over wireless LAN," in *IEEE INFOCOM*, Apr. 2003.

[2] Q. Xia *et al.*, "Active queue management with dual virtual proportional integral queues for TCP uplink/downlink fairness in infrastructure WLANs," *IEEE trans. on Wireless Comm.*, vol. 7, pp. 2261 – 2271, Jun. 2008.

[3] F. Keceli *et al.*, "Achieving fair TCP access in the IEEE 802.11 infrastructure basic service set," in *ICC*, May 2008.

[4] M. Gong *et al.*, "Queue management strategies to improve TCP fairness in IEEE 802.11 wireless LANs," in *WiOpt*, Apr. 2006.

[5] R. Pan *et al.*, "Choke: a stateless active queue management scheme for approximating fair bandwidth allocation," in *IEEE INFOCOM*, Mar. 2000.

[6] J. hwan Kim *et al.*, "Long-term flow fairness with fixed memory space," *IEEE Comm. Letters*, vol. 13, pp. 998 – 1000, Dec. 2009.

[7] D. Qiao and K. G. Shin, "Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF," in *IEEE IWQOS*, 2002.