TCP Fairness for Uplink and Downlink Flows in WLANs

Juho Ha and Chong-Ho Choi
School of Electrical Engineering and Computer Science, and ASRI
Seoul National University, Seoul, Korea.
{hazo,chchoi}@csl.snu.ac.kr

Abstract—In WLANs, fairness is an important issue because the channel is shared by many users. This paper proposes a dual queue based scheme in an access point (AP) for TCP fairness among uplink flows and downlink flows. First, we discuss the unfairness problem of TCP flows due to different responses to data packet drop and TCP ACK packet drop at the AP. The proposed scheme employs two queues, one for the data packets of downlink TCP flows and another for the ACK packets corresponding to uplink TCP flows. The performance of the proposed scheme is evaluated by simulation and the results are presented. The dual queue scheme is simple and effective for resolving the TCP unfairness problem.

I. INTRODUCTION

Nowadays, Wireless LANs (WLANs) based on the IEEE 801.11 standard are widely deployed. As the number of WLAN users has grown rapidly, service fairness among users has become an important issue. The IEEE 802.11 standard defines two MAC techniques: the distributed coordination function (DCF) and the point coordination function (PCF). Currently, most of the WLAN devices employ only DCF due to its simplicity and efficiency. DCF provides fair channel access opportunities among mobile stations.

The most popular network configuration of WLANs is the infrastructure mode, in which all the mobile stations communicate through an access point (AP). WLANs in the infrastructure mode can provide network access at public areas, such as campus, cafes and hotels. While DCF allows all the competing stations equal opportunity for media access, it does not guarantee fair service provision at the transport layer when it is applied to the infrastructure mode. Pilosof et al. [1] pointed out that the interaction between DCF and TCP can cause unfairness among uplink and downlink TCP flows in the infrastructure mode. Because most internet services run over TCP connections, we will restrict our attention to TCP flows.

In this paper, we propose a dual queue based scheme that can solve this unfairness problem. In this scheme, an AP employs two interface queues, one for the data packets of downlink TCP flows and another for the ACK packets corresponding to uplink TCP flows. The AP serves the queues with different probability to control the ratio of TCP data sending rate and TCP ACK sending rate. Through analysis and simulations, we show that TCP fairness for uplink and downlink flows can be achieved by adjusting the queue selection probability. The proposed scheme is simple, easy to deploy and only needs minor changes in AP queueing mechanism.

This paper is organized as follows. In section II, we present some previous works on this problem. We propose our scheme in section III, and its performance is evaluated by simulation in section IV. In section V, we conclude the paper.

II. RELATED WORK

Many MAC protocols have been proposed to achieve fairness at the transport layer in the infrastructure mode WLANs. The researchers showed that fairness at the MAC layer protocol does not guarantee fairness at the transport layer. In the infrastructure mode, an AP shares wireless channel with its mobile stations. When the AP competes with $N$ mobile stations, it obtains only $1/(N+1)$ of transmission opportunities. Because all the downlink flows communicates through the AP, the downlink flows have to share this portion. This behavior at the MAC layer leads to uplink/downlink asymmetry at the transport layer.

The authors of [2], [3] and [4] proposed MAC protocols, which leads to TCP fairness among uplink and downlink flows. In [2], the contention windows of mobile stations are dynamically adjusted. By increasing the contention windows, the AP has more chances to access the wireless channel, and so the fairness among uplink and downlink flows improves. Similarly, by adjusting DIFS in AP, the AP can get higher priority in accessing the channel [3]. In [4], each station defers channel access based on the next packet information that is collected at AP. All these schemes are based on MAC layer scheduling and give the AP more chances to access the channel than the mobile stations. Since these schemes do not consider the interaction between the MAC and TCP protocols, it is difficult to predict the results when they are applied to TCP flows. Moreover the MAC layer of mobile stations has to be modified in [2] and [4].

Pilosof et al. [1] showed that the buffer size of AP plays an important role in unfairness. They presented a simple solution where the AP manipulates the TCP advertised window field of ACK packets. However this scheme is very complex to implement since the AP needs to manipulate TCP headers of all packets.

To provide the fairness, the MAC QoS parameters of TCP data packets and ACK packets for uplink and downlink flows were set to different values [5]. In this scheme, the TCP ACK packets are transmitted with minimal queueing, while the downlink TCP data packets are transmitted by TXOP bursting.
It can be deployed only in the 802.11e networks, and it also needs to modify the MAC protocol of mobile stations.

In [6], per-flow queueing was deployed to alleviate the TCP unfairness. It performs well, but it is very complex to manage per-flow queues for all the uplink and downlink TCP flows. Also Kim [7] proposed a scheme which improves per-station fairness of TCP flows based on channel access time of each mobile station. It works well but requires some computational work in AP.

III. PROPOSED SCHEME

Many researchers found the cause of the uplink and downlink TCP unfairness at the MAC layer or at the interaction between the MAC and the transport layer protocols. The solution for this problem varies depending on how one sees the cause of the problem. As [1] and [6], we consider that the main cause of the unfairness is the packet dropping mechanism at the AP queue. The AP queue has two types of TCP packets. One type is the data packets for downlink TCP flows and another type is the ACK packets corresponding to uplink TCP flows. These two types of packets are buffered at the same queue, but the consequence of a packet drop is quite different. If a data packet for a downlink TCP flow is dropped, the congestion window of the TCP flow is reduced to half by duplicate ACKs. However, if an ACK packet for an uplink flow is dropped at the AP queue, the TCP congestion window size may not change due to the cumulative TCP ACK mechanism. When packets are dropped at the AP queue, the congestion window size of a downlink TCP flow tends to decrease, while the probability for an uplink TCP flow to decrease its congestion window is relatively small.

We propose a dual queue based scheme to alleviate this asymmetric behavior. Two queues are employed at AP; one queue for the downlink TCP data packets, and another queue for the ACK packets corresponding to the uplink TCP flows. The AP can control the service rate of each queue by selecting each queue with different probability when MAC service is ready. To determine the selection probability, we need to analyze the relation between TCP behavior and the dual queue. The analysis of the per-flow queue in [6] can easily be applied to the dual queue scheme by examining the similarities between the two schemes.

Let \( B_D \) and \( B_A \) denote the data queue size and the ACK queue size, respectively. Also \( w_r \) denotes the receiver advertised window size, and \( b \) denotes the number of data packets for an uplink TCP flow that are acknowledged by one ACK. If the delayed ACK of TCP is not used, the value of \( b \) is 1. The data queue is selected with probability \( p \), while the ACK queue is selected with probability \( q = 1 - p \). Let \( R_A \) denote the bandwidth used by AP. Then \( pR_A \) is the service rate of the data queue and \( qR_A \) is the service rate of the ACK queue. Assume that there are \( m \) downlink and \( n \) uplink TCP flows.

When \( B_D < mw_r \), the data queue can be a bottleneck for downlink TCP flows and packet dropping may occur at the data queue of AP. In the steady state, each of the downlink TCP flows operates in the congestion avoidance state and the congestion window size varies between \( B_D/2m \) and \( B_D/m \). If the average round trip time of downlink TCP flows is \( T_d \), the time taken for the congestion window to change from \( B_D/2m \) to \( B_D/m \) is \( B_D/T_d/2m \). Assuming that the propagation delay in wired link is negligible and that there is no congested link in the wired links, the round trip time can be approximated by the average queueing delay at the AP data queue, i.e.,

\[
T_d = \frac{3B_D}{4pR_A}.
\]

We can derive the average downlink TCP data rate per flow \( R_d \) as

\[
R_d = \frac{B_D/m w_d w_r}{B_D/T_d/2m} = \frac{pR_A}{m}.
\]  

When \( B_D \geq mw_r \), the downlink TCP data packets can not fill up the data queue, and no dropping occurs at the data queue. In this case, the average queueing delay of packets at AP corresponds to that of M/M/1 queueing system. Since the round trip time is approximated by the average delay at AP, then

\[
T_d = \frac{1}{pR_A - mR_d}, \quad pR_A > mR_d.
\]

The TCP window size of downlink flows increases to \( w_r \), since no dropping occurs at the AP queue. Then the average downlink TCP data rate per flow is expressed as

\[
R_d = \frac{w_r}{T_d} = w_r(pR_A - mR_d),
\]

and it leads to

\[
R_d = \frac{w_r pR_A}{1 + mw_r} \approx \frac{pR_A}{m},
\]

which is the same as (1).

For uplink TCP flows, packet dropping at the AP queue does not affect the congestion window size due to the cumulative ACK mechanism, and so we assume that the congestion window size of uplink TCP flows are fixed at \( w_r \). Therefore, the average uplink TCP data rate per flow \( R_u \) is

\[
R_u = \frac{w_r}{T_u},
\]

where \( T_u \) is the average round trip time of downlink TCP flows. Here \( T_u \) can be approximated by the average queueing time at AP, too.

When \( B_A < nw_r/b \), the ACK queue of AP is always filled with ACKs. Therefore every ACK packets should wait for \( B_A/qR_A \), i.e., the time taken for \( B_A \) packets to be transmitted. Then the average uplink TCP data rate per flow is

\[
R_u = \frac{w_r}{B_A qR_A}.
\]

When \( B_A \geq nw_r/b \), no packet dropping occurs at the AP ACK queue, and the average queueing time at the AP corresponds M/M/1 queueing delay.

\[
T_u = \frac{1}{qR_A - nR_u/b}, \quad qR_A > nR_u/b.
\]
Hence the average uplink TCP data rate per flow is expressed as

$$R_u = w_r(qR_A - nR_u/b),$$

and it becomes

$$R_u = \frac{w_r q R_A}{1 + w_r n/b} \approx \frac{b q R_A}{n}.$$ 

Then the up/down throughput ratio $R$ is

$$R = \frac{R_u}{R_d} = \begin{cases} \frac{np b}{mq b} & \text{if } B_A < n w_r/b \\ \frac{np b}{mq b} & \text{if } B_A \geq n w_r/b. \end{cases} \quad (2)$$

In order to achieve the fairness among uplink and downlink flows, the average uplink and downlink TCP data rates should be same. From (2) and $p + q = 1$, we can derive the necessary values for $p$ and $q$ as

$$p = \begin{cases} \frac{w_r}{w_r + B_A} & \text{if } B_A < n w_r/b \\ \frac{w_r}{w_r + B_A} & \text{if } B_A \geq n w_r/b, \end{cases} \quad (3)$$

$$q = \begin{cases} B_A & \text{if } B_A < n w_r/b \\ B_A & \text{if } B_A \geq n w_r/b. \end{cases} \quad (4)$$

In the dual queue scheme, the AP selects a queue to transmit a packet using the queue selection probability $p$ and $q$. In order to implement this scheme, AP should classify a packet as a data packet or an ACK packet, and know the number of uplink and downlink flows. One may classify the packets into two categories by simply checking the packet size. The number of uplink and downlink TCP flows can be counted by reading the TCP header of packets.

IV. Simulation Study

To verify the effectiveness of the proposed scheme, we performed simulations with ns-2 [8]. Figure 1 shows the network topology used in the simulation. There are $m$ downlink and $n$ uplink TCP flows at the AP. Each station has only one flow, and each flow shares neither a source nor a destination node. The capacity of the wired links is 100 Mbps and the propagation time is 5 ms for each link. Simulation parameters for the wireless link are given in Table I and are compatible with the IEEE 802.11b standard. Since the bandwidth of the wired link is much higher than that of the wireless link, the wireless link is the only bottleneck link. The maximum window size ($w_r$) of each flow is set to 42 and the delayed ACK of TCP is not used ($b = 1$).

First we ran simulation for the basic AP employing the droptail queue scheme. In this case, both downlink TCP data packets and uplink TCP ACK packets were buffered in a single queue. The queue size was set to 100 packets and there were two uplink flows and six downlink flows. Figure 2(a) shows the throughput of each flow for this scenario. The unfairness among the uplink and downlink TCP flows is significant. The average throughput of uplink flows are 543 kbps while that of downlink flows is 167 kbps.

Figure 2(b) shows the result when the dual queue scheme is deployed to the same scenario, where U1 and U2 denote the uplink TCP flows and D1-D6 denote the downlink TCP flows. In this case we set the size of data queue and ACK queue to 50 packets each, because the queue was divided into two. We can see that the fairness among uplink and downlink TCP flows improves greatly with the dual queue scheme. The average throughput of uplink and downlink flows are 166 kbps and 173 kbps respectively. There is little difference among the throughputs of uplink and downlink flows. If we simulate with a larger queue size, the fairness will be better.

To show that the queue selection probabilities (3) and (4) are reasonable, we simulated with various number of flows. We considered three cases. In the first case, the number of downlink flows was varied from two to ten while the number of uplink flows was fixed at two. In the second case, the numbers of uplink and downlink flows were equal, and the numbers were changed from two to ten with an increment of two. In the third case, the number of uplink flows was varied from two to ten while the number of downlink flows was fixed at two. Figure 3 shows the average throughput ratio between uplink and downlink flows in log scale. Figure 3(a) is the result of the first case, and Fig. 3(b) is the result of the second case. In the case of the single queue scheme, the throughput ratio increases enormously as the number of flows increases. But the ratio stays near 1 at the dual queue scheme, which
Fig. 2. TCP throughput (2 uplink flows and 6 downlink flows)

Fig. 3. TCP throughput ratio

Fig. 4. Fairness index
implies excellent fairness among uplink and downlink TCP flows. From the result we can say that the selection probability for each queue was chosen properly. Also we calculated the Jain’s fairness index [9] to quantify the fairness for each flow. As shown in Fig. 4, in the case of the single queue scheme, the fairness becomes worse as the number of flows increases. For the dual queue scheme, the fairness index is nearly 1 when the number of uplink flows is fixed to two, implying that high degree of fairness is achieved among the flows. The number of uplink flows increases, the fairness index for the dual queue scheme drops by 27%, but it is still much better than the single queue scheme. In the third case, the average throughput ratio between uplink and downlink flows increases from 5.66 to 549 as the number of uplink flows increases for the single queue scheme, while it decreases from 0.91 to 0.43 for the dual queue scheme. The fairness index varies between 0.477 and 0.671 for the single queue scheme while it varies between 0.556 and 0.997 for the dual queue scheme. This means that the dual queue scheme shows better performance than the single queue scheme in the third case also. The graphs related to this case were not presented here due to space limitation.

To see the impact of the queue size, we calculated the Jain’s fairness index for various queue sizes. Simulation was performed with two uplink flows and six downlink flows. In the case of the single queue scheme, we can see in Fig. 5 that the fairness can be improved by increasing the queue size. The queue size plays an important role in the interactions between MAC and TCP. By increasing the queue size, the packet droppings of downlink flows are reduced and the fairness improves. The fairness is achieved when the queue size is somewhere between 200 and 500. Since the number of the maximum in-flight packets generated by eight TCP flows are 336, the fairness may be achieved when the queue size of AP is greater than this value. But increasing the queue size induces a large queueing delay. In the dual queue scheme, the fairness is achieved when queue size is as small as 20. The dual queue scheme enables fair services without employing a large AP buffer.

We simulated the cases where flows have different RTTs. In Fig. 1 of four uplink flows and four downlink flows, we modified the propagation time of the two wired links from 5ms to 10-95ms. Each of the modified links connects the router and a wired node, one for an uplink TCP flow and the other for a downlink TCP flow. The fairness indices in Table II were calculated among the six flows with short RTT (group A) and also between the two flows with long RTT (group B). The result shows that the dual queue scheme gives much higher fairness than the single queue scheme even when there exist significant RTT differences among flows.

V. CONCLUSION

In this paper, we presented the dual queue scheme to solve the TCP unfairness among uplink and downlink flows. In order to solve this problem, we studied the packet dropping behavior and derived a simple formula for the throughput ratio between uplink and downlink flows in the dual queue scheme. To make the throughput ratio equal to 1, we showed how to set the queue selection probabilities for the downlink data queue and the uplink ACK queue. The simulation results show that the TCP fairness for the dual queue scheme improves significantly compared to that of the single queue scheme. The advantage of the dual queue scheme is that it is very simple, effective, requires only small changes in the AP interface queue and no changes in the MAC layer.

ACKNOWLEDGEMENT

This work has been partly supported by POSCO. We thank Eun-chan Park for the discussions on TCP unfairness problem and the helpful suggestions.

REFERENCES


![Fig. 5. Fairness index as the queue size changed](image-url)