Journal of Institute of Control, Robotics and Systems (2013) 19(7):653-660
http://dx.doi.org/10.5302/J.ICROS.2013.13.1926 ISSN:1976-5622 eISSN:2233-4335

IEEE 802.11n 무선랜에서 상향링크 TCP 플로우간 형평성 향상을 위한 TCP ACK 압축기법

TCP Acknowledgement Compression for Fairness Among Uplink TCP Flows in IEEE 802.11n WLANs

김 민 호, 박 은 찬*, 김 웅 섭
(Minho Kim[1], Eun-Chan Park[2], and Woongsup Kim[2])
[1]Digital Media & Communications R&D Center, Samsung Electronics Co., Ltd.
[2]Department of Information and Communication Engineering, Dongguk University-Seoul

Abstract: This paper deals with the problem of unfairness among uplink TCP (Transmission Control Protocol) flows associated with frame aggregation employed in IEEE 802.11n WLANs (Wireless Local Area Networks). When multiple stations have uplink TCP flows and transmit TCP data packets to an AP (Access Point), the AP has to compete for channel access with stations for the transmission of TCP ACK (acknowledgement) packets to the stations. Due to this contention-based channel access, TCP ACKs tend to be accumulated in the AP's downlink buffer. We show that the frame aggregation in the MAC (Medium Access Control) layer increases TCP ACK losses in the AP and leads to the serious unfair operation of TCP congestion control. To resolve this problem, we propose the TAC (TCP ACK Compression) mechanism operating at the top of the AP's interface queue. By exploiting the properties of cumulative TCP ACK and frame aggregation, TAC serves only the representative TCP ACK without serving redundant TCP ACKs. Therefore, TAC reduces queue occupancy and prevents ACK losses due to buffer overflow, which significantly contributes to fairness among uplink TCP flows. Also, TAC enhances the channel efficiency by not transmitting unnecessary TCP ACKs. The simulation results show that TAC tightly assures fairness under various network conditions while increasing the aggregate throughput, compared to the existing schemes.

Keywords: IEEE 802.11n, fairness, frame aggregation, TCP ACK

I. INTRODUCTION

In recent years, the exploding use of mobile devices such as smart phones and tablet PCs accelerates the demand for Internet access via WLANs (Wireless Local Area Networks) [1]. Most Internet services use TCP (Transmission Control Protocol) as a transport-layer protocol and the amount of uplink traffic is expected to increase rapidly due to emerging services such as peer-to-peer contents sharing, audio/video streaming, and mobile video telephony.

This paper focuses on the fairness problem among uplink TCP flows in infrastructure WLANs. Consider that several wireless STAs (stations) are associated with an AP (Access Point) and STAs transmit TCP data packets to the AP, as depicted in Fig. 1. Then, the AP should compete with STAs to transmit the corresponding TCP ACKs (acknowledgments), according to the channel access mechanism of WLAN, i.e., DCF (Distributed Coordination Function). Since DCF is designed to assure fair channel access opportunity among competing STAs including AP, the AP necessarily has a little chance to serve TCP ACKs as the number of STAs increases. This bottleneck for the TCP ACKs in AP's downlink buffer eventually results in ACK loss due to buffer overflow. This ACK loss along with TCP's cumulative ACK mechanism affects the behavior

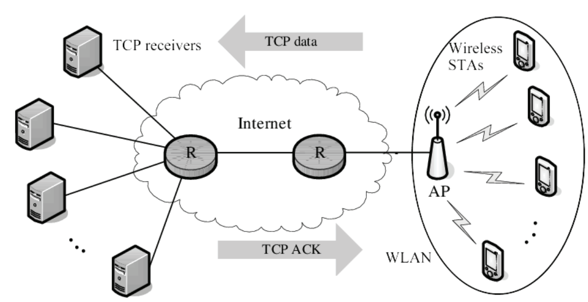


그림 1. 무선랜에서 상향링크 TCP 플로우의 통신망 구조.
Fig. 1. Network topology of multiple uplink TCP flows in WLAN.

* 책임저자(Corresponding Author)
Manuscript received May 30, 2013 / revised June 5, 2013 / accepted June 5, 2013
김민호: 삼성전자 DMC R&D 센터(minomino0717@gmail.com)
박은찬, 김웅섭: 동국대학교-서울, 컴퓨터정보통신공학부
(ecpark@dongguk.edu/woongsup@dongguk.edu)
※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science, and Technology [Grant No. 2012-0002019].

of TCP flows in an asymmetric way and leads to the unfairness problem among uplink TCP flows in infrastructure WLANs.

The unfairness problem has been addressed in the literatures [2-5]. The works in [2] and [3] aim to prevent buffer overflow in the AP by giving the AP more channel access chances. This can be accomplished by providing high priority to the AP with the aid of differentiated channel access mechanism of IEEE 802.11e [2], or by assuring bi-directional channel access for the STA and AP [3]. However, this approach has a drawback, i.e., the aggregate throughput may be decreased, because the prioritized channel access by the AP increases the probability of collision with STAs [2], or the bi-directional channel access opportunity given to the AP may be wasted due to lack of backlogged TCP ACK [3]. In [4], the generation of TCP ACK is regulated to assure fairness by intentionally dropping the received TCP data packet in the AP based on its queue occupancy. This is effective to assure fairness, but it may decrease throughput since the TCP data packets that were successfully delivered to the AP are dropped. To mitigate the unfairness among uplink TCP flows, a specific scheduling algorithm is proposed in [5]; the AP schedules the TCP ACK transmission so that the TCP flow with smaller CWND (congestion window) receives TCP ACK preferentially. On the other hand, the unfairness problem arising between uplink TCP flows and downlink TCP flows in infrastructure WLANs has been widely studied and many solutions have been proposed [6-11].

The IEEE 802.11n standard [12] introduces several new MAC (Media Access Control) features such as frame aggregation, block ACK, and reverse direction grant, to increase the efficiency of MAC. With the frame aggregation scheme, multiple packets are aggregated and transmitted in a single frame so as to enhance throughput by reducing the channel access overheads, e.g., PHY/MAC header, backoff time, and MAC-layer ACK frame transmission time. Unless otherwise stated, we refer to a packet as what the upper layer transfers to the MAC layer and a frame as what MAC layer transfers to the lower layer. Also, there are two kinds of ACKs, TCP ACK and MAC ACK, we distinguish them by indicating the terms “TCP” or “MAC” in front of ACK. To our best knowledge, none of the existing studies considers the effect of the IEEE 802.11n frame aggregation on the unfairness problem among uplink TCP flows. This is the very first study revealing that the unfairness problem becomes remarkably exacerbated due to an unintended interaction between MAC frame aggregation and TCP congestion control. Moreover, the existing approaches are not suitable for the case where frame aggregation is employed, which will be verified by simulation study.

In this paper, we provide a simple solution to resolve the unfairness problem. We propose the TAC (TCP ACK Compression) mechanism operating at the top of AP's IFQ (Interface Queue). The key idea of TAC is to prevent TCP ACK loss due to buffer overflow, which is the primary reason of unfairness. For this purpose, we make use of the properties of TCP cumulative ACK and frame aggregation. Once TCP ACKs arrive to the AP in bursts due to frame aggregation, TAC serves only the representative ACK and discards unnecessary other TCP ACKs by exploiting the nature of cumulative TCP ACK. In this way, TAC mitigates the unfairness problem by reducing the queue occupancy and it also increases the total throughput by not transmitting the unnecessary TCP ACKs. The *ns-2* simulation [13] results show that TAC tightly assures fairness among uplink TCP flows while enhancing aggregate throughput up to 30%, compared to the existing schemes.

The rest of the paper is organized as follows. In Section II, we introduce the frame aggregation scheme in IEEE 802.11n and state the unfairness problem related to frame aggregation. In Section III, we propose the TAC mechanism as a solution for the unfairness problem. Then, we present simulation results in Section IV to evaluate and compare the performance of TAC with other schemes. Finally, the conclusion follows in Section V.

II. PROBLEM STATEMENT

IEEE 802.11n introduces two kinds of frame aggregation schemes, A-MSDU (Aggregated MAC-level Service Data Unit) and A-MPDU (Aggregated MAC-level Protocol Data Unit) [14]. Throughout this paper, we consider only A-MPDU with BA (Block ACK) because it can achieve higher throughput than A-MSDU in an error-prone wireless channel. With the A-MPDU scheme, only corrupted MPDU sub-frames can be selectively retransmitted thanks to individual error detection code for each sub-frame. When the STA gets a chance to access channel, it transmits an A-MPDU frame consisting of several TCP data packets. We consider that each MPDU sub-frame contains one TCP data packet. After transmitting the aggregated frame, the STA requests for BA to the AP. On the BA request, the AP informs the STA of the transmission success or failure of each sub-frame by transmitting BA. When TCP data packets are delivered to the final TCP receiver via the AP, the TCP receiver generates the corresponding TCP ACKs and they will arrive to the AP. Then, the AP competes with other STAs to get the channel access and it finally transmits an A-MPDU frame consisting of several TCP ACKs to the STA.

The frame aggregation can increase the channel efficiency by removing several overheads; however, it makes a negative effect on the fairness among uplink TCP flows. The reason of unfairness related with frame aggregation can be analyzed from two viewpoints; (*i*) the increase in the delay of AP's channel access and TCP ACK loss in the AP, and (*ii*) the difference in the CWND of TCP flows and the cumulative TCP ACK mechanism.

Several TCP data packets aggregated into a single frame will be delivered to the TCP receiver in bursts, and the corresponding TCP ACKs will also be generated and arrive to the AP in bursts. Compared to the case without frame aggregation, the transmission time of each STA increases due to aggregation and the time interval between two consecutive channel accesses by the AP will increase accordingly. Therefore, the AP has to accommodate the burst TCP ACKs until it gets the channel access chance and the probability of ACK loss due to buffer overflow in the AP increases, which is one reason of the unfairness problem. This problem becomes worsen due to the unfair TCP behavior according to the cumulative ACK mechanism. The increment of TCP flow's CWND is proportional to the number of TCP ACKs received in the aggregated frame. On the burst arrival of TCP ACKs, the TCP sender (i.e., STA) increases CWND rapidly. As the degree of frame aggregation increases, the difference among CWND values of TCP flows become larger. Due to this difference in CWND and the property of cumulative TCP ACK mechanism, some TCP flows with larger CWND can tolerate the loss of TCP ACK in the AP (i.e., they increase CWND as if there were no TCP ACK losses at all and they do not retransmit the TCP data packets) while other TCP flows with smaller CWND become susceptible to the TCP ACK loss (i.e., time-out may occur and they reset CWND to the initial value, as well as unnecessarily retransmitting the TCP data packets that were successfully delivered to the AP). Consequently, the frame aggregation exacerbates the unfairness problem among uplink TCP flows.

To verify the unfairness problem, we performed preliminary simulations with the network topology in Fig. 1, where there are 20 STAs and each STA has one uplink TCP flow. The A-MPDU is employed with the maximum aggregated frame size, $L_{\max}$, and AP can buffer up to B packets. The size of TCP data packet and transmission rate are set to the same for all the STAs. Other simulation settings are the same as described in Sec. IV. Fig. 2 shows the aggregate throughput and Jain's fairness index [15] for the limited (B=200) and infinite buffer sizes (B=10,000) when $L_{\max}$ varies from 1 KB to 32 KB. In the case of infinite buffer size, the value of B is set to be much higher than the product of the maximum size of TCP CWND and the number of TCP flows, so that the buffer

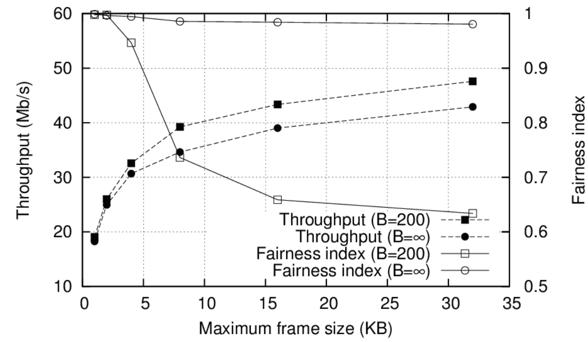


그림 2. 최대 집적 프레임의 크기에 따른 전체 처리율과 형평성 지표.

Fig. 2. Aggregate throughput and fairness index for several values of the maximum aggregated frame size.

can hold all TCP ACKs without overflow. From Fig. 2, we can observe the effects of frame aggregation and AP buffer size on fairness and the trade-off between throughput and fairness.

- When $L_{\max}$ is small (1~2 KB), the unfairness problem is negligible, regardless of the buffer size. This is because the buffer size is large enough to hold TCP ACKs without buffer overflow.
- As $L_{\max}$ increases, the unfairness problem becomes severe in the case of the limited buffer size. In contrast, the problem is insignificant with the infinite buffer, which implies that the problem results from the buffer overflow in the AP.
- The aggregate throughput increases as $L_{\max}$ increases for both cases. The throughput with the infinite buffer is smaller than that with the limited buffer, because the queuing delay in the AP with the infinite buffer increases RTT (Round Trip Time) of TCP flows.

Moreover, we observe the loss rate of TCP ACKs in AP's downlink buffer when the buffer size is limited, i.e., B = 200, and $L_{\max}$ varies from 1 KB to 32 KB. In Fig. 3, the points with ×-mark indicate the ACK loss rate of individual 20 TCP flows and the square and circle points indicate their average and standard deviation values, respectively. The results in Fig. 3 reconfirm the reason of unfairness problem. When $L_{\max}$ = 1 KB, i.e., the frame aggregation is not applied, ACK loss does not occur at all for most TCP flows and only a few losses occur for a certain flow. Also, when $L_{\max}$ = 2 KB, there is no remarkable difference in the TCP ACK loss rates among TCP flows, the loss rates range between 0 and 2.6%. In these cases, the unfairness problem hardly happens, as already shown in Fig. 2. However, the deviation of loss rate, as well as its average value, increases significantly as $L_{\max}$ increases. When $L_{\max}$ is increased to 32 KB, the average and standard deviation of ACK loss rate increase

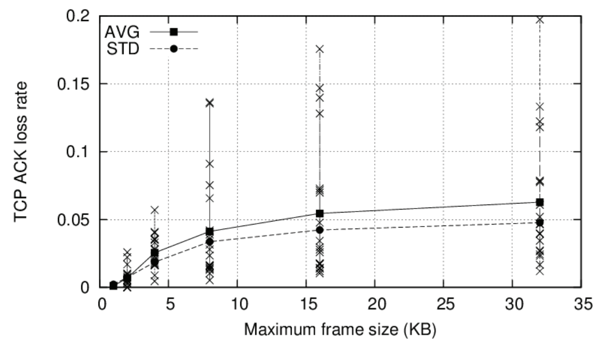


그림 3. 최대 집적 프레임의 크기에 따른 플로우별 TCP ACK 손실률.

Fig. 3. Per-flow TCP ACK loss rate for several values of the maximum aggregated frame size.

by 63 and 22 times, respectively, compared to the case of L_{max} = 1 KB. Especially when L_{max} = 16 KB, 8 TCP flows suffer from small loss rate less than 2%, whereas 4 TCP flows suffer from serious loss rate exceeding 12%. Also when L_{max} = 32 KB, the minimum and maximum values of loss rate are 1.2% and 19.8%, respectively. As a result, the deviation in the TCP ACK loss rates shown in Fig. 3 debases the fairness among TCP flows.

III. TCP ACK COMPRESSION MECHANISM

1. Basic algorithm

To resolve the unfairness problem, we propose the TAC mechanism which is implemented at the top of AP's IFQ. The key idea of TAC is motivated by the nature of cumulative TCP ACK mechanism combined with frame aggregation. The TCP ACK cumulatively acknowledges all the TCP data packets up to the sequence number indicated in the TCP ACK. Recall that multiple TCP ACKs will arrive in bursts at AP's IFQ when the frame aggregation is used. We define the *RACK* (*Representative TCP ACK*) as the TCP ACK with the highest sequence number among these burst TCP ACKs. For the time being, we do not consider the loss of TCP data/ACK packets. It is noteworthy that the TCP ACKs except RACK are redundant because RACK can representatively acknowledge the successful delivery of all the TCP data packets transmitted in bursts.

The goal of TAC is to determine RACK and to enqueue only RACK to AP's IFQ. For this purpose, TAC manages a logical queue to pick out the RACK. We assume that TAC can identify TCP flows by observing the destination address or port number. When the frame aggregation is used, the time gap between the consecutive TCP ACKs that belong to the same TCP flow and arrive to the AP is at most T_ϵ, which depends on several delays in the network, e.g., link transmission delay, node processing delay, or queuing delay. The design guideline to set the value of T_ϵ

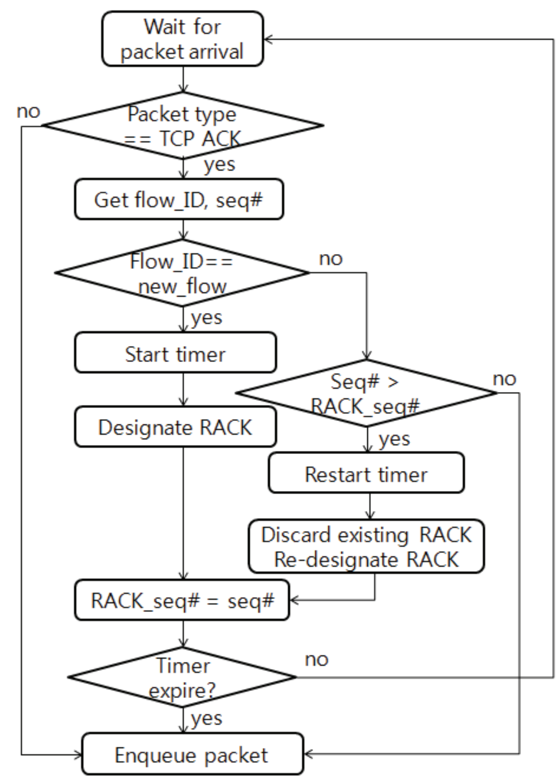


그림 4. TCP ACK 압축 메커니즘의 흐름도.

Fig. 4. Flow chart of TCP ACK compression mechanism.

will be given in the following sub-section. Consider that TAC maintains a timer whose value is T_ϵ for each TCP flow. When the first TCP ACK of a specific TCP flow arrives in the AP, TAC designates this packet as *RACK*, and starts the timer and waits for the subsequent TCP ACKs until the timer reaches T_ϵ. If a subsequent TCP ACK, referred to as a *new* packet, arrives before the timer expires, TAC observes its destination address or port number, as well as the sequence number. If it is determined to belong to the same flow with the RACK and its sequence number is higher than that of RACK, the new packet becomes as RACK and the previous RACK is discarded. After then, TAC restarts the timer and waits for the subsequent TCP ACKs as the same way in the above. If there is no new arrival of TCP ACKs within T_ϵ, TAC enqueues the current RACK to the IFQ so that it is delivered to the MAC layer. If the sequence number of new packet belonging to the same flow with the RACK is not greater than that of RACK, this new packet is not redundant; it is probably the duplicate TCP ACK to request retransmission of lost TCP data packet and to inform the TCP sender of network congestion. Thus, in this case, TAC immediately enqueues the new packet to the IFQ. In this way, TAC can enqueue only RACK to the IFQ, while discarding unnecessary TCP ACKs. Fig. 4 shows the flow chart of TAC.

2. Discussion of TAC

The proposed TAC algorithm requires a minimal change in the AP without any changes in the STA, and it modifies neither the MAC protocol nor the TCP protocol at all. Even with this minimal change, the TAC is designed to operate in a robust way under the cases where TCP data packet or RACK is lost and where uplink TCP flows coexist with downlink TCP flows Also, the effect of T_ϵ on the performance of TAC needs to be investigated. We discuss the following several points related to the operation of TAC.

2.1 Loss of TCP data packet

We consider two reasons of TCP data packet loss.

- **Loss due to channel error or collision in the wireless link:** TAC operates in the same way even in this case, as long as A-MPDU is implemented such that it assures in-order delivery. If the transmission of a sub-frame in the A-MPDU frame failed, the corrupted sub-frame is retransmitted in the MAC layer, and the MAC layer does not deliver sub-frames that were successfully received to the upper layer until it successfully receives the retransmitted sub-frame. Thus, the loss of TCP data packet can be recovered by the MAC-layer retransmission, and in-order TCP data packets arrive to the AP in bursts.
- **Loss due to buffer overflow in the wired link:** In this case, TAC receives duplicate TCP ACKs, which imply TCP data packet loss and invoke TCP retransmission. Recall that TAC discards TCP ACKs only if the new packet has higher sequence number than the current RACK. Therefore, TAC does not discard these duplicate TCP ACKs and it does not make any negative effect on the operation of TCP congestion control.

2.2 Loss of RACK

RACK can also be corrupted due to wireless channel error or collision, but this can be recovered by the MAC-layer retransmission. It is important to note that the time of MAC-layer retransmission (e.g., several tens of milliseconds) is much shorter than TCP time-out (e.g., several hundreds of milliseconds), and thus, the loss of RACK can be recovered before TCP time-out occurs. Moreover, the error rate of RACK is usually lower than that of TCP data packet because the size of TCP ACK packet (40 bytes including IP header) is much smaller than that of TCP data packet (500 ~ 1500 bytes) and that the error rate of A-MPDU sub-frame is usually proportional to the size of sub-frame.

2.3 Coexistence with downlink TCP flows

Recall that TAC is applied only to TCP ACKs for uplink TCP flows. If there exist downlink TCP flows, their TCP data packets arrive to the AP. They bypass the TAC and are enqueued to the IFQ of AP. Also, the TCP ACKs for downlink TCP flows will arrive from the STA to the AP but they are delivered from the lower layer to the MAC layer without any processing by the TAC. Thus, TAC works well if downlik TCP flows coexist with uplink TCP flows.

2.4 Effect of RACK timer

The TAC algorithm has the sole design parameter, T_ϵ, which is used to determine RACK among burst TCP ACKs. If it is set to a small value, TAC makes the premature decision although subsequent TCP ACKs will arrive soon, and then, it enqueues redundant RACKs. The extreme case where T_ϵ is zero becomes equivalent to the case where TAC is not implemented, i.e., every TCP ACK is enqueued as an individual RACK. Otherwise if T_ϵ is set to a large value, TAC waits for subsequent TCP ACKs long time unnecessarily, which will increase the delay of RACK transmission. However, in both cases, TAC does not result in any critical negative effect on TCP behaviors.

We provide a guideline to set the value of T_ϵ. Firstly, the time gap between two consecutive TCP ACKs cannot exceed the RTT (e.g., 100 ms). Secondly, it cannot be smaller than the transmission delay of the slowest link in the network (e.g., 0.8 ms when the packet size is 1 KB and the link capacity is 10 Mb/s). Thirdly, the time gap probably increases as the number of links between TCP sender and receiver increases and the network becomes congested. Taking all these points into account, we conclude that the appropriate range of T_ϵ lies between several milliseconds and a few tens of milliseconds for the typical network configuration.

IV. PERFORMANCE EVALUATION

1. Simulation configuration

We evaluate the performance of TAC via *ns-2* simulation [13]. We consider the network topology in Fig. 1, where N STAs exist and each STA has one uplink TCP flow. We used TCP-NewReno, the most common version of TCP in the current Internet, and the greedy FTP traffic was generated and transferred over the TCP flows. The sizes of TCP data packet was set to 1024 bytes. The size of maximum CWND and IFQ were set to 50 packets and 200 packets, respectively. The parameters of MAC/PHY were set according to the IEEE 802.11n standard. In order to focus on the unfairness resulting from TCP behaviors and to rule out the unfairness in multi-rate WLANs, we set the transmission rate of all the STAs to the same value of 65 Mb/s. The capacity of all the wired links in Fig. 1 was set to 100 Mb/s. As performance measure, Jain's fairness index and aggregate throughput were considered. The performances were evaluated and compared for the following four schemes.

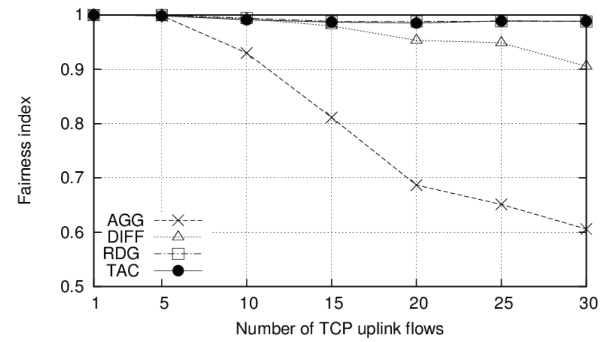
- AGG: This is the baseline scheme that only employs A-MPDU without any mechanisms to assure fairness.
- DIFF: This scheme differentiates the channel access chances between AP and STAs to mitigate the unfairness problem. The value of minimum contention window was set to 7 and 15 for the AP and STAs, respectively, to allow the preferential channel access of the AP. This scheme can reduce the TCP ACK loss due to buffer overflow in the AP, which is similar to the idea in [2].
- RDG: The RDG (Reverse Direction Grant) mechanism with A-MPDU is employed to grant the bi-directional channel access opportunity to STA and AP, as similar to [3]. If a STA gets the channel access chance and finishes transmitting TCP data packets to the AP, then the channel access opportunity is given to the AP so that the AP can transmit its backlogged TCP ACKs without the channel access procedure. This scheme reduces the channel access delay for TCP ACKs, as well as reducing ACK loss in the AP.
- TAC: This is the proposed mechanism. The value of T_ϵ was set to 5 ms according to the guideline given in the previous section.

In all the schemes, A-MPDU was implemented such that a STA including AP aggregates as many packets as possible within the maximum size of aggregated frame. If the STA (or AP) gets the chance of channel access but the number of backlogged packets is not large enough to construct the maximum allowed size of aggregated frame, which is possible due to TCP congestion control mechanism, it constructs the aggregated frame with the existing packets in its transmission queue.

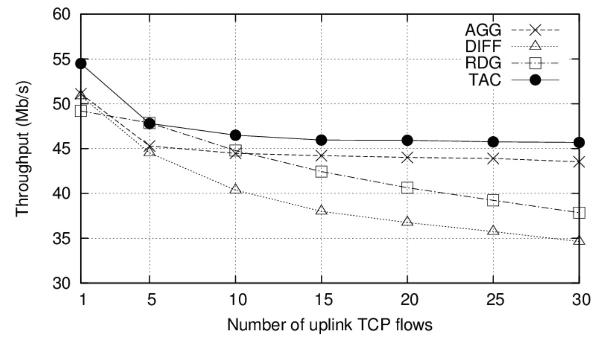
2. Effect of number of flows

The first simulation investigates how the number of uplink TCP flows (or the number of STAs) affects fairness and throughput. This simulation was performed in the case of error-free ideal channel condition and the maximum size of aggregated frame, $L_{\max}$, was set to 16 KB.

Fig. 5(a) shows the fairness index (FI) when N changes from 1 to 30. In the case of AGG, FI decreases as N increases, because a large number of TCP ACKs are accumulated in the IFQ of AP as N increases, which results in ACK losses due to buffer overflow. Especially when N = 30, FI of AGG is about 0.6. However, the fairness is tightly attained in RDG and TAC; FI is at least 0.99 for the entire range of N. This is because they can prevent buffer overflow by maintaining queue occupancy to a moderate level. In the case of DIFF, FI slightly decreases as N increases, e.g., it is about 0.9 when N = 30, implying that the performance of DIFF is not well scalable with respect to N.



(a) Fairness index.

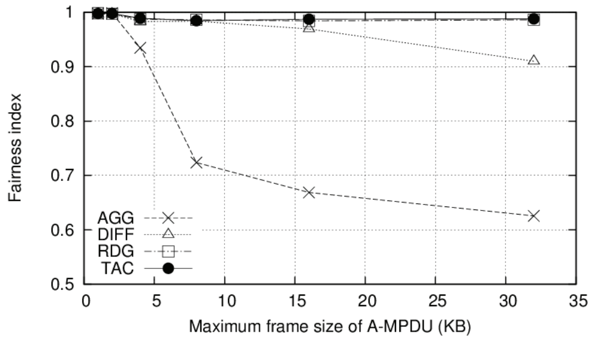


(b) Aggregate throughput.

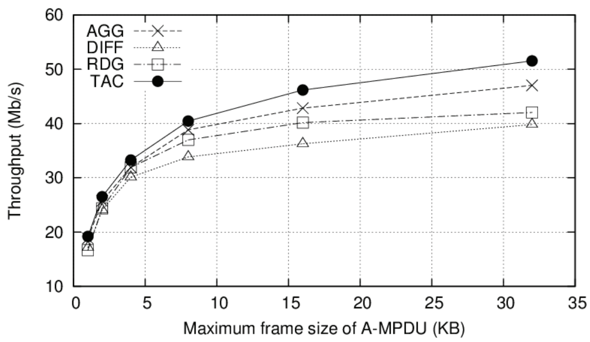
그림 5. 상향링크 TCP 플로우 수에 따른 여러 가지 기법의 성능 비교.

Fig. 5. Performance comparison of several schemes with respect to the number of uplink TCP flows.

The throughput with respect to the value of N is shown in Fig. 5(b). The throughputs of AGG and TAC are kept nearly constant as N varies from 5 to 30. The reason is as follows. In contrast to the case where STAs have infinite amount of UDP traffic, the collision probability does not notably increase even though N increases, because TCP adjusts its transmission rate based on the congestion control mechanism and the effective number of contending STAs is not proportional to N but remains nearly constant [16]. However, in the cases of DIFF and RDG, the throughput remarkably decreases as N increases; it decreases from 51 and 49 Mb/s to 35 and 38 Mb/s, respectively, as N increases from 1 to 30. The throughput decrease of DIFF and RDG results from the increase of collision probability. In the case of DIFF, the smaller size of AP's contention window contributes to increasing the channel access probability of AP to serve TCP ACKs fast; however, the aggressive channel access of AP also increases the collision probability, which degrades the throughput. This is similar to the case of RDG; the STAs always participate in the competition of channel access because STAs receive TCP ACKs immediately after sending TCP data packets, and thus, the collision probability increases as N increases. The throughput improvement of TAC over DIFF and RDG is up to 32% and 21%, respectively. The simulation results in



(a) Fairness index.



(b) Aggregate throughput.

그림 6. 최대 집적 프레임 크기에 따른 여러 가지 기법의 성능 비교.

Fig. 6. Performance comparison of several schemes with respect to the maximum aggregated frame size.

Fig. 5 confirm that the proposed TAC scheme tightly assures fairness, as well as improving total throughput, while the other schemes suffer from the trade-off between fairness and throughput.

3. Effect of frame aggregation

Next, we focus on the effect of frame aggregation on the fairness and throughput. In this simulation, N was set to 20 and L_{max} was varied from 1 KB to 32 KB. The wireless channel was considered to be error-prone with the moderate bit error rate of 10^{-5}, so that the error rates of A-MPDU sub-frame containing a TCP data packet and an ACK packet are about 8% and 0.3%, respectively.

The effect of L_{max} on fairness is shown in Fig. 6(a), which is similar to that of N (compare Fig. 6(a) with Fig. 5(a)). The performance of AGG is significantly degraded as L_{max} increases, while those of RDG and TAC are almost immune to the change of L_{max}, i.e., FI is not below 0.98, regardless of the value of L_{max}. As L_{max} increases, DIFF does not work well; FI drops to 0.91 when L_{max} = 32 KB.

On the other hand, as shown in Fig. 6(b), the throughputs of all the schemes increase as L_{max} increases, because the channel access overhead is effectively decreased as more A-MPDU sub-frames are aggregated. The throughput of AGG is higher than those of DIFF and RDG by up to 18% and 12%, respectively. Together with the results in Fig. 6(a), these results mean that DIFF and RDG assure fairness at the cost of aggregate throughput, whereas AGG improves throughput at the cost of fairness. However, TAC outperforms all the other schemes in terms of both fairness and throughput. The throughput of TAC is higher than those of AGG, DIFF, and RDG by up to 10%, 29%, and 23%, respectively. It is worthwhile to note that the throughput improvement by TAC over the other schemes increases as L_{max} increases. This is because TAC does not transmit unnecessary TCP ACKs, whose number increases in proportion to the value of L_{max}. Consequently, TAC increases the overall throughput, as well as achieving fairness.

V. CONCLUSION

In this paper, we have investigated the effect of frame aggregation on the fairness among uplink TCP flows in IEEE 802.11n WLANs. We have shown that the frame aggregation exacerbates the unfairness problem, and we have analyzed its reason from the viewpoints of the TCP ACK loss in AP's buffer and the characteristic of TCP congestion control. To resolve this problem, we have proposed the TAC mechanism. By discarding unnecessary TCP ACKs and serving only the RACK in AP's IFQ, TAC prevents buffer overflow and assures fairness among uplink TCP flows. At the same time, TAC enhances the aggregate throughput by effectively reducing the number of TCP ACKs to be transmitted. The strength of TAC is that it needs to be implemented only in the AP without any additional implementation in the STAs, and it does not modify the standard TCP and MAC protocols. The simulation results have confirmed that TAC outperforms the conventional approaches in terms of both fairness and aggregate throughput under various network configurations.

REFERENCES

[1] L. Munõz, R. Agero, J. Choque, J. A. Lrastorza, L. A. G. Sánchez, M. Petrova, and P. Möahönen, “Empowering next generation wireless personal communication netorks,” *IEEE Communication Magazine*, vol. 42, no. 5, pp. 64-70, May 2004.

[2] D. J. Leith and P. Clifford, “Using the 802.11e EDCF to achieve TCP upload fairness over WLAN links,” *Proc. of Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks* (WiOPT), pp. 109-118, Apr. 2005.

[3] H. T. Wu and S. D. Cheng, “DCF+: An enhancement for reliable transport protocol over WLAN,” *Journal of Computer Science and Technology*, vol. 18, no. 2, pp. 201-209, Mar. 2003.

[4] Y. Hirano and T. Murase, “Uplink TCP traffic control with monitoring downlink buffer for throughput fairness over wireless LANs,” *Proc. of Personal, Indoor and Mobile Radio Communications* (PIMRC), pp. 737-741, Sep. 2009.

[5] F. Keceli, I. Inan, and E. Ayanoglu, “TCP ACK congestion control and filtering for fairness provision in the uplink of IEEE 802.11 infrastructure basic service set,” *Proc. of IEEE International Conference on Communications* (ICC), pp. 4512-4517, Jun. 2007.

[6] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, “Understanding TCP fairness over wireless LAN,” *Proc. of IEEE INFOCOM*, pp. 863-872, Apr. 2003.

[7] Y. Wu, Z. Niu, and J. Zheng, “Study of the TCP upstream/downstream unfairness issue with per-flow queueing over infrastructure-mode WLANs,” *Wirless Communications and Mobile Computing*, vol. 5, no. 4, pp. 459-471, Jun. 2005.

[8] D. J. Leith, P. Clifford, D. W. Malone, and A. Ng, “TCP fairness in 802.11e WLANs,” *IEEE Communications Letters*, vol. 9, no. 11, pp. 964-966, Nov. 2005.

[9] Q. Wu, M. Gong, and C. Williamson, “TCP fairness issues in IEEE 802.11 wireless LANs,” *Computer Communications*, vol. 31, no. 10, pp. 2150-2161, Jun. 2008.

[10] Q. Xia, X. Jin, and M. Hamdi, “Active queue management with dual virtual proportional integral queues for TCP uplink/downlink fairness in infrastructure WLANs,” *IEEE Trans. on Wireless Communications*, vol. 7, no. 6, pp. 2261-2271, Jun. 2008.

[11] E.-C. Park, D.-Y. Kim, H. Kim, and C.-H. Choi, “A cross-layer approach for per-station fairness in TCP over WLANs,” *IEEE Trans. on Mobile Computing*, vol. 7, no. 7, pp. 898-911, Jul. 2008.

[12] IEEE 802.11 Working Group, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput,” *IEEE Std. 802.11n-2009*, Oct. 2009.

[13] NS-2 Network Simulator, http://www.isi.edu/nsnam/ns.

[14] D. Skordoulis, Q. Ni, H. H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour, “IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs,” *IEEE Wireless Communications*, vol. 15, no. 1, pp. 40-47, Feb. 2008.

[15] D.-M. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1-14, Jun. 1989.

[16] R. Bruno, M. Conti, and E. Gregori, “Analytical modeling of TCP clients in Wi-Fi hot spot networks,” *Proc. of NETWORKING 2004*, pp. 626-637, May 2004.

Minho Kim

received B.S. degree from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, in 2007, and Ph.D. degree from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea, in 2013. He is currently a senior engineer with Samsung Electronics, Co., Ltd. His research interests are in the area of wireless networks, including wireless LAN, mobile WiMAX, link adaptation, multicasting, wireless TCP.

Eun-Chan Park

received B.S., M.S., and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea, in 1999, 2001, 2006, respectively. He is currently an Assistant Professor in the Department of Information and Communication Engineering, Dongguk University-Seoul, Korea since 2006. His research interests include performance analysis, resource allocation, quality of service, congestion control, cross-layer optimization in wired and wireless networks.

Woongsup Kim

received B.S. degree in Computer Engineering from Seoul National University, Korea in 1998, M.S. degree in Computer and Information Science from University of Pennsylvania, USA, in 2001, Ph.D. degree in Computer Science and Engineering from Michigan State University, USA, in 2006 respectively. Since 2007, he has been with the Department of Information and Communication Engineering, Dongguk University-Seoul, Korea, where he is currently a Assistant Professor. His research interests are in the areas of software engineering, and cloud computing.