Augmented Split-TCP over Wireless LANs

Hakyung Jung, Nakjung Choi, Yongho Seok, Taekyoung Kwon and Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {hkjung, fomula, yhseok, tk, yhchoi}@mmlab.snu.ac.kr

Abstract— **This paper introduces a new split-TCP approach for improving TCP performance over IEEE 802.11-based wireless LANs. TCP over wireless LANs is not aggressive, which is a fundamental reason for poor performance. We propose augmented split-TCP (AS-TCP) to mitigate this problem. Our scheme extends the split-connection approach that divides a connection into two different connections at a split point such as an access point. Using AS-TCP, a mobile host emulates TCP ACK packets using MAC ACK frames, instead of receiving real TCP ACK packets. We compared AS-TCP with both normal TCP and I-TCP (indirect TCP) by simulation. Results show that AS-TCP achieves higher throughput, fairer resource allocation and, in power-saving mode, shorter delays.**

I. Introduction

Transmission control protocol (TCP) has been very widely used as a transport layer protocol to provide a reliable stream service through the use of acknowledgments (ACKs). However, it is well-known that TCP, as used in wired networks, is not well suited to a wireless network environment because it has no ability to differentiate between packet loss due to network congestion and wireless channel errors. TCP reacts even to losses that are not related to congestion by initiating congestion control and avoidance algorithms, resulting in degraded end-to-end performance. Several approaches, such as split-TCP, local retransmission, and forward error correction, have been proposed to address this problem [1][2][3]. However, when these schemes are applied to the wireless LANs, they do not achieve the desired performance enhancement since most data packets can be transmitted successfully at the transport layer, by means of retransmissions in the MAC layer, which is not the case in other wireless networks. To alleviate the performance degradation by TCP over wireless LANs, new protocols [4] [5] have recently been proposed. In these protocols, modifications have been made, not in the transport layer but in the link or MAC layer. Although these new protocols show some improvement in terms of either fairness or balance between uplink and downlink, the improvement of TCP throughput itself has not been significant.

The fundamental reason for the poor performance of TCP over wireless LANs is that TCP is not aggressive. There are three important factors in this phenomenon: First, TCP suffers from self-contention between forward TCP data and backward TCP ACKs even in a single TCP flow. When TCP runs over a wireless LAN, where a shared channel is used for multiple access, TCP data and TCP ACKs will compete the channel, which may cause collision and degrade the overall performance. Second, in congested wireless LANs, only a small number of mobile hosts contend for a channel [6]. After a TCP sender transmits its *cwnd*[1] data, it cannot send any more data until it receives a TCP ACK from the recipient. In other words, a mobile host waiting for a TCP ACK cannot deliver data packets to the MAC layer, so it cannot participate in contention for the wireless channel. Third, the IEEE 802.11 power saving mode (PSM) can impede TCP [7]. When 802.11 PSM is activated, an AP buffers any packets destined for mobile hosts until the next beacon interval. TCP ACKs from TCP receivers have to be buffered at the AP, resulting in the increase of round trip time (RTT) especially on short-lived TCP flows. Without addressing these three factors, it is hard to achieve good TCP performance.

The rest of this paper is organized as follows. In Section II, we discuss related works on enhancing the performance of TCP flows over wireless links. Section III characterizes the problem of TCP over wireless LANs, and then proposes a novel scheme called augmented split-TCP (AS-TCP) to improve TCP performance. Section IV presents simulation results in terms of aggregated throughput, fairness index and mean delay. We conclude this paper in Section V.

II. Related Work

A. Improving TCP performance over wireless networks

To improve the performance of TCP over wireless networks, I-TCP [3] uses a split-connection approach. It divides a TCP connection into two distinct connections: a wired connection between a fixed host and an AP, and a wireless connection between an AP and a mobile host. When an AP receives a TCP data packet through the wired connection, it sends an TCP ACK corresponding to the TCP data packet back to the fixed host and transfers the TCP data packet to the wireless connection. The main advantage of this approach is that transmission errors over wireless links can be hidden from the TCP sender, which is in the wired part of the network, and TCP over the wireless link can be optimized independent of the wired connection. However, it has the disadvantage that it violates the end-to-end semantics of TCP. For example, a TCP ACK may be delivered to a TCP sender before the associated TCP data is actually delivered to the TCP recipient.

In another study [1], a TCP snoop protocol makes changes to the network layer by introducing a new module called a snoop agent at an AP. This agent buffers TCP data packets

[1]The TCP congestion window ($cwnd$) is a state variable that limits the amount of data that a TCP sender can send.

which are destined for mobile hosts and which have not yet been acknowledged by them. When the loss of a packet is detected by the arrival of duplicated TCP ACKs, or by TCP timeout, the agent performs local retransmission across the wireless link. This scheme also prevents the TCP sender from invoking unnecessary fast-retransmission by dropping the duplicated TCP ACKs at the AP. It improves on the split-connection approach by preserving the end-to-end semantics and using the soft-state at an AP.

B. Improving TCP performance over 802.11-based WLANs

DCF+ is a recent approach to enhancing TCP performance over wireless LANs. DCF+ is based on the WLAN MAC layer [4], and extends IEEE 802.11 DCF while remaining compatible with it. DCF+ assumes that a recipient has a data frame ready to be transmitted back to the sender. Standard DCF requires that a source station transmitting a data frame must receive a MAC ACK frame from the destination station. DCF+ makes the ACK frame act as an RTS frame sent by the destination. Then, the source replies with a CTS and the destination can immediately transfer data frames which are ready to send to the source station. This procedure ensures that the next transmission in reverse direction follows without contention, and hence reduces the time required for successful transmission.

When several mobile hosts upload packets using TCP, flows with only a small number of packets in flight (e.g. newly started TCP flows) are more susceptible to timeouts than flows with a large number of packets in flight. Small flows can therefore be starved for a long time due to timeouts induced by packet losses. This unfairness of uploading TCP flows can be resolved by using additional flexibilities of the IEEE 802.11e MAC, which allows the values of interframe space (IFS) called Arbitrary IFS (AIFS) and CW_{min} to be set on a per-class basis for each mobile host. Mobiles with small values of IFS and CW_{min} have more opportunities for packet transmissions. By reducing IFS and CW_{min} for an AP, and increasing them for other mobile hosts, the AP gets a more equitable share of opportunities to access the wireless channel [5].

III. The Proposed Algorithm

A. Basic Idea

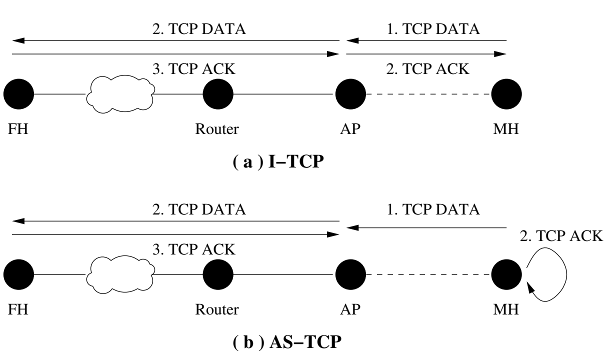


Fig. 1. TCP connection.

In an IEEE 802.11-based wireless LAN, both TCP and the IEEE 802.11 DCF support their own independent retransmission mechanism. TCP retransmits data packets when packet losses are detected by duplicated ACKs or TCP timeout. A similar retransmission function is implemented in the IEEE 802.11 DCF. In the absence of MAC ACKs, a data packet is retransmitted either 4 or 7 times, depending on its length. Additionally, the binary exponential backoff mechanism in 802.11 DCF gives it a similar role to that of TCP in congestion control as well as in collision avoidance. When there is network congestion in the wireless link, mobile hosts are more likely to attempt to transmit data packets at the same time. If two or more mobile hosts pick up the same slot in their own contention window, data packets will collide, increasing the size of each contention window. The larger the contention window, the more time mobile hosts have to wait before attempting to retransmit the data packets.

This functional duplication between TCP and DCF motivates our work to eliminate TCP ACK signaling in wireless areas. By leaving the reliability of services in wireless areas as a matter to be handled by the IEEE 802.11 MAC, it is possible to eliminate the need for acknowledgments by TCP receivers. It is this ACK signaling that leads to the lack of aggressiveness by TCP in wireless LANs, so that replacing actual TCP ACK signaling with an emulation might be expected to improve the performance of TCP flows. As described in Section II-B, previous work on TCP in wireless LANs has focused on downloading TCP flows. We consider uploading TCP flows as well.

Our approach involves changing the behavior of a mobile host with TCP flows. Instead of waiting for TCP ACKs from a recipient, a mobile that sent data packets now regards MAC ACKs as indicating successful reception of data packets, and TCP ACKs can be eliminated from the wireless link. To extend this concept to a more general case, including wired network sections, we extend the idea of I-TCP, which is to divide a TCP connection into two distinct connections : wired and wireless. Figure 1 summarizes our proposed protocol.

We expect AS-TCP to be particularly effective when 802.11 PSM is activated. An AP that uses PSM should buffer every packet including TCP ACKs destined for mobile hosts, and only forwards buffered packets at the next beacon interval. This delay to TCP ACKs inevitably results in the increase of round trip time (RTT) in TCP flows. In our scheme, however, the performance of TCP flows is not restricted by PSM since mobiles do not have to receive TCP ACKs over wireless links.

B. Protocol description

User datagram protocol (UDP) has no mechanism such as flow or congestion control within itself, so splitting UDP connections makes no difference. TCP is responsible for flow and congestion control. Therefore, a split-connection approach can improve the performance of TCP flows by optimizing the wired and wireless sections independently on the basis of their distinct characteristics. In our split-connection approach, a new sublayer called the TCP-aware sublayer (TAS) is inserted

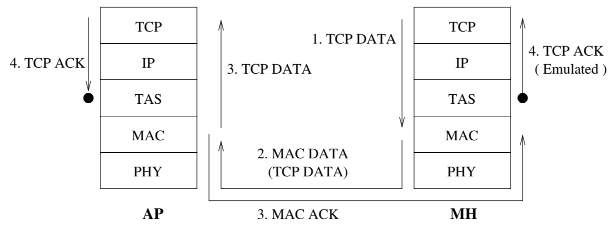


Fig. 2. Modified protocol stack.

between the MAC layer and the IP layer to emulate TCP ACKs with MAC ACKs.

The beginning and end stages of AS-TCP, in which TCP connections are established and closed, are unchanged from normal TCP. To establish a connection, TCP uses a three-way handshake exchanging SYN segment which carries an initial sequence number, and its ACK segment. Since sequence numbers are sent and acknowledged during the handshake, both sides need to agree on the initial sequence number. The TCP ACK specifies the sequence number of the next octet that the receiver expects to receive, which is usually one greater than the previously received one. To create a TCP ACK packet from a MAC ACK frame, TAS must know the sequence number. By snooping on the handshake of the upper layer, TAS can obtain and maintain the numbers.

When a mobile host transfers a data packet, it passes through TAS, as shown in Figure 2. If a MAC data frame is transferred successfully and a MAC ACK frame reception is reported to the TAS, it will transform the MAC ACK frame into a TCP ACK packet using the sequence number obtained during the handshake. Then it sends the TCP ACK packet to the upper layer. TCP receives the TCP ACK and transmits next TCP data packet. Meanwhile, the AP which received TCP data from the mobile host generates a TCP ACK to inform the mobile host of a successful reception. That ACK message will be discarded at the TAS of the AP. The AP will then establish a new connection with the TCP sink, and transfer TCP data using the normal TCP protocol. When MAC transmission fails, the 802.11 increments the retry counter associated with that frame. If the retry limit, usually 7, is reached, then the frame is discarded, and its loss is reported to the upper-layer protocol. This report is intercepted by the TAS of the mobile host, and the TAS does not generate a TCP ACK. After a timeout period, the TCP sender will notice a transmission failure, and initiate retransmission of the data packet.

C. Implementation issues

1) Deployment issue: As described in Section III-B, both AP and mobile host need to be modified to deploy AS-TCP. To make deployment easier, we could install the whole AS-TCP facility in the mobiles without changing any AP. In that case, the mobile host would have to drop TCP ACKs sent by an AP, which would waste the wireless channel on meaningless TCP ACKs. In general, an AP is a low-priced product and it is sometimes hard or impossible to upgrade its software or change its operation. But a router usually can be loaded with a new version of its software. Therefore, to make a full use of existing equipment, such as APs, for the deployment of AS-TCP, it is better to choose the nearest router which is directly connected to an AP in the wired part of the network as the split-point. Moving the split point from an AP to the nearest router hardly affects the operation of AS-TCP because the link between an AP and a router has sufficient bandwidth and incurs negligible delay.

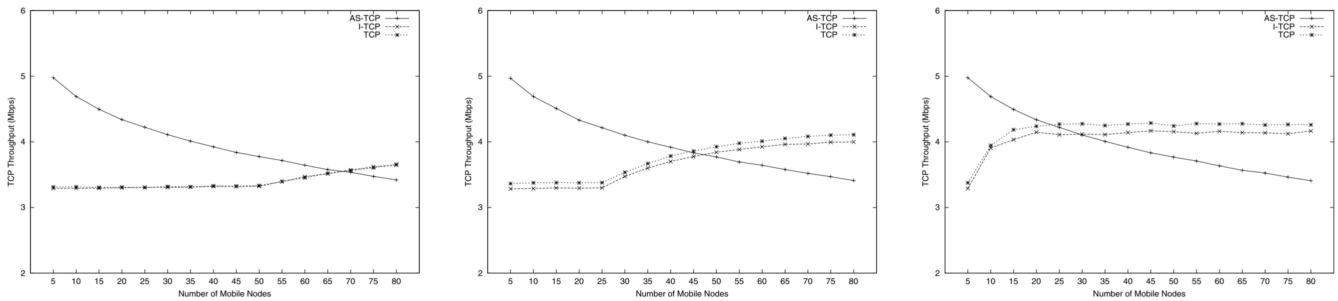
2) Flow-rate issue: In a split-TCP approach, the wireless part of a TCP connection over a wireless LAN is one-hop distant, with a very low RTT, so a high TCP throughput can be achieved. But a wired part of the connection usually has a lower throughput, due to a higher RTT and a long path. A high RTT stops TCP from transmitting so much data, owing to its flow control mechanism. Therefore, the throughput of the wireless part is higher than that of the wired part. This situation may cause packet, to be dropped at a split point, due to the limited length of the interface queue at an AP. To balance the data rates of wired and wireless parts in an end-to-end TCP connection, an AP intentionally fails to acknowledge receipt of a packet if the interface queue at the AP is long. After several retries have failed at the MAC layer, the TCP packet contained in the MAC data frame is also dropped, and the TCP congestion control mechanism is invoked. At that point, the wireless part of the TCP connection may reduce its sending rate.

IV. Performance Evaluation

Using the NS2 simulator, we have evaluated our scheme under various scenarios, varying the maximum interface queue length (IFQLEN) at an AP, the packet error rate and the power saving mode. Three performance metrics, aggregated throughput, fairness index and mean delay, were measured and analyzed for three algorithms: normal TCP, I-TCP and AS-TCP. The network topology tested in these simulations is depicted in Figure 1, and consists of n mobile hosts, an AP, a router connected to the AP and n fixed hosts; n varies from 5 to 80. Each mobile host generates an uploading TCP flow, with a packet size of 1000 bytes, which it sends to the corresponding fixed host in the wired part of the network, using an FTP session. The bandwidth of every wired link is set to 100Mbps and the wireless link capacity between the mobiles and the AP is set to 11Mbps. The link delay between each fixed host and the router is set to 10ms, and that between the router and the AP is set to 5ms.

A. TCP throughput with varying IFQLEN

Figure 3 compares the aggregated throughput of AS-TCP with that of two alternatives, normal TCP and I-TCP, as the number of mobile hosts varies. In these simulations, different interface queue lengths on the AP are considered. Three graphs show that AS-TCP outperforms the alternatives when the number of flows is small or moderate. The mobile hosts can send as many packets as possible since they do not have to wait for TCP ACKs coming from or via the AP before transmitting the next packet. The aggregate throughput of the



(a) IFQLEN=1000 (b) IFQLEN=500 (c) IFQLEN=100

Fig. 3. TCP throughput w.r.t. number of mobile hosts.

alternative protocols grows a little with the number of mobile hosts does, but the aggregate throughput is lower than that of AS-TCP. That is because that TCP over a wireless LAN is not an aggressive protocol, as mentioned earlier. Conversely, the aggregated throughput of AS-TCP decreases with the number of mobile hosts. The reason is that AS-TCP is an aggressive wireless LAN protocol, so more mobile hosts participate in contention as their number increases. Packet collision occurs frequently, resulting in low utilization of the wireless channel. If an additional mechanism, such as S-EDCF [10], which aims to reach an optimal contention window as quickly as possible, were used in conjunction with AS-TCP, we should be able to maintain its aggregate throughput.

B. Fairness index with varying IFQLEN

Figure 4 compares the fairness index of our scheme with that of two alternatives, normal TCP and I-TCP, as the number of mobile hosts varies. In these simulations, different interface queue lengths at the AP are also considered. The fairness index [9] is a normalized measure that ranges between zero and one, and is evaluated as follows,

$$fairness\ index \ = \ \frac{\left(\sum_f T_f/\phi_f\right)^2}{number\ of\ flows \times \sum_f \left(T_f/\phi_f\right)^2},$$

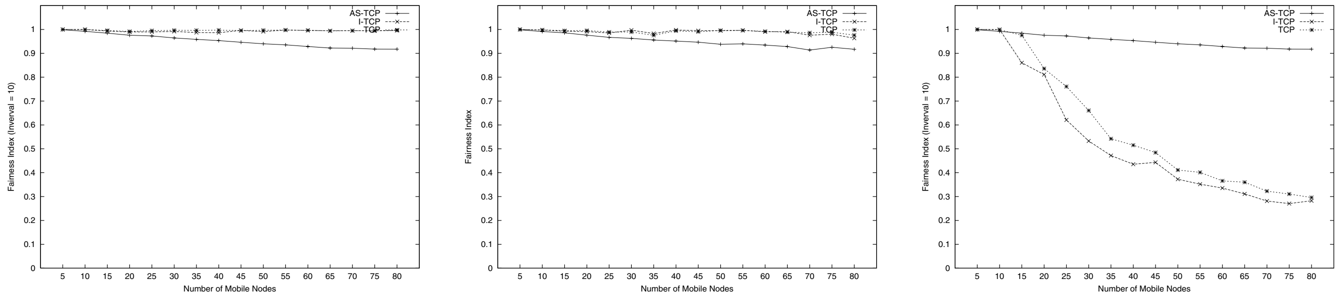
where T_f denotes the throughput of flow f, and ϕ_f denotes its weight. The fairness index indicates how fairly resources are allocated among the given contenders. When the interface queue length of an AP is large enough (i.e., larger than 500) all the schemes show an almost fair resource allocation, regardless of the number of mobile hosts. The fairness index of our scheme is a little smaller than that of the alternatives, but their difference is nearly negligible. When the queue at the AP is small (i.e., 100) the fairness of the aggregate throughput of both alternatives becomes worse as the number of mobile hosts increases. Our scheme almost maintains its fairness despite the inadequate queue length. That is because the two alternatives drop TCP ACKs from the queue at the AP, which results in unfairness among TCP flows, while with our scheme AS-TCP on a mobile only receives emulated TCP-ACKs if packet transmission is successful at the MAC layer. In other words, the fairness index of AS-TCP is not influenced by packet that are lost due to the elimination of real TCP ACKs at the AP.

C. TCP throughput with varying packet error rate

Compared with the two alternatives, AS-TCP performs retransmission only at the MAC layer, not at the transport layer. This might reduce the reliability of AS-TCP. To observe how packet errors affect AS-TCP, we considered an environment which is prone to packet errors. Figure 5 compares the aggregated TCP throughput of our scheme with that of the two alternatives for different packet error rates on the wireless channel. In this simulation, the interface queue length at the AP was set to 1000. As shown in Figure 5, the aggregated TCP throughput of each scheme decreases as the packet error rate increases. However, AS-TCP still shows a substantially higher aggregate TCP throughput than the two alternatives for all three packet error rates.

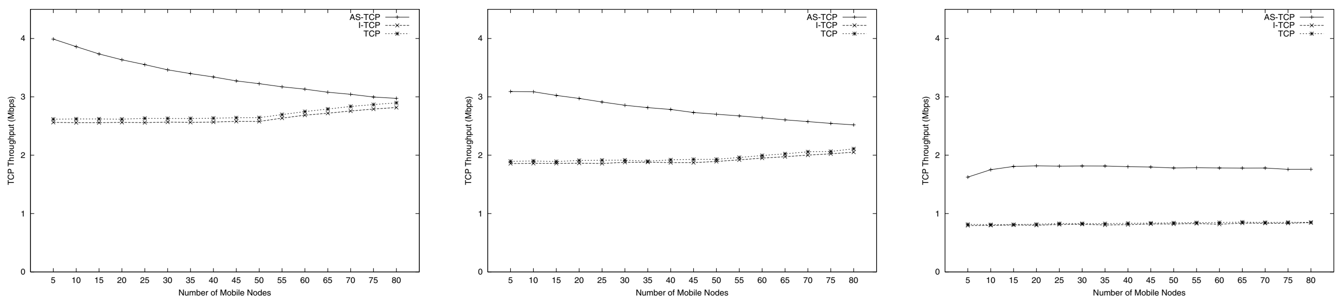
D. Mean delay in power-saving mode

It is well-known that IEEE 802.11 PSM suffers from poor performance. Moreover, when TCP is used over IEEE 802.11 PSM, TCP ACKs intended for mobile hosts are buffered at an AP until the next beacon. This latency increases the RTT for the short-lived normal TCP connections, resulting in severe performance degradation. To evaluate the performance of our scheme under IEEE 802.11 PSM, individual five mobile hosts opened a TCP connection to the corresponding fixed host, using FTP, and sent a file. We measured how long it took to transmit the whole file, while varying the file size from 10 to 100KB. Figure 6 compares the latency of our scheme with that of the two alternatives under different PSM conditions: no PSM, PSM with a 50ms beacon interval, and PSM with a 200ms beacon interval. As expected, using IEEE 802.11 PSM, the latency of the two alternatives deteriorates as the beacon interval gets longer. But, with AS-TCP, the delays are almost constant, regardless of PSM. That is because TCP ACKs do not have to be buffered at an AP and wait until the next beacon; instead, emulated TCP ACKs are returned immediately after successful transmission at the MAC layer.



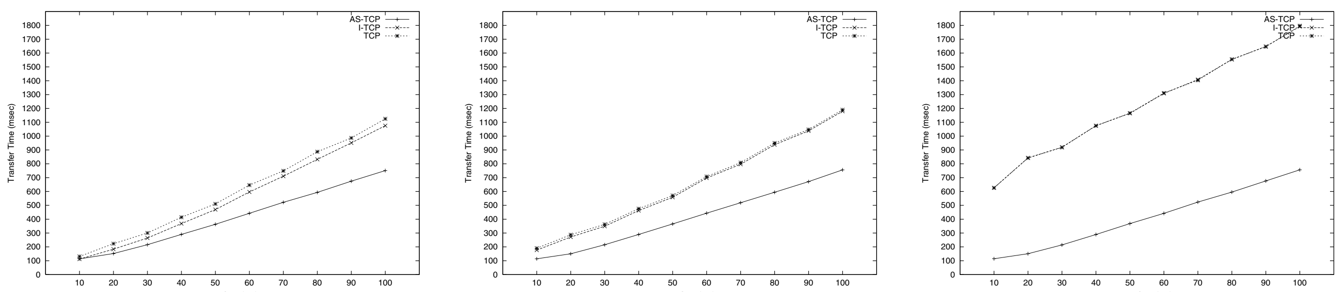
(a) IFQLEN=1000 (b) IFQLEN=500 (c) IFQLEN=100

Fig. 4. Fairness index w.r.t. number of mobile hosts.



(a) Packet error rate=0.1 (b) Packet error rate=0.2 (c) Packet error rate=0.4

Fig. 5. TCP Throughput w.r.t. number of mobile hosts.



(a) No power-saving mode (b) Power-saving mode (50ms) (c) Power-saving mode (200ms)

Fig. 6. Transfer time w.r.t. file size.

E. Remarks on normal TCP and I-TCP

In all these simulations, normal TCP and I-TCP show similar results in terms of aggregate throughput, fairness and latency. This is because I-TCP focuses on improving the performance of *downloading* TCP flows from a fixed host to a mobile host in *general* wireless networks. Depending on wireless channel conditions, there may be frequent frame errors in the wireless part of the network. Therefore, a downloading TCP flow can derive a performance gain from I-TCP by avoiding wasteful TCP packet retransmissions in a wired part of the network, with its low bandwidth and long latency. TCP packets lost in the wireless part are locally retransmitted from a split point such as a base-station, thus preventing the end-to-end throughput between a fixed and a mobile host from being degraded. However, in case of a wireless LAN, if a MAC frame containing a TCP packet transmission fails in the wireless part of the network, the lost frame is retransmitted at the MAC layer. That is, in a wireless LAN because packet loss is an infrequent event in view of TCP, the situation is unchanged from normal TCP that has a nearly lossless wireless part.

V. Conclusions

Improving the TCP performance in IEEE 802.11-based wireless LANs is an issue of current priority. TCP shows poor performance over wireless LANs because it is not aggressive in this context, although it behaves differently on other wireless networks. To solve this problem, we have proposed augmented split-TCP (AS-TCP), a new approach which eliminates real TCP ACK packets by exploiting the duplicated functionality of the MAC and transport layers. We have compared the performance of AS-TCP with normal TCP and I-TCP. Simulation results show that AS-TCP outperforms these two alternatives in terms of aggregated TCP throughput and fairness index, even in lossy environments. Additionally, when IEEE 802.11 PSM is enabled, AS-TCP can achieve reasonable latency, even ever short-lived connections.

REFERENCES

[1] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, vol. 1, pp. 469-481, December 1995.

[2] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Computer Communication Review*, Vol. 27, Issue 5, pp. 19-43, October 1997.

[3] A. Bakre and B.R. Badrinath, "I-TCP:indirect tcp for mobile hosts," Proc. *Distributed Computing Systems*, 1995.

[4] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LAN: analysis and enhancement," Proc. *INFOCOM*, 2002.

[5] D.J. Leithm and P. Clifford, "Using the 802.11e EDCF to achieve TCP upload fairness over WLAN links," *WIOPT*, 2005.

[6] S. Choi, K. Park, and C. Kim, "On the performance characteristics of WLANs: revisited," Proc. *ACM SIGMETRICS*, 2005.

[7] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," Proc. *Mobile computing and networking*, 2002.

[8] K. Ratnam and I. Matta, "WTCP: an efficient mechanism for improving TCP performance overwireless links," Proc. *ISCC*, 1998.

[9] R. Jain, G. Babic, B. Nagendra, and C. Lam, "Fairness, call establishment latency and other performance metrics," Tech. Rep. ATM Forum/96-1173, ATM Forum Document, August 1996.

[10] J. Ryu, Y. Seok, Y. Choi, T. Kwon, and J.-M. Bonnin, "S-EDCF: EDCF based on superslot and pseudo collision," ACM *MobiCom* Student Poster Session, 2005.