

# Fair and Efficient TCP Access in IEEE 802.11 WLANs<sup>†</sup>

Feyza Keceli, Inanc Inan, and Ender Ayanoglu

Center for Pervasive Communications and Computing, Department of Electrical Engineering and Computer Science  
The Henry Samueli School of Engineering, University of California, Irvine  
Email: {fkeceli, iinan, ayanoglu}@uci.edu

**Abstract**— We illustrate the transport layer unfairness problem in the IEEE 802.11 Wireless Local Area Networks (WLANs). We propose a novel and simple analytical model to calculate the per-flow congestion window limit that provides fair TCP access in a wired/wireless scenario where the wireless hop is an 802.11 link. The proposed analysis is unique in that it considers the effects of varying number of uplink and downlink TCP flows, varying Round Trip Times (RTTs) of TCP connections, and the use of delayed TCP Acknowledgment (ACK) mechanism. The comparison with simulation results validates the accuracy of the analytical estimations on the congestion window limit. When the TCP connections use the congestion window limits calculated from the proposed model, not only fair access can be provisioned, but also the channel can be utilized efficiently in a wide range of scenarios.

## I. INTRODUCTION

An IEEE 802.11 Wireless Local Area Network (WLAN) is built around a Basic Service Set (BSS) [1]. In the common deployment, the wireless stations communicate with an Access Point (AP) which provides the connection of the WLAN to the wired network. This constitutes the infrastructure BSS.

In the Medium Access Control (MAC) layer, the IEEE 802.11 WLAN employs a mandatory contention-based channel access function called the Distributed Coordination Function (DCF). The DCF adopts Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and binary exponential backoff. In DCF, the wireless stations, having all equal contention parameters, have equal opportunity to access the transmission medium. Over a sufficiently long interval, this results in station-based fair access which is also referred as MAC layer fair access.

On the other hand, per-station MAC layer fairness does not translate into achieving per-flow transport layer fairness [2]. In DCF, the AP has the same access priority with the wireless stations. Therefore, an approximately equal bandwidth that an uplink 802.11 station may get is shared among all downlink traffic in the AP. This results in a considerable asymmetry between per-flow uplink and downlink bandwidth in a typical 802.11 WLAN scenario (where the downlink traffic load is usually much larger than the uplink traffic load). The effects

of this asymmetry can be detrimental in the case it couples with the bi-directional and reliable data delivery method of Transmission Control Protocol (TCP). As will be described in more detail in Section II, unfair bandwidth allocation is observed between not only uplink and downlink TCP flows but also individual uplink TCP flows.

The TCP uplink and downlink asymmetry problem in the IEEE 802.11 infrastructure BSS is first studied in [2]. A maximum TCP congestion window setting regarding the MAC buffer size of AP is proposed. MAC parameter differentiation [3], [4], parameter adaptation [5], backoff enhancements [6], [7], ACK filtering and congestion control [8], per-flow queueing [9], [10], and rate limitation at the AP [11] are among several techniques that are proposed to overcome TCP unfairness problem in the WLAN.

In this paper, we propose a simple analytical method to calculate the maximum TCP congestion window size that achieves fair TCP access in the IEEE 802.11 WLAN. The proposed analysis shows that this maximum value can be approximated by a simple linear function of the bandwidth of the 802.11 WLAN, the wired link delay of the TCP connection, the MAC buffer size of the AP, and the number of TCP data packets each TCP acknowledgment (ACK) packet acknowledges. The proposed analysis is generic so that it considers varying number of uplink and downlink TCP flows, the use of delayed TCP ACK algorithm, and varying Round Trip Times (RTTs) among TCP connections. Via simulations, we show that the analytically calculated maximum congestion window setting provides fair access and high channel utilization. As we will also describe, the proposed analysis framework can also be used to calculate the required size of the AP buffer for fair TCP access provisioning given the limit on the TCP congestion window size.

As for the organization of this paper, we illustrate the TCP unfairness problem in Section II. Section III describes the proposed analytical method to calculate the maximum TCP congestion window size that achieves per-flow fair access in the WLAN. The performance evaluation is the topic of Section IV. We provide our concluding remarks in Section V.

## II. TCP UNFAIRNESS IN THE 802.11 WLAN

In the 802.11 WLAN, a bandwidth asymmetry exists between contending upload and download flows. This is due to

<sup>†</sup> This work is supported by the Center for Pervasive Communications and Computing, and by National Science Foundation under Grant No. 0434928. Any opinions, findings, and conclusions or recommendations expressed in this material are those of authors and do not necessarily reflect the view of the National Science Foundation.

the fact that the MAC layer contention parameters are all equal for the AP and the stations. If  $K$  stations and an AP are always contending for the access to the wireless channel, each host ends up having approximately  $1/(K + 1)$  share of the total transmit opportunities over a long time interval. This results in  $K/(K + 1)$  of the transmissions being in the uplink, while only  $1/(K + 1)$  of the transmissions belong to the downlink flows.

The TCP receiver returns TCP ACK packets to the TCP transmitter in order to confirm the successful reception of the data packets. In the case of multiple uplink and downlink flows in the WLAN, returning TCP ACKs of upstream TCP data are queued at the AP together with the downstream TCP. When the bandwidth asymmetry in the forward and reverse path builds up the queue in the AP, the dropped packets impair the TCP flow and congestion control mechanisms which assume equal transmission rate both in the forward and reverse path [12].

Any TCP data packet that is dropped from the AP buffer is retransmitted by the TCP sender following a timeout or the reception of duplicate ACKs. Conversely, any received TCP ACK can *cumulatively* acknowledge all the data packets sent before the data packet for which the ACK is intended for, i.e., a consequent TCP ACK can compensate for a dropped TCP ACK. When the packet loss is severe in the AP buffer, the downstream flows will experience frequent timeouts thus congestion window size decreases, resulting in significantly low throughput. On the other hand, due to the cumulative property of the TCP ACK mechanism, upstream flows with large congestion windows will not experience such frequent timeouts. In the latter case, it is a low probability that many consecutive TCP ACK losses occur for the same flow. Conversely, the upstream flows with small congestion windows (fewer packets currently on flight) may also experience timeouts and decrease their congestion windows even more. Therefore, a number of upstream flows may starve in terms of throughput while some other upstream flows enjoy a high throughput. In summary, the uplink/downlink bandwidth asymmetry creates a congestion at the AP buffer which results in unfair TCP access.

Fig. 1 shows the average throughput observed by each TCP flow for a scenario of 30 stations and an AP in an ns-2 simulation [13]. Each station runs an FTP session over TCP with a peer station located in the wired network. There are 10 uplink and 20 downlink FTP sessions. In Fig. 1, the flow indices from 1 to 10 denote the uplink TCP connections. The rest are the downlink connections. Each station has 802.11g physical layer with physical data rate set to 54 Mbps [14]. Other simulation parameters are as specified in Section IV. The results illustrate the unfairness in the throughput achieved by the uplink and downlink FTP flows. All of the downlink TCP flows starve in terms of throughput. Only 6 uplink connections can achieve high throughput, while 3 uplink connections are totally shut down.

### III. TCP FAIRNESS ANALYSIS

We consider a typical network topology where a TCP connection is initiated between a wireless station and a wired

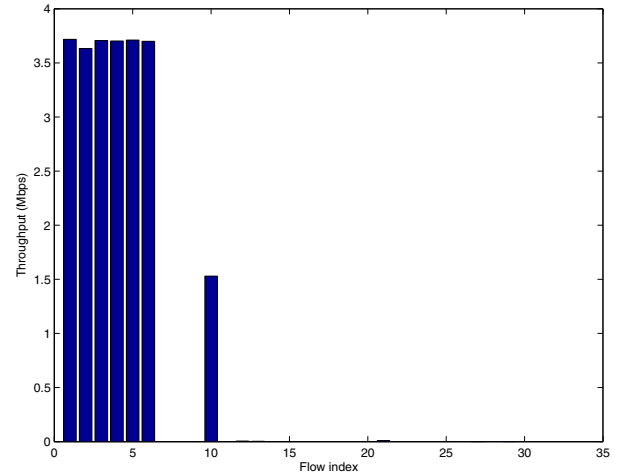


Fig. 1. Average throughput of 10 downlink and 20 uplink TCP connections. The flow indices from 1 to 10 denote the uplink connections. The rest are the downlink connections.

station either in the downlink or the uplink of the WLAN. The WLAN traffic is relayed to the wired network through the AP and vice versa. Let Round Trip Time (RTT) denote the average length of the interval from the time a TCP data packet is generated until the corresponding TCP ACK packet arrives. RTT is composed of three main components as follows.

- **Wired Link Delay ( $LD$ ):** The flow-specific average propagation delay of the packet between the AP and the wired node.
- **Queueing Delay ( $QD$ ):** The average delay experienced by a packet at the wireless station buffer until it reaches to the head of the queue. Note that due to the unequal traffic load at the AP and the stations,  $QD_{AP}$  and  $QD_{STA}$  may highly differ.
- **Wireless Medium Access Delay ( $AD$ ):** The average access delay experienced by a packet from the time it reaches to the head of the MAC queue until the transmission is completed successfully.

Then, RTT is calculated as follows.

$$RTT = 2 \cdot LD + QD_{AP} + QD_{STA} + AD_{AP} + AD_{STA} \quad (1)$$

Note that RTT is calculated as in (1) irrespective of the direction of the TCP connection. On the other hand, specific values of  $AD$  and  $QD$  depend on the packet size, the number of contending stations, etc. Therefore, RTT of an uplink connection may differ from RTT of a downlink connection.

As it is discussed in Section II, at high load, the AP can get only  $1/n$  share of the total transmissions over a long time interval which results in unfair access between the TCP connections. On the other hand, fair access can be achieved if the AP gets a higher share of the bandwidth such that this share prevents packet drops at the AP queue. Providing the AP a predetermined share of the total transmissions can be achieved by limiting the number of packets in flight (i.e., in nonsaturation\*). This approach simply translates to limiting the congestion window of each TCP flow. In the sequel, we

will calculate the maximum congestion window size of each TCP flow that prevents packet losses in the AP buffer.

For our analysis, let the AP and the stations use equal access parameters as suggested in [1]. Unless otherwise stated, let each TCP data packet be acknowledged by a TCP ACK packet. Moreover, define the cycle time as the duration in which an arbitrary tagged station successfully transmits one packet on average in the 802.11 WLAN [15], [16]. We claim that if the system is to be stabilized at a point such that no packet drops occur at the AP queue, then the following conditions should hold.

- *All the non-AP stations are in nonsaturated condition.*

Let's assume a station has  $X$  packets (TCP data or ACK) in its queue. A new packet is generated only if the station receives packets (TCP ACK or data) from the AP (as a result of ACK-oriented rate control of TCP). Let  $Y > 1$  users to be active. Every station (including the AP) sends one packet successfully every cycle time [15]. In the stable case, while the tagged station sends  $Y$  packets every  $Y$  cycle time, it receives only one packet. Note that the AP also sends  $Y$  packets every  $Y$  cycle time, but on the average,  $Y - 1$  of these packets are destined to the stations other than the tagged one. Therefore, after  $Y$  cycle times, the tagged stations queue size will drop down to  $X - Y + 1$ . Since  $Y > 1$ , the tagged stations queue will get empty eventually. A new packet will only be created when the AP sends a TCP packet to the tagged station which will be served before it receives another packet (on average). This proves that all the non-AP stations are in nonsaturated condition if no packet losses occur at the AP.

- *The AP contends with at most one station at a time on average.*

Following the previous claim, a non-AP station (which is nonsaturated) can have a packet ready for transmission if the AP has previously sent a packet to the station. There may be transient cases where the instantaneous number of active stations may become larger than 1. On the other hand, as we have previously shown, when  $Y > 1$ , the queue at any non-AP station eventually empties. If we assume the transient duration to be very short, the number actively contending stations on average is one. Therefore, at each DCF cycle time, the AP and a distinct station will transmit a packet successfully.

We define  $CT_{AP}$  as the duration of the average cycle time during which the AP sends an arbitrary packet (TCP data or ACK) successfully. We will derive  $CT_{AP}$  in Section III-B. Let the average duration between two successful packet transmissions of an arbitrary flow at the AP (or at the non-AP station) be  $CT_{flow}$ . Assuming there are  $n_{up}$  and  $n_{down}$  upload and download TCP connections respectively, we make the following approximation based on our claim that the AP

contends with one station on average

$$CT_{flow} \cong (n_{up} + n_{down}) \cdot CT_{AP}. \quad (2)$$

As it will be shown by comparing with simulation results in Section IV, the approximation in (2) leads to analytically correct results.

Then, the throughput of each station (whether it is running an uplink or a downlink TCP connection) is limited by  $1/CT_{flow}$  (in terms of packets per second). We can also write the TCP throughput using  $W_{lim}/RTT$ , where we define  $W_{lim}$  as the maximum TCP congestion window size for a TCP connection. We calculate  $W_{lim}$  such that if a flow uses a congestion window limit larger than  $W_{lim}$ , this results in packet losses at the AP buffer. As described in Section II, the drop of a TCP data packet (for downlink TCP connections) directly results in a timeout or the generation of duplicate TCP ACKs. On the other hand, due to the cumulative nature of TCP ACK packets, the uplink flows can still maintain their high throughput if the ACK drop ratio does not go over a threshold. As a result, unfair resource allocation in the WLAN is observed.

Following our previous claims,  $QD_{STA} = 0$  (the stations are nonsaturated),  $QD_{AP} = (BS_{AP} - 1) \cdot CT_{AP}$  (we consider the limiting case when the AP buffer is full, but no packet drop is observed), and  $AD_{AP} + AD_{STA} = CT_{AP}$  (the AP contends with one station on average), where  $BS_{AP}$  is the buffer size of the AP MAC queue. Using  $1/CT_{flow} = W_{lim}/RTT$ , we find

$$W_{lim} = \frac{2 \cdot LD}{CT_{flow}} + \frac{BS_{AP}}{n_{up} + n_{down}}. \quad (3)$$

Note that  $CT_{flow}$  is an indication of the bandwidth at the bottleneck (at the AP). If the data rate exceeds this bandwidth, the excess data will be queued at the AP, eventually overflowing the AP buffer. We calculate  $W_{lim}$  considering a full AP buffer, therefore,  $W_{lim}$  is the *maximum* congestion window limit for a TCP connection that prevents the packet drops at the AP queue of size  $BS_{AP}$ .

We can make following observations from (3).

- The first term is the effective number of packets that are in flight in the wired link for any flow, while the second term is the number of packets that are in the AP buffer for the same flow.
- The first term is a function of  $LD$ . As we will show, the  $W_{lim}$  calculated is resilient to varying  $LD$  among connections.

#### A. Delayed TCP Acknowledgements

In the delayed TCP ACK mechanism, the TCP receiver acknowledges every  $b$  TCP data packets ( $b > 1$ ). A typical value (widely used in practice) is  $b = 2$ .

The use of delayed TCP ACK mechanism changes the system dynamics. On the other hand, we still employ our assumption that the AP contends one station at a time on the average. As will be shown by comparison with simulation results in Section IV, this assumption leads to analytically correct results in the case of delayed TCP acknowledgments.

\* Saturation is the limit reached by the system when each station *always* has a packet to transmit. In nonsaturation, the (nonsaturated) stations experience idle times when they have no packet to send.

We update (2) and (3) accordingly for delayed TCP acknowledgments. Let the average duration between two successful packet transmissions of the flow at the non-AP station be  $CT_{flow,del}$  when delayed TCP acknowledgment mechanism is used.  $CT_{flow,del}$  is calculated considering that an ACK packet transmission at the AP corresponds to the generation of  $b$  data packets in the uplink.

$$CT_{flow,del} \cong \left(\frac{n_{up}}{b} + n_{down}\right) \cdot CT_{AP} \quad (4)$$

$$W_{lim} = \frac{2 \cdot LD}{CT_{flow,del}} + \frac{BS_{AP}}{n_{up}/b + n_{down}} \quad (5)$$

### B. Calculating $CT_{AP}$

A random access system such as 802.11 DCF exhibits cyclic behavior. As previously mentioned, a cycle time is defined as the duration in which an arbitrary tagged user successfully transmits one packet on average. Using the approach in [15], we can derive the explicit mathematical expression for the average DCF cycle time. Then, we can use the average cycle time to predict the average throughput and the average service time. The simple cycle time analysis assesses the saturation (asymptotic) performance of the DCF accurately (when each contending AC always has a packet in service). Due to space limitations, we do not present the derivation details of DCF cycle time in this paper and refer the reader to [15].

We are interested in the case when there are two active (saturated) stations (as the AP contends with one station at a time). The average cycle time in this scenario can easily be calculated using the model in [15]. In our case, the AP sends the TCP ACK packets of the uplink TCP connections and the TCP data packets of the downlink TCP connections which contend with the TCP ACK packets of the downlink TCP connections and the TCP data packets of the uplink connections that are generated at the stations. Note that the cycle time varies according to the packet size of contending stations. Then,

$$CT_{AP} = \sum_{p_1 \in S} \Pr(p_{AP} = p_1) \sum_{p_2 \in S} \Pr(p_{STA} = p_2) CT_{p_1, p_2} \quad (6)$$

where  $S = \{ACK, DATA\}$  is the set of different types of packets,  $\Pr(p_{AP} = p_1)$  is the probability that the AP is sending a packet of type  $p_1$ ,  $\Pr(p_{STA} = p_2)$  is the probability that the non-AP station is sending a packet of type  $p_2$ , and  $CT_{p_1, p_2}$  is the average cycle time when one station is using a packet of type  $p_1$  and the other is using a packet type of  $p_2$ . We differentiate between the data and the ACK packets because the size of the packets thus the cycle time duration depends on the packet type.

Using simple probability theory, we can calculate  $\Pr(p_{AP})$  and  $\Pr(p_{STA})$  as follows

$$\Pr(p_{AP} = p_1) = \begin{cases} \frac{n_{down}}{n_{up}/b + n_{down}}, & \text{if } p_1 = DATA \\ \frac{n_{up}/b}{n_{up}/b + n_{down}}, & \text{if } p_1 = ACK, \end{cases} \quad (7)$$

$$\Pr(p_{STA} = p_2) = \begin{cases} \frac{n_{up}}{n_{down}/b + n_{up}}, & \text{if } p_2 = DATA \\ \frac{n_{down}/b}{n_{down}/b + n_{up}}, & \text{if } p_2 = ACK. \end{cases} \quad (8)$$

### C. Implementation

We calculate the maximum flow-specific congestion window limit  $W_{lim}$  that achieves fair TCP access in a wired/wireless scenario where the wireless hop is an 802.11 link. A control block located at the AP can modify the receiver window field of the ACK packets that are all relayed through the AP with  $W_{lim}$ .

In order to calculate  $W_{lim}$ , the control block at the AP needs to estimate  $LD$  and  $b$ . The AP may distinguish among TCP connections via the IP addresses and the ports they use. The proposed control block located at the AP may run a simple exponential averaging algorithm to calculate the average time that passes between sending a data (ACK) packet into the wired link and receiving the ACK (data) packet which generated by the reception of the former packet (which is  $2 \cdot LD$ ). The TCP header of consecutive ACK packets may be parsed to figure out the value of  $b^\dagger$ .

The proposed analysis can also be valuable for deciding on the AP buffer size that provides fair TCP access. The 802.11 vendors may use the proposed method with statistics of TCP connections and WLAN traffic to decide on a *good* size of AP buffer to optimize the chip size.

$$BS_{AP} = (W_{lim} - \frac{2 \cdot LD}{CT_{flow}}) \cdot (n_{up}/b + n_{down}) \quad (9)$$

It is also worth to note that although the analytical calculation uses a simple cycle time method in calculating  $CT_{AP}$  and  $CT_{flow}$ , the AP may use a measurement-based technique rather than the model-based technique used in this paper.

## IV. PERFORMANCE EVALUATION

In this section, we validate the analytical results obtained from the proposed model via comparing them with the simulation results obtained from ns-2 [13].

We consider a network topology where each wireless station initiates a connection with a wired station and where the traffic is relayed to/from the wired network through the AP. The TCP traffic uses a File Transfer Protocol (FTP) agent which models bulk data transfer. TCP NewReno with its default parameters in ns-2 is used. All the stations have 802.11g PHY [14] with 54 Mbps and 6 Mbps as the data and the basic rate respectively. The wired link data rate is 100 Mbps. The default DCF MAC parameters are used [1]. The packet size is 1500 bytes for all flows. The MAC buffer size at the stations and the AP is set to 100 packets.

We obtained  $W_{lim}$  via simulations in such a way that increasing the maximum TCP congestion window size of TCP connections by one results in a packet loss ratio larger than 1% at the AP buffer.

<sup>†</sup>Note that the proposed analysis can easily be extended for the case when the value of  $b$  differs among TCP connections.

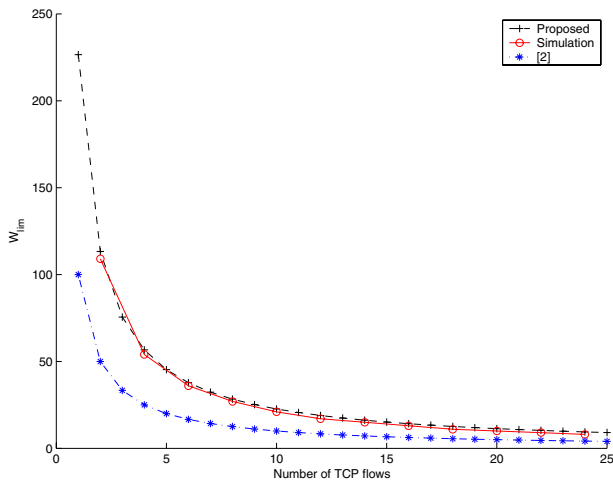


Fig. 2. Congestion window limits calculated by the proposed model, the simulation, and [2].

In the first set of experiments, we set the wired link delay of each connection to 50 ms. Each TCP data packet is acknowledged by an ACK packet ( $b = 1$ ). In Fig. 2, we compare the estimation of (3) on the congestion window limit with the values obtained from the simulation results and the proposed method of [2] for increasing number of TCP connections. The number of upload flows is equal to the number of download flows. As Fig. 2 implies, the analytical results for the proposed model and the simulation results are well in accordance. The analysis in [2] calculates the congestion window limit by  $BS_{AP}/(n_{up} + n_{down})$  and underestimates the actual TCP congestion window limit.

The total throughput of the system when the TCP connections employ analytically calculated congestion window limits in simulation for increasing number of TCP connections is shown in Fig. 3. As the comparison with [2] reveals, the congestion window limits calculated via our method result in approximately 35% - 50% higher channel utilization for the specific scenario. Although the corresponding results are not displayed, both methods achieve perfect fairness in terms of per-connection FTP throughput (Jain's fairness index [17],  $f > 0.9999$  where 1 shows perfect fairness).

In the second set of experiments, we consider a scenario where wired link delays ( $LD$ ) among TCP connections differ. First TCP connection has a 1 ms wired link delay, and  $n^{th}$  connection has  $n$  ms larger wired link delay than  $(n - 1)^{th}$  connection. We first calculate the congestion window limits via the proposed analysis in this paper and in [2] for varying number of uplink and downlink connections, and then employ the corresponding values in simulation. Fig. 4 shows the individual throughput for each TCP connection for the proposed model and [2] for 4 different scenarios. In Fig. 4, for any scenario, the first half are upload flows and the rest are download flows. As the results present, the maximum congestion window values calculated by the proposed model maintains fair access even in the case of varying wired link delays. On the other hand,

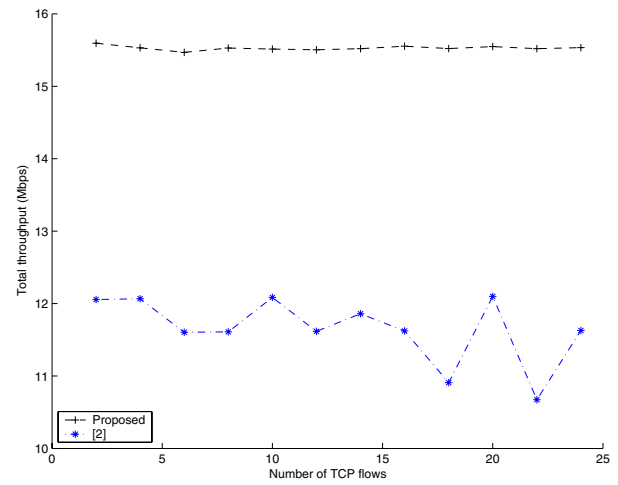


Fig. 3. Total throughput of the system when the TCP connections employ analytically calculated congestion window limits by the proposed model and [2].

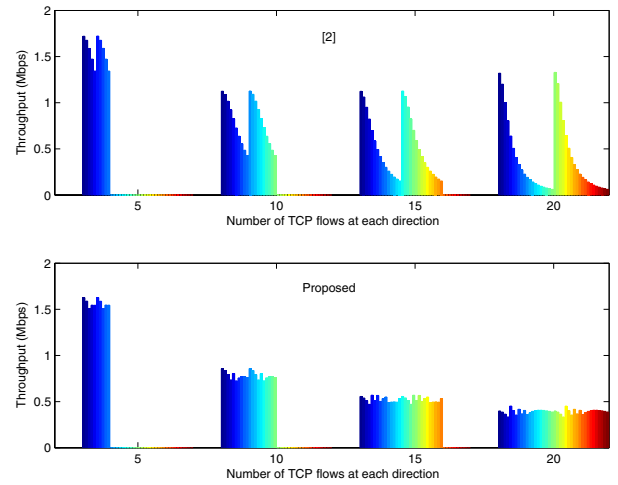


Fig. 4. Individual throughput of TCP connections when the TCP connections with different wired link delays employ the analytically calculated congestion window limits by the proposed model and [2].

the method proposed in [2] fails to do so.

In the third set of experiments, we consider a scenario where the TCP connections use the delayed ACK mechanism with  $b = 2$ . We consider 9 different scenarios where in each scenario the number of uplink and downlink TCP connections varies. In the first three scenarios, the number of downlink flows is set to 5 and the number of uplink flows is varied among 5, 10, and 15, respectively. Varying the number of uplink flows in the same range, the next three scenarios use 10, and the following three scenarios use 15 downlink flows. In Fig. 5, we compare the estimation of (5) on the congestion window limit with the values obtained from the simulation results and the proposed method of [2]. The analytical results for the proposed model and the simulation results are well in accordance. The total throughput of the system when the TCP connections employ the analytically calculated conges-

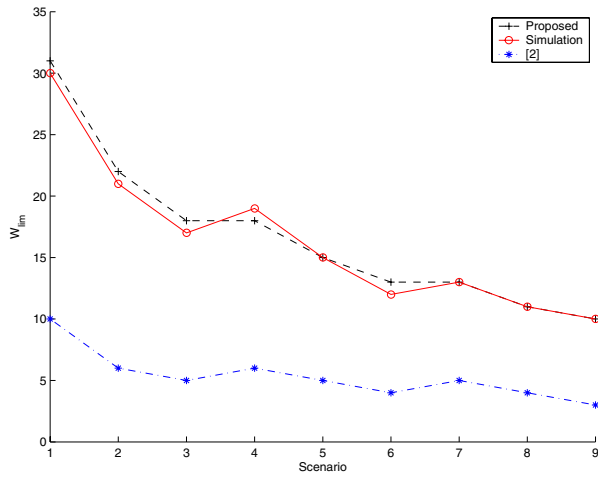


Fig. 5. Congestion window limits calculated by the proposed model, the simulation, and [2] when TCP connections employ delayed ACK mechanism with  $b = 2$ .

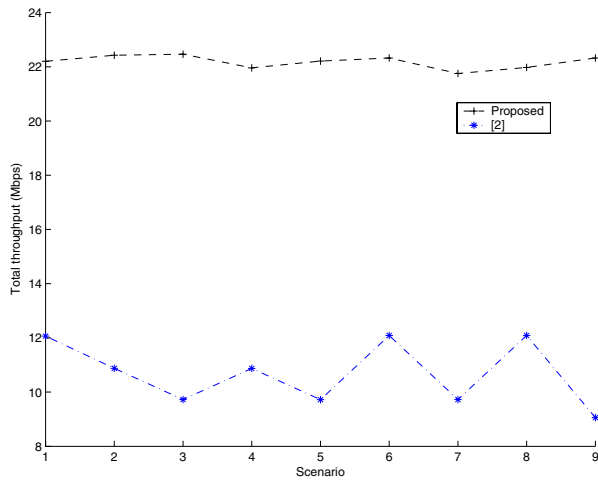


Fig. 6. Total throughput of the system when the TCP connections employ analytically calculated congestion window limits by the proposed model and [2] in the case delayed ACK mechanism is used ( $b = 2$ ).

tion window limits is shown in Fig. 6. As the comparison with [2] reveals, the congestion window limits calculated via our method result in approximately 90% - 105% higher channel utilization. Although the corresponding results are not presented, the congestion window limits calculated by both the proposed method and the method of [2] achieve perfectly fair resource allocation in terms of throughput (Jain's fairness index,  $f > 0.998$ ). On the other hand, the proposed method results in a significantly higher channel utilization.

## V. CONCLUSION

In this paper, we focused on TCP unfairness problem in an IEEE 802.11 infrastructure BSS. We have presented a novel and simple analytical model to calculate the TCP congestion window limit that provides fair TCP access in a wired/wireless scenario where the wireless hop is an 802.11 link. The key contribution of this study is that the proposed analytical

model considers varying wired link delays among connections, varying number of uplink and downlink connections, and the use of delayed ACK mechanism. A simple control block at the AP may calculate the congestion window limit for each individual TCP connection, and update the receiver window sizes accordingly. Via simulations, we have shown that the congestion window limits calculated via the proposed analysis provides fair TCP access and high channel utilization. The same model can also be used to decide on the required AP buffer size for fair TCP access given the maximum TCP congestion window sizes used by the connections. The cycle time analysis can be extended for IEEE 802.11e WLANs [18] as in [16], therefore the analysis in this paper can also be extended for the case when MAC parameter differentiation is used.

## REFERENCES

- [1] *IEEE Standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE 802.11 Std., 1999.
- [2] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN," in *Proc. IEEE Infocom '03*, April 2003.
- [3] C. Casetti and C. F. Chiasserini, "Improving Fairness and Throughput for Voice Traffic in 802.11e EDCA," in *Proc. IEEE PIMRC '04*, September 2004.
- [4] D. J. Leith, P. Clifford, D. Malone, and A. Ng, "TCP Fairness in 802.11e WLANs," *IEEE Commun. Lett.*, pp. 964–966, November 2005.
- [5] J. Freitag, N. L. S. da Fonseca, and J. F. de Rezende, "Tuning of 802.11e Network Parameters," *IEEE Commun. Lett.*, pp. 611–613, August 2006.
- [6] S. W. Kim, B.-S. Kim, and Y. Fang, "Downlink and Uplink Resource Allocation in IEEE 802.11 Wireless LANs," *IEEE Trans. Veh. Technol.*, pp. 320–327, January 2005.
- [7] J. Jeong, S. Choi, and C.-K. Kim, "Achieving Weighted Fairness between Uplink and Downlink in IEEE 802.11 DCF-based WLANs," in *Proc. IEEE QSHINE '05*, August 2005.
- [8] 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," in *Proc. IEEE ICC '07*, June 2007.
- [9] Y. Wu, Z. Niu, and J. Zheng, "Study of the TCP Upstream/Downstream Unfairness Issue with Per-flow Queueing over Infrastructure-mode WLANs," *Wireless Commun. and Mobile Comp.*, pp. 459–471, June 2005.
- [10] J. Ha and C.-H. Choi, "TCP Fairness for Uplink and Downlink Flows in WLANs," in *Proc. IEEE Globecom '06*, November 2006.
- [11] N. Blefari-Melazzi, A. Detti, I. Habib, A. Ordine, and S. Salsano, "TCP Fairness Issues in IEEE 802.11 Networks: Problem Analysis and Solutions Based on Rate Control," *IEEE Trans. Wireless Commun.*, pp. 1346–1355, April 2007.
- [12] H. Balakrishnan, V. Padmanabhan, and R. H. Katz, "The Effects of Asymmetry on TCP Performance," *ACM Baltzer Mobile Networks and Applications (MONET)*, 1999.
- [13] (2006) The Network Simulator, ns-2. [Online]. Available: <http://www.isi.edu/nsnam/ns>
- [14] *IEEE Standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz Band*, IEEE 802.11g Std., 2003.
- [15] K. Medepalli and F. A. Tobagi, "Throughput Analysis of IEEE 802.11 Wireless LANs using an Average Cycle Time Approach," in *Proc. IEEE Globecom '05*, November 2005.
- [16] I. Inan, F. Keceli, and E. Ayanoglu, "Performance Analysis of the IEEE 802.11e Enhanced Distributed Coordination Function using Cycle Time Approach," in *Proc. IEEE Globecom '07*, November 2007.
- [17] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley and Sons, 1991.
- [18] *IEEE Standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Medium access control (MAC) Quality of Service (QoS) Enhancements*, IEEE 802.11e Std., 2005.