

# Weighted Fair Uplink/Downlink Access Provisioning in IEEE 802.11e 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 present the unfairness problem between the uplink and the downlink flows in the IEEE 802.11e infrastructure Basic Service Set (BSS) when the default settings of the Enhanced Distributed Channel Access (EDCA) parameters are used. First, we propose an analytical model to calculate the EDCA parameter set that ideally achieves a predetermined utilization ratio between uplink and downlink flows. Next, we design a simple and practical model-assisted measurement-based dynamic parameter adaptation algorithm. In the design of the proposed algorithm, we also consider the interactions of the bi-directional communication structure of Transmission Control Protocol (TCP) with the Medium Access Control (MAC) layer algorithm. Via simulations, we show that our solution provides short- and long-term fair access for uplink and downlink TCP and User Datagram Protocol (UDP) flows with different wired link propagation delays in a wired-wireless heterogeneous scenario. In the meantime, the Quality-of-Service (QoS) requirements of coexisting real-time flows can also be maintained.

## I. INTRODUCTION

The IEEE 802.11 provides the commonly deployed standard for wireless local area networks (WLANs). The IEEE 802.11 standard [1] defines Distributed Coordination Function (DCF) as a contention-based Medium Access Control (MAC) mechanism. The 802.11e standard [2] updates the MAC layer of the former 802.11 standard for Quality-of-Service (QoS) provision. In particular, the Enhanced Distributed Channel Access (EDCA) function of 802.11e is a QoS enhancement of the DCF. The EDCA scheme (similarly to DCF) uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and slotted Binary Exponential Backoff (BEB) mechanism as the basic access method. The major enhancement to support QoS is that EDCA differentiates packets using different priorities and maps them to specific Access Categories (ACs) that use separate queues at a station. Each  $AC_i$  within a station ( $0 \leq i \leq 3$ ) contends for the channel independently of the others. Levels of services are provided through different assignments of the AC-specific EDCA parameters; Contention Window (CW) sizes, Arbitration Interframe Space (AIFS) values, and Transmit Opportunity (TXOP) limits.

<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 DCF and the EDCA are defined such that each station in a BSS uses the same contention parameter set. Therefore, fair access can be achieved in the MAC layer for all the contending stations in terms of the average number of granted access opportunities, over a sufficiently long interval. However, this does not translate into achieving a fair share of bandwidth between uplink and downlink flows in the 802.11e infrastructure BSS. An AC of the Access Point (AP) which serves all downlink flows has the same access priority with the same AC of the stations that serve uplink flows. Therefore, an approximately equal number of accesses that an uplink AC may get is shared among all downlink flows in the same AC of the AP. This leads to the uplink/downlink unfairness problem in the WLAN where each individual downlink flow gets comparably lower bandwidth than each individual uplink flow gets at high load.

The results may even be more catastrophic in the case of TCP flows. The TCP receiver returns TCP ACK packets to the TCP transmitter in order to confirm the successful reception of 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 [3].

TCP's timeout mechanism initiates a retransmission of a data packet if it has not been acknowledged during a *timeout* duration. When the packet loss is severe in the AP buffer, downstream flows will experience frequent timeouts resulting in significantly low throughput. On the other hand, any received TCP ACK can cumulatively acknowledge all the data packets sent before the data packet for which the ACK is intended to. Therefore, upstream flows with large congestion windows will probably not experience such frequent timeouts. Conversely, flows with small congestion window (fewer packets currently on flight) may experience frequent timeouts and decrease their congestion windows even more. This results in unfairness between the TCP upstream flows on top of the unfairness between the uplink and the downlink. The uplink/downlink unfairness problem of default EDCA algorithm will be demonstrated in Section III via simulations.

We now provide a brief literature overview on the unfairness problem discussed above. A TCP-specific transport layer so-



where  $x = \max(y \mid d_y = \max_z(d_z \mid d_z \leq n))$  which shows  $x$  is assigned to the largest index value within a set of TCs that have the largest AIFSN value which is smaller than or equal to  $n + AIFSN_{min}$ .

Given  $p_{c_i}$ , we can calculate the expected number of backoff slots  $E_i[t_{bo}]$  that  $TC_i$  waits before attempting a transmission. Let  $W_{i,k} = 2^{\min(k, m_i)}(CW_{i,min} + 1) - 1$  be the CW size of  $TC_i$  at backoff stage  $k$  where  $CW_{i,max} = 2^{m_i}(CW_{i,min} + 1) - 1$ ,  $0 \leq m_i < r_i$ . Note that, when the retry limit  $r_i$  is reached, any packet is discarded.

$$E_i[t_{bo}] = \frac{1}{1 - p_{c_i}^{r_i}} \sum_{k=1}^r p_{c_i}^{k-1} (1 - p_{c_i}) \frac{W_{i,k}}{2}. \quad (4)$$

Then, the transmission probability of  $TC_i$  is

$$\tau_i = \frac{1}{E_i[t_{bo}] + 1}. \quad (5)$$

The nonlinear system of equations (1)-(5) can be solved numerically to calculate average collision and transmission probabilities of each  $TC_i$  for an arbitrary setting of EDCA parameters. We provide the validation of the proposed analytical model in [17].

### B. Weighted Fairness between Uplink and Downlink Flows

Let  $\gamma_i$  be the probability that the transmitted packet belongs to an arbitrary user from  $TC_i$  given that the transmission is successful. Also, let  $p_{s_{i,n}}$  be the probability that a successfully transmitted packet at backoff slot  $n$  belongs to  $TC_i$ . Then,

$$\gamma_i = \frac{\sum_{n=d_i+1}^{W_{min}} b'_n p_{s_{i,n}}}{\sum_{n=d_i+1}^{W_{min}} (b'_n \sum_{\forall j} p_{s_{j,n}})}, \quad (6)$$

$$p_{s_{i,n}} = \begin{cases} \frac{N_i \tau_i}{(1 - \tau_i)} \prod_{i': d_{i'} \leq n-1} (1 - \tau_{i'})^{N_{i'}}, & \text{if } n \geq d_i + 1 \\ 0, & \text{if } n < d_i + 1. \end{cases} \quad (7)$$

Let  $U$  denote the utilization ratio between the downlink and the uplink transmissions of an AC. Let  $N_{TXOP,i}$  denote the maximum number of packets that can fit in one TXOP of  $TC_i$ . Then, for our running example with one AC,

$$U = \frac{\gamma_1 \cdot N_{TXOP,1}}{\gamma_0 \cdot N_{TXOP,0}}. \quad (8)$$

Without loss of generality, the EDCA parameters of the stations,  $AIFS_0$ ,  $CW_{min,0}$ , and  $N_{TXOP,0}$ , are fixed at predetermined values. Then, the EDCA parameters of the TC at the AP,  $AIFS_1$ ,  $CW_{min,1}$ , and  $N_{TXOP,1}$ , that achieve a required utilization ratio  $U_r$  can be calculated numerically using (6)-(8). The details on the implementation of the numerical solution are stated in a technical report [19].

### C. Dynamic Parameter Adaptation

Analytical models usually ignore non-ideal channel conditions and transport layer algorithm details to maintain simplicity. Including such details make finding the optimum EDCA setting for any scenario analytically hard and complex. Our

solution is finding an initial parameter set that would *ideally* satisfy  $U_{r,l}$ , for all  $l$ , and then update the set dynamically regarding simple network measurements. Note that we reintroduce the subscript  $l$  which denotes the AC index.

According to our proposed algorithm, the AP carries out the dynamic adaptation on the access parameters of each AC every  $\beta$  beacon intervals which is called an *adaptation interval* in the sequel. If a new flow starting transmission or an existing flow becoming inactive is detected at the last adaptation interval, the algorithm decides on new EDCA parameters using the proposed analytical calculation. Otherwise, fine tuning on the  $CW$  and the  $TXOP$  values of the  $AC_l$  at the AP is carried out to make measured utilization ratio  $U_{m,l}$  as close to the required downlink/uplink access ratio  $U_{r,l}$ .

We use a simple algorithm to estimate the number of active flows. The AP counts the number of unique source and destination MAC addresses observed from incoming frames. Let  $n_{u,l}$  and  $n_{d,l}$  denote the number of uplink and downlink flows labeled as active for  $AC_l$ . If the AP receives an  $AC_l$  packet with the corresponding MAC address not on its list, it adds the new MAC address to the list and increments  $n_{u,l}$  or  $n_{d,l}$ . If the AP does not receive any  $AC_l$  packet with the corresponding MAC address during the last adaptation interval, it deletes the MAC address from the list and decrements  $n_{u,l}$  or  $n_{d,l}$ . Then, we define the required utilization ratio as

$$U_{r,l} = \frac{n_{d,l}}{n_{u,l}}. \quad (9)$$

Note that we define  $U_r$  regarding the downlink and the uplink load in terms of the total number of flows. On the other hand,  $U_r$  can be assigned in many other ways. The value of  $U_r$  depends on how the network manager wants to define the weight of uplink and downlink flows within an AC. The adaptation in the sequel is independent of how  $U_r$  is assigned.

If  $U_{r,l}$  has been changed during the last adaptation interval, EDCA parameters are analytically calculated for  $U_l = U_{r,l}$  as previously described and the fine tuning phase is skipped. Otherwise, fine tuning on  $CW_{min}$  and  $TXOP$  is performed as follows. Every adaptation interval, the AP measures the number of successful uplink and downlink transmissions,  $n_{t_{u,l}}$  and  $n_{t_{d,l}}$ , respectively, where  $n_{t_{d,l}}/n_{t_{u,l}}$  is the measured utilization ratio  $U_{m,l}$  in the last adaptation interval. If  $U_{m,l} < (1 - \alpha) \cdot U_{r,l}$ , then  $CW_{min,1,l}$  is decremented (where  $0 \leq \alpha \leq 1$ ). Similarly, if  $U_{m,l} > (1 + \alpha) \cdot U_{r,l}$ , then  $CW_{min,1,l}$  is incremented. Otherwise, no action is taken. Note that using steps equal in value to 1 in the  $CW_{min}$  adaptation is sufficient since the analytical calculation already provides a good initial guess on the access parameters. If the controller detects that the prioritization among ACs cannot be maintained with the new  $CW_{min,1,l}$  assignment,  $N_{TXOP,1,l}$  (or  $CW_{min,0,l}$ ) is doubled, and a new  $CW_{min,1,l}$  is calculated analytically. The AP announces the new parameter settings for the stations via the beacon packet at the next beacon interval.

#### D. Interactions of the Transport Layer Protocols and the 802.11 Medium Access Control Layer

The algorithm proposed in Section II-C is directly applicable for weighted UDP fair access provisioning, while weighted TCP fairness provisioning requires a small modification in channel utilization measurements. The AP counts only the number of successful TCP data packets (excluding TCP ACKs). The definitions of  $n_d$  and  $n_u$  are updated accordingly. Differentiating TCP data packets from TCP ACK packets can be performed via a simple packet size comparison.

*Extra Prioritized TCP Downlink Access:* We also propose an alternative simple and practical solution for TCP that achieves  $U$  in (9). The key observation is that, if we assume there are no packet losses in TCP connections (infinitely large interface buffers or TCP congestion window sizes are set such that the total number of TCP packets on flight cannot be larger than the AP buffer size [4]), per-flow fair access can be achieved for all TCP flows in the 802.11 WLAN irrespective of the EDCA parameter selection. Our alternative solution is giving the AP extra priority in channel access for ACs that carry TCP traffic, so that the corresponding AC queues of the AP always remain nonsaturated. The reasoning behind this novel idea is two-fold; *i)* this avoids the TCP packet drops at the AP which is the main cause for unfairness, and *ii)* such an approach makes all of the non-AP stations saturated (since TCP is bi-directional), and in saturation, 802.11 MAC is fair to all competing stations (thus all stations will have equal number of transmit accesses). Note that although the non-AP stations are saturated in such a case, no buffer overflow is actually observed due to our practical assumption that the TCP senders and receivers set their TCP congestion window sizes regarding their available buffer space.

Note that there is an implicit assumption of TCP agents generating an ACK packet for each received data packet in our alternative solution. When the delayed TCP ACK mechanism is used (when every  $b^*$  TCP data gets acknowledged with one TCP ACK), the alternative solution prioritizes the downlink flows over uplink flows. Since all the stations are saturated, the upstream access is fairly distributed among upstream data link of uplink flows and the upstream TCP ACK link of downlink flows. On the other hand, every one upstream TCP ACK stands for  $b$  data packets. Since the AP queues are nonsaturated, the downlink TCP flows enjoy  $b$  times larger data packet transmissions than the uplink flows can transmit over a sufficiently long interval. In a practical WLAN scenario, it is highly probable that the downlink load will be significantly higher than the uplink load. Therefore, maintaining a downlink to uplink access ratio larger than 1 is practical.

In simulations, we observed that doubling the EDCA TXOP calculated via the analytical model is sufficient to provide the AP the extra priority. Although we used larger EDCA TXOP size, using smaller CW values is also applicable.

\* $b > 1$ , a typical value is 2.

### III. NUMERICAL AND SIMULATION RESULTS

We carried out simulations in ns-2 [20] in order to evaluate the performance of the proposed weighted fairness adaptation algorithm. For the simulations, we employ the IEEE 802.11e EDCA MAC simulation module for ns-2.28 [21].

We consider a network topology where each wireless station initiates a connection with a wired station where the traffic is relayed to/from the wired network through the AP. Unless otherwise stated, the wireless channel is assumed to be errorless.

The data connections use either UDP or TCP NewReno. The UDP traffic uses a Constant Bit Rate (CBR) application. The TCP traffic uses a File Transfer Protocol (FTP) agent which models bulk data transfer. Unless otherwise stated, TCP flows are considered to be lasting through the simulation duration and called long. On the other hand, in some experiments, short TCP flows are used which consist of 31 packets and leave the system after all the data is transferred. The default TCP NewReno parameters in ns-2 are used. At the stations, UDP flows are mapped to an AC with  $CW_{min} = 31$  and  $CW_{max} = 511$ . TCP flows use an AC with  $CW_{min} = 63$  and  $CW_{max} = 1023$ . For both ACs, AIFSN values are set to 2 and initial TXOP limits are 0. These access parameters are selected arbitrarily (the unfairness problem exists regardless of the selection if AP and the stations use equal values). All the stations are assumed to use 54 Mbps and 6 Mbps as the data and basic rate (802.11g) respectively while wired link data rate is 100 Mbps. The packet size is 1500 bytes for all flows. The buffer size at the stations and the AP is set to 100 packets per AC. The receiver advertised congestion window limits are set to 42 packets for each flow. Note that the scale on the buffer size and TCP congestion window limit is inherited from [4] (which is the first work that investigated TCP uplink/downlink unfairness problem in the WLAN). Although the practical limits may be larger, the unfairness problem exists as long as the ratio of buffer size to congestion window limit is not arbitrarily large (which is not the case in practice). We found  $\beta = 5$ , and  $\alpha = 0.05$  to be appropriate through extensive simulations. We use Jain's fairness index as a performance metric [22] where  $f = 1$  shows perfect fairness.

Except the final set, the experiments employ TCP NewReno without delayed ACKs. In these cases, the results for the proposed dynamic algorithm and the extra prioritized downlink access method closely follow each other. Due to space limitations, we only present the results for the extra prioritized downlink access method. On the other hand, as previously stated, this method cannot provide an arbitrary weight when delayed ACKs are used as in the final set of experiments. Therefore, the results for the proposed dynamic algorithm are presented for the delayed ACK scenario.

*a) Equal Wired Link Delay:* In the first set of experiments, we investigate the system performance for a practical scenario when wired link delay is same among TCP connections. We generate 10 TCP and 10 UDP flows both in the uplink and downlink. Each flow starts at the same time and the simulation duration is 100 seconds. The results of

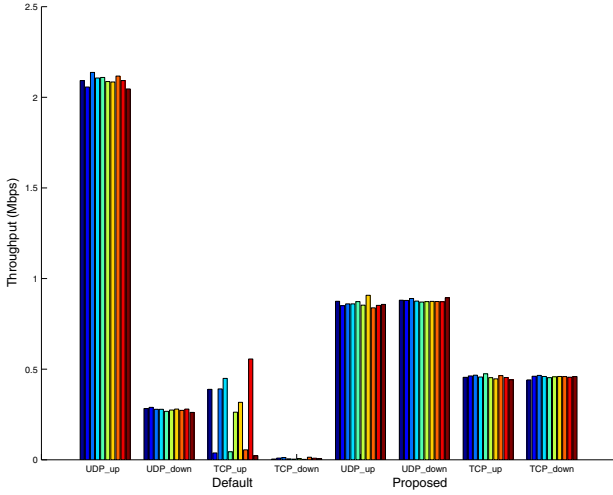


Fig. 3. Individual throughput of UDP and TCP flows when *Default* EDCA parameters and the *Proposed* adaptation algorithm are used.

the simulations when we employ either the default EDCA algorithm or the proposed method are provided in Fig. 3. Each column in Fig. 3 represents the average throughput that an individual flow gets. When default EDCA is used, *i)* there exists throughput unfairness between the uplink and the downlink TCP and UDP flows, *ii)* there exists throughput unfairness among uplink TCP connections, and *iii)* data packet losses at the AP buffer almost shut down all downlink TCP connections. When we run the proposed algorithm designed for weighted fairness support in the downlink and uplink, we define the downlink/uplink utilization requirement as  $U_r = 1$  since the number of downlink and uplink flows are equal for both ACs. The results illustrate that  $U = 1$  is perfectly achieved in terms of throughput for both UDP and TCP flows. The prioritization between ACs is maintained.

*b) Varying Wired Link Delay:* In the second set of experiments, we let wired link delays differ among TCP connections. The wired link delay of the first TCP connection is set to 24 ms and each newly generated TCP connection is assigned 4 ms larger wired link delay than the previous one. UDP wired link delay is constant for each connection. Each flow starts at different times and the simulation duration is 300 seconds. The first downlink UDP, uplink UDP, downlink TCP and uplink TCP connection start at  $t = 5$  s,  $t = 10$  s,  $t = 7$  s and  $t = 12$  s respectively. Then, a new flow of the same type arrives every 10 s. No other flow arrives after 202 s. Fig. 4 and Fig. 5 show the instantaneous UDP and TCP throughput of individual uplink and downlink flows respectively for the proposed algorithm. Since the proposed algorithm adaptively updates EDCA parameters, it maintains fair resource allocation (the downlink flows do not starve in terms of throughput). The results for the default EDCA (where the unfairness between uplink and downlink for both UDP and TCP and the unfairness among individual TCP flows both in the uplink and downlink are evident) are not included due to space limitations, but they can be found in [19].

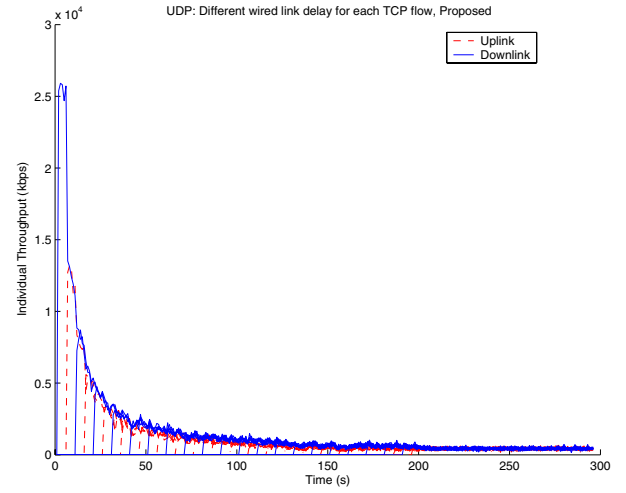


Fig. 4. The instantaneous UDP throughput of individual uplink and downlink flows for the proposed algorithm.

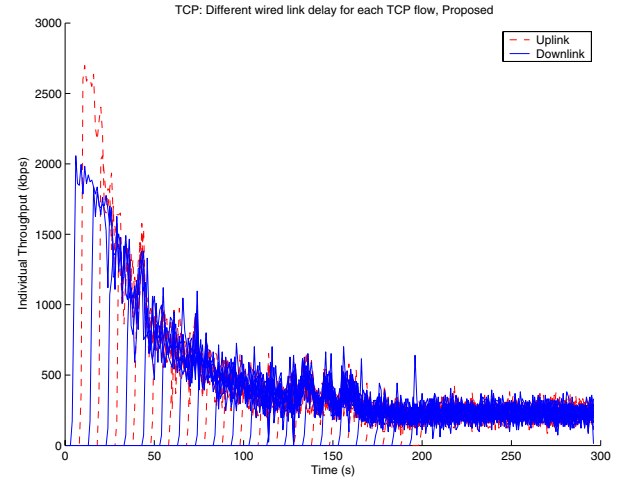


Fig. 5. The instantaneous TCP throughput of individual uplink and downlink flows for the proposed algorithm.

*c) Short TCP Flows:* We have repeated the simulation set of experiment 2 when half of the TCP flows model short flows. The simulation duration is 450 s. No other flow arrives after 302 s. The short and long TCP flows are alternatively initiated both in the downlink and uplink. Fig. 6 shows the total transmission duration for individual short TCP flows for the proposed algorithm and the default EDCA. Note that flow indices from 1 to 15 represent uplink TCP flows while flow indices from 16 to 30 represent downlink TCP flows. The file transfers with short durations can be completed in a considerably shorter time when the proposed algorithm is used, therefore we also conclude that the proposed algorithm is short-term fair. At high load, short flows experience significantly long delays and connection timeouts when default constant EDCA parameter selection is used.

*d) Coexisting Multimedia Flows:* In another set of experiments, we consider three types of traffic sources; audio, video,

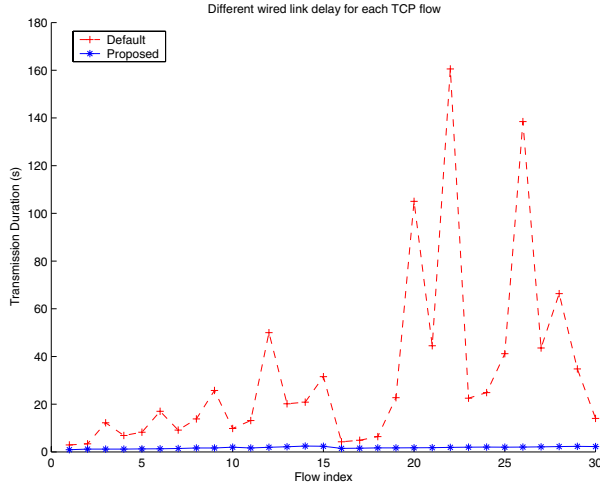


Fig. 6. The total transmission duration for individual short TCP flows.

and data. The audio traffic model implements a Voice-over-IP (VoIP) application as a Constant Bit Rate (CBR) traffic profile at 24 kbps. The constant audio packet size is 60 bytes. For the video source models, we have used traces of real H.263 video streams [23]. The mean and maximum video payload size is 2419 bytes and 3112 bytes respectively. The mean video data rate is 255 kbps. The audio flows are mapped to an AC with  $CW_{min} = 7$  and  $CW_{max} = 15$ . The video flows use an AC with  $CW_{min} = 15$  and  $CW_{max} = 31$ . For both ACs, AIFSN values are set to 2 and TXOP limits are 0. Table I shows the average throughput of uplink and downlink data flows when there are 10 voice and 10 video flows both in the uplink and downlink (a total of 40 flows with QoS requirements). We also compare the results with the proposed algorithm of [14]. The proposed adaptive algorithm effectively manages fair resource allocation for any number of stations while default EDCA and [14] fail to do so. Note that we have not included the average throughput of the flows with QoS requirements in Table I, since all audio and video flows get necessary bandwidth to serve offered load with zero packet loss rate. Table II compares the average delay of each QoS flow in each direction for default EDCA, [14], and the proposed algorithm. As the results show, the QoS flows experience slightly larger delays when the proposed algorithm is used (due to smaller CW and larger TXOP assignment for data flows). On the other hand, the delay increase is well within the limits of QoS requirements. Moreover, fair resource allocation for data flows is provided.

*e) Delayed TCP ACKs and Wireless Channel Errors:* In the fifth set of experiments, we repeat the second experiment set when TCP connections use the delayed TCP ACK mechanism (we set  $b = 2$ ) and wireless channel is assumed to be an Additive White Gaussian Noise (AWGN) channel. On top of the energy-based PHY model of ns-2, we implemented a BER-based PHY model according to the framework presented in [24] using the way of realization in [25]. Our model considers the channel noise power in Signal-to-Noise Ratio (SNR) [21]. A packet is detected at the receiver if the received power is

TABLE I  
TOTAL THROUGHPUT (IN MBPS) OF TCP FLOWS IN THE SAME DIRECTION.

		Number of TCP Flows		
		5	15	25
Default	TCP <sub>uplink</sub>	12.6231	15.1818	15.4346
	TCP <sub>downlink</sub>	0.0592	0.0357	0.0444
[14]	TCP <sub>uplink</sub>	6.5726	7.1682	7.2290
	TCP <sub>downlink</sub>	6.8154	4.8323	3.6299
Proposed	TCP <sub>uplink</sub>	6.9516	6.7476	6.8174
	TCP <sub>downlink</sub>	6.8587	6.8026	6.9033

TABLE II  
AVERAGE DELAY (IN MS) OF QoS FLOWS.

		Number of TCP Flows		
		5	15	25
Default	VoIP <sub>uplink</sub>	0.8861	1.0466	1.0813
	VoIP <sub>downlink</sub>	0.9600	1.1305	1.1669
	Video <sub>uplink</sub>	2.7317	3.6611	3.3161
	Video <sub>downlink</sub>	2.2466	3.5819	4.1146
[14]	VoIP <sub>uplink</sub>	1.2330	1.4213	1.3300
	VoIP <sub>downlink</sub>	1.2069	1.4803	1.3981
	Video <sub>uplink</sub>	3.6529	5.4018	4.8875
	Video <sub>downlink</sub>	2.9872	5.5328	5.3437
Proposed	VoIP <sub>uplink</sub>	1.3802	1.3302	1.5665
	VoIP <sub>downlink</sub>	1.4909	1.4759	1.8427
	Video <sub>uplink</sub>	4.3051	5.0728	5.3568
	Video <sub>downlink</sub>	4.3989	4.7185	5.3781

over a specified threshold. Then, the packet error probability is calculated using the theoretical model presented in [24] regarding the channel SNR, the modulation type used for the transmission of the packet, and the packet size. If a randomly drawn number is higher than the calculated probability, the packet is assumed to be received correctly, otherwise the packet is labelled as erroneously decoded, and discarded.

We set wireless channel noise levels such that each station experience a finite packet error rate (PER). We repeat the tests for AWGN channel SNR values when PER is 0%, 0.1% or 1% for UDP and TCP data packets. Table III shows the fairness index of UDP/TCP uplink and downlink flows for the period of time after all the connections are initiated (after 202 s.). We also include total throughput per AC per direction. The results reveal that the proposed algorithm maintains fair access while considerably improving the performance when compared to the default EDCA. The proposed algorithm provides fair resource allocation among individual flows in the same direction and between uplink and downlink flows using the same AC. For default EDCA, both the UDP and TCP downlink starve in terms of throughput, while fair allocation among TCP flows in the same direction cannot be maintained. The interested reader is referred to [19] for a comparison with the results of [14] in this scenario.

TABLE III

FAIRNESS INDEX AND TOTAL THROUGHPUT (MBPS) OF UDP AND TCP FLOWS WHEN DELAYED TCP ACK MECHANISM IS EMPLOYED ( $b = 2$ ). WLD AND PER STAND FOR WIRED LINK DELAY AND PACKET ERROR RATE, RESPECTIVELY.

		Default				Proposed			
		UDP		TCP		UDP		TCP	
		Uplink	Downlink	Uplink	Downlink	Uplink	Downlink	Uplink	Downlink
PER % 0	$f$	0.9998	0.9985	0.7501	0.7364	0.9997	0.9995	0.9995	0.9898
	Thp.	20.5860	2.2308	2.4571	0.1072	8.6535	8.7528	4.8133	4.8700
PER % 0.1	$f$	0.9999	0.9998	0.8682	0.8677	0.9999	0.9999	0.9999	0.9860
	Thp.	20.4910	2.2536	2.4934	0.1498	8.7322	8.7425	4.8208	4.7935
PER % 1	$f$	0.9999	0.9998	0.8606	0.8351	0.9999	0.9999	0.9999	0.9875
	Thp.	20.5020	2.2610	2.3180	0.1369	8.6342	8.5933	4.7930	4.7666

#### IV. CONCLUSIONS

We have proposed a novel model-assisted measurement-based dynamic EDCA parameter adaptation algorithm that achieves a predetermined utilization ratio between uplink and downlink flows of the same AC while keeping the prioritization among ACs. The key contribution is the following: The proposed algorithm is unique in that it dynamically adapts the EDCA parameters initially calculated analytically regarding the number of active connections and a required utilization ratio (rather than heuristically setting constant EDCA parameter values for any scenario). A novel and simple analytical model is proposed to calculate the EDCA parameter set that ideally achieves a predetermined utilization ratio between uplink and downlink flows. Another key contribution of this study is that our solution differentiates the design of weighted fair access provisioning for UDP and TCP. It is shown that the proposed model-assisted measurement-based algorithm maintains fair and efficient resource allocation while preserving QoS in a wider range of scenarios (in an erroneous wireless channel with any number of coexisting UDP and short-lived or long-lived TCP connections while TCP connections use delayed TCP ACK mechanism) when compared to the models previously proposed in the literature. The proposed algorithm is fully compliant with the 802.11e standard.

#### REFERENCES

- [1] *IEEE Standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE 802.11 Std., 1999.
- [2] *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.
- [3] H. Balakrishnan, V. Padmanabhan, and R. H. Katz, "The Effects of Asymmetry on TCP Performance," *ACM Baltzer Mobile Networks and Applications (MONET)*, 1999.
- [4] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN," in *Proc. IEEE Infocom '03*, April 2003.
- [5] 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.
- [6] J. Ha and C.-H. Choi, "TCP Fairness for Uplink and Downlink Flows in WLANs," in *Proc. IEEE Globecom '06*, November 2006.
- [7] 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.
- [8] M. Gong, Q. Wu, and C. Williamson, "Queue Management Strategies to Improve TCP Fairness in IEEE 802.11 Wireless LANs," in *Proc. IEEE WiOpt '06*, April 2006.
- [9] 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.
- [10] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN," in *Proc. ACM Mobicom '00*, August 2000.
- [11] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in *Proc. ACM Mobicom '00*, August 2000.
- [12] 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.
- [13] I. Tinnirello and S. Choi, "Efficiency Analysis of Burst Transmissions with Block ACK in Contention-Based 802.11e WLANs," in *Proc. IEEE ICC '05*, May 2005.
- [14] D. J. Leith, P. Clifford, D. Malone, and A. Ng, "TCP Fairness in 802.11e WLANs," *IEEE Commun. Lett.*, pp. 964–966, November 2005.
- [15] 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.
- [16] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Trans. Commun.*, pp. 535–547, March 2000.
- [17] I. Inan, F. Keceli, and E. Ayanoglu, "Performance Analysis of the IEEE 802.11e Enhanced Distributed Coordination Function using Cycle Time Approach," ArXiv cs.OH/ 0704.1838, April 2007. [Online]. Available: arxiv.org
- [18] —, "Saturation Throughput Analysis of the 802.11e Enhanced Distributed Channel Access Function," in *Proc. IEEE ICC '07*, June 2007.
- [19] F. Keceli, I. Inan, and E. Ayanoglu, "Fairness Provision in the IEEE 802.11e Basic Service Set," ArXiv cs.OH/ 0704.1842, April 2007. [Online]. Available: arxiv.org
- [20] (2006) The Network Simulator, ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns
- [21] IEEE 802.11e HCF MAC model for ns-2.28. [Online]. Available: http://newport.eecs.uci.edu/~fkeceli/ns.htm
- [22] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley and Sons, 1991.
- [23] P. Seeling, M. Reisslein, and B. Kulapala, "Network Performance Evaluation Using Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial," *IEEE Commun. Surveys and Tutorials*, vol. 6, no. 2, pp. 58–78, 2004. [Online]. Available: http://www.eas.asu.edu/trace
- [24] D. Qijao and S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," in *Proc. IEEE ICC '01*, June 2001.
- [25] M. Lacage. (2006) Ns-2 802.11 Support. INRIA Sophia Antipolis. France. [Online]. Available: http://spoutnik.inria.fr/code/ns-2