A-DCF: Design and Implementation of Delay and Queue Length Based Wireless MAC

Hojin Lee†, Sangwoo Moon‡, and Yung Yi‡

Abstract—**Optimal CSMA, which is fully distributed wireless MAC theory, has provided a rule of dynamically adapting CSMA parameters according to some theoretically developed principles, and has reported to offer nice analytical guarantees on throughput and fairness. Despite a couple of research efforts that transfer Optimal CSMA to practical protocols, e.g., O-DCF, our evaluation results show that they are still far from being deployable in practice mainly due to bad performance with TCP. In this paper, we first investigate how Optimal CSMA based MAC conflicts with TCP and degrades end-to-end performance, if poorly transferred to practice. Then, we propose a new wireless MAC protocol, called A-DCF, that inherits the basic framework and rationale of Optimal CSMA and O-DCF, but are largely re-designed to make A-DCF work well with TCP. The key idea of A-DCF lies in smartly exploiting both queue length and delay which widens our design space for compatibility with TCP. Our extensive simulation and experimental results demonstrate that A-DCF outperforms the traditional 802.11 and O-DCF. Particularly, we report our implementation code of A-DCF as a device driver module. To our knowledge, it is the first driver-level implementation of an Optimal CSMA based MAC protocol, being of broad interest to the community.**

I. Introduction

It has long been a challenging issue to achieve high end-to-end TCP performance in practice over wireless multi-hop networks. Many reasons for such challenges are reported by an extensive array of research, where two of them are critical. First, significantly complex interference patterns appear due to their dependence on topology and flow configurations. For example, 802.11 DCF is known to work very well in symmetric contentions, but to perform badly in asymmetric contentions [1], [2]. Second, TCP, which was originally designed without considering complex issues generated by wireless environments, enlarges the performance challenges [3], [4]. A large volume of research has been made to tackle such issues, including new CW (Contention Window) selection algorithms, MACAW for avoiding hidden terminals, frame[1] aggregation schemes, and some schemes with physical-layer aid. They also include a large volume of research papers on making TCP work well in wireless environments. See Section II for a detailed list of the related work.

In this paper, we aim at solving those two major problems, where we take an approach that is motivated by a recent theoretical advance, called Optimal CSMA. The operation principle of Optimal CSMA is that (i) each node has the individual queues for each of its outgoing links, (ii) each link's supply-demand differential is quantified based on the links' queue length, and (iii) links with larger differentials are given higher priority in outgoing transmissions. Optimal CSMA is known to have provable performance guarantees in terms of throughput and fairness [5]–[9].

Our work is highly motivated by the recent work, called O-DCF [10], that is a practical MAC protocol inspired by the Optimal CSMA theory. The major focus of O-DCF is to adaptively and smartly select a specific *combination* of access probability (i.e., contention window) and transmission length. O-DCF uses the fact that for symmetric contentions such as fully-connected topologies (see Fig. 2(a)), heavy collision is the major challenge, so reasonably low access probability is the key to high performance, whereas for asymmetric contentions such as the flow-in-the-middle topologies (see Fig. 2(b)), unfairness is the major issue, so access differentiation among the flows experiencing different contention levels becomes the critical part. The contribution of O-DCF lies in autonomous selection of access probability and transmission length without the topological knowledge of neighboring interference patterns. However, in spite of the novel transfer from theory to practice by O-DCF, O-DCF focuses only on optimizing the MAC-layer performance, and both our simulation and experiment results show that O-DCF is still far from being deployable in practice mainly due to the poor performance when used with TCP.

In this paper, we propose a new wireless MAC protocol, called A-DCF (Advanced DCF), which in part shares the basic framework with O-DCF such as per-neighbor aggressiveness control, but is re-designed for good performance under TCP. The key philosophy lies in jointly using queue length and delay, each of which has its own aspects, summarized as in what follows:

(a) Delay-based control: To quantify the supply-demand differential, A-DCF uses the sojourn time of the HOL (Head-Of-Line) packets of per-neighbor queue, rather than the queue length. In queue length based MACs (e.g., Optimal CSMA and O-DCF), a high degree of coupling between queue length and TCP exists, generating a vicious cycle, which is detrimental to fair medium access, that links with heavier contention is hard to be fed back by upper-layer TCP packets and thus their MAC aggressiveness keeps being low. Delay-based control helps such links to be decoupled from TCP ACK-clocking, and to gain higher aggressiveness quickly, escaping from the vicious cycle.

(b) Queue-based access probability boosting: In spite of

†: Software R&D Center, Samsung Electronics, South Korea, Email: hojin79.lee@samsung.com. ‡: EE Dept., KAIST, South Korea, Emails: swmoon@lanada.kaist.ac.kr, yiyung@kaist.edu.

This work was supported by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2012M3A6A6054195) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2013R1A2A2A01067633).

[1] We use the terminology 'frame' for a MAC-level data segment and 'packet' for an upper-layer one, e.g., TCP.

large benefits from delay-based control by decoupling MAC from TCP, in severer asymmetric contention scenarios more interfered flows still turn out to suffer from TCP timeouts due to intra-contention within TCP DATA and TCP ACK. To tackle this issue, we use queue length as a sign of backlogs of TCP ACK packets at TCP sinks, and let such links quickly obtain the medium access. This access probability boosting enables TCP DATAs and TCP ACKs to be appropriately interleaved, making TCP flows avoid a large number of timeouts.

(c) Queue-based retry limit and demand adaptation: In the topology where some links have absolute advantage over other links, e.g., the information asymmetry (see Fig. 2(d)), the disadvantaged flow (DF) experiences heavy RTS/CTS signaling failures. This is mainly because the advantaged flow (AF) often temporarily dominates the medium, and thus the retry limit of DF's source is easily reached, leading to TCP sources' frequent CWND (Congestion Window) reductions via fast recovery and/or timeouts. In such severe asymmetric contention cases, mild approaches such as delay-based access differentiation and access probability boosting helps a lot, but do not completely solve the performance problem. To address this problem, we additionally allow DFs to adaptively increase the retry limit[2], and enforce AFs to reduce its medium aggressiveness by decreasing the potential demands in the supply-demand chain.

We evaluate the performance of A-DCF through extensive NS-3 simulations and real implementations on the 14-node wireless multi-hop testbed, whose source codes and manuals are publicly available in [11]. We implement A-DCF on top of a legacy 802.11 chip, by modifying a device drive module in the ath9k driver package [12]. We use the ath9k's functionality of accessing (i) the chip registers and (ii) TX control descriptors for various per-frame controls. Our implementation conforms to the specifications that cannot be modified by the chip, e.g., the allowable CW values $(2^n - 1)$ and the maximum packet aggregation size. Namely, the implementation of A-DCF is fully compatible with the legacy 802.11 chips. Our performance evaluation demonstrates that A-DCF outperforms 802.11 and O-DCF especially when it is used under TCP.

II. Related Work

Optimal CSMA theory has been developed for fair and throughput-optimal access [5]–[9]. The key idea lies in dynamically adjusting media access aggressiveness based on the dynamic control of the CSMA-induced Markov chain using the MCMC (Markov Chain Monte Carlo) technique. The authors in [13], [14] present the gap between theory and practice, reporting "uncareful" implementation of Optimal CSMA theory may not perform well in practice. O-DCF [10] is the first work to propose a practical MAC protocol inspired by Optimal CSMA theory, where the main idea is to find a suitable combination of contention window size and transmission length so that O-DCF works well under different contention scenarios. As mentioned earlier, A-DCF basically shares the research direction with O-DCF. However, A-DCF differs from O-DCF in that O-DCF is designed without seriously considering TCP as a transport protocol, which is of critical importance in practice. Moreover, O-DCF has been implemented using an overlay platform, which may be enough just for the performance evaluation of O-DCF, but somewhat limited to further research efforts in the community. Our work A-DCF further optimizes O-DCF by largely modifying O-DCF, and also provides the kernel-level device driver (also the driver for O-DCF for performance evaluation), which we believe is of independent and broader interest to the community.

Other researches on TCP over Optimal CSMA include [15], [16]. The authors in [15] show the problem of TCP performance over Optimal CSMA, and propose an approach that opens multiple sessions in proportion to RTT. This multi-session approach helps in breaking the vicious cycle and thus improving the performance, but explicit management of multiple sessions does not seem very practical whereas A-DCF does not require any modification above the MAC layer. The authors in [16] theoretically study the round-trip-delay based TCP (such as TCP-Vegas), and thus no real implementation. Delay as a mesure of supply-demand differential in the theoretical study of cross-layer design has been used in other work, e.g., [17]. Both [16] and [17] similarly use delay-based medium access, but the motivation significantly differs from our work in that our joint use of queue and delay comes much from practical one, i.e., decoupling TCP ACK-clocking and MAC queue dynamics for practice. There exists efforts [18]–[20] which bridge the gaps between theory on queue-based MAC (not necessarily CSMA) and practice by reflecting queue length information over 802.11.

There exists an extensive array of research on the performance problems of 802.11 DCF, as well as solutions, especially in its application to wireless multi-hop networks. Problems of bad performance and throughput unfairness are revealed in the challenging topologies such as HT, IA, FIM [1], [2], [21]–[24]. Proposals to handle such problems include pure MAC-level approaches and those with help of the physical-layer, ranging from dynamic adjustment of CW [25]–[28], packet aggregation [29], [30] to the new collision handling [31], [32] using the PHY information. Finally, making TCP work well over wireless multi-hop networks has been long studied for a couple of decades, e.g., instability and fairness problems [4], [33], in many cases requiring the modification of TCP or the support of the architecture change. We refer the readers to the survey of [34].

III. A-DCF Design

In this section, the design of A-DCF is described. To this end, each feature in our design is presented, followed by its rationale in conjunction with microbenchmark simulation results. We start by the overview of our design and the architecture of A-DCF.

A. Overview: Architecture and Aggressiveness Control

Architecture: In A-DCF, each node runs a per-neighbor control for accessing the medium by maintaining per-neighbor

[2] Every station in 802.11 has the notion of retry limit that is a counter representing the maximum allowable retransmission trials, see Section III-D for more details.

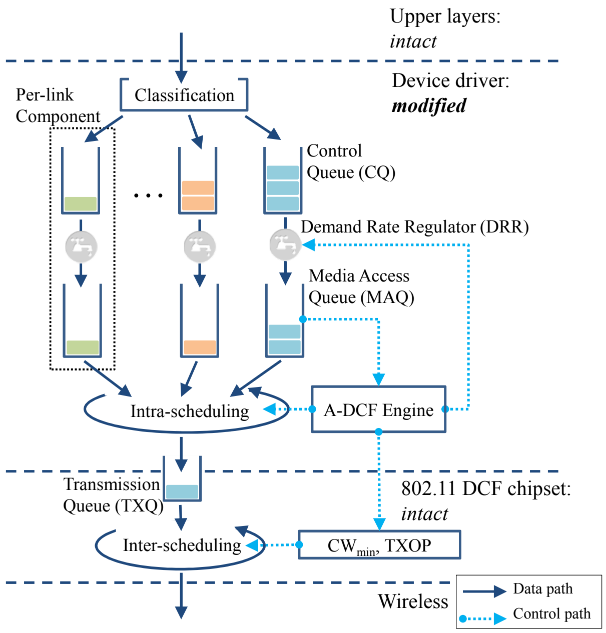


Fig. 1. Architecture of A-DCF

states, as shown in Fig. 1. Those states are used to determine how aggressively the node should access the medium in transmitting frames in a (link-level) destination-dependent manner. To this end, we maintain two per-neighbor queues: CQ (Control Queue) and MAQ (MAC Queue). CQ has the role of buffering the packets from upper-layers, where each packet from upper-layers is first classified according to its destination, and then enqueued into its per-neighbor CQ as frames. MAQ functions as a per-neighbor state that is importantly used to determine frames' medium access aggressiveness. A notion of Demand Rate Regulator (DRR) resides between a CQ and a MAQ, and controls the dequeuing rate from the CQ to the MAQ. How the dequeuing rate is decided is critical in achieving fair medium access in A-DCF (see Section III-B). Then, the service from a MAQ occurs when the HOL (Head-Of-Line) frame of the MAQ is moved into TXQ (Transmission Queue) from which a 802.11 chip fetches for actual transmission. When frames are moved into TXQ[3], 802.11 DCF parameters such as $\mathrm{CW}_{\min}$ and TXOP are appropriately set for controlling access aggressiveness. In presence of multiple backlogged neighbors (and thus multiple MAQs), as an intra-scheduling phase, the MAQ with the largest index (which is the largest sojourn time of the HOL frame in A-DCF) is scheduled, where multiple frames in the served MAQ can be served simultaneously at this scheduling chance, as detailed in the next paragraph.

Per-neighbor aggressiveness control: CSMA has two critical parameters for controlling its aggressiveness: (a) access probability and (b) transmission length. In many practical MACs such as 802.11, access probability is typically controlled by contention window (CW) size, and transmission length corresponds to the number of consecutive transmitted frames without separate media sensing. By aggressiveness we simply mean the product of access probability and transmission length, which are controlled differently for different neighboring links.

Aggressiveness in A-DCF is basically controlled by the following simple rule:

$$\begin{aligned}&\text{Aggressiveness (acces prob.} \times \text{trans. length)}\\&\qquad = \exp(\text{sojourn time of HOL pkt. of MAQ}), \quad (1)\end{aligned}$$

where we simply call the sojourn time of HOL frame of a MAQ as *delay* of the MAQ. Intuitively, the delay of MAQ tracks how well a link has been served over time.

The key design aspects of A-DCF lie in which combination of access probability and transmission length should be chosen for TCP over A-DCF to achieve high performance. When a frame (or a multiple of frames) from a MAQ is moved to TXQ by the intra-scheduling for being ready for actual transmission, A-DCF's procedure of setting CSMA parameters is divided into the following three steps:

(1) Initial access probability: For a frame f enqueued to TXQ, using its per-neighbor state (i.e., its MAQ's delay), an initial CW is smartly selected, where the basic principle is that the frames from under-served MAQs in terms of delay are assigned smaller CWs.

(2) BEB for actual CW: Once the initial CW size is chosen as a function of MAQ's delay, the actual medium access is attempted, allowing BEB (Binary Exponential Backoff) to occur, which corresponds to a distributed search of the actual access probability.

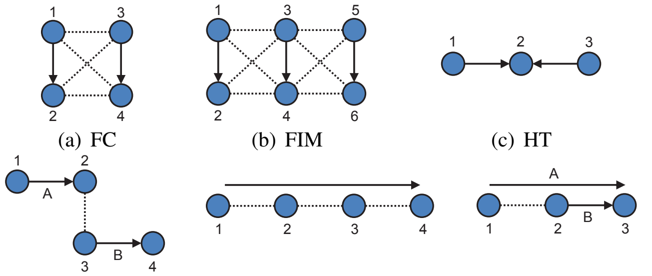
(3) Transmission length selection: Once the actual CW is obtained after BEB, it is converted to an access probability, and then the transmission length is determined from (1) by considering the corresponding MAQ's delay and the maximum transmission length specified in the legacy 802.11 chip.

We comment that A-DCF shares the overall architecture (e.g., MAQs and CQs) and the access aggressiveness control procedure (i.e., three basic steps) with O-DCF. However, we expect that they seem to be needed in any practical MAC protocol motivated by per-neighbor queue based MAC from theory (e.g., Max-Weight and Optimal CSMA). The actual contribution of A-DCF lies in proposing new rules of control methods, which show the drastic performance difference as discussed throughout this paper. In presenting our design components, we will also provide microbenchmark simulation results to directly see the effect of each design feature over the simple and atomic topologies in Fig. 2 that are popularly used for wireless multi-hop MAC research.

B. Using Delay as Supply–Demand Differential

As mentioned earlier, in A-DCF we use a notion of *delay* of a MAQ of link l which corresponds to the sojourn time of a HOL frame of MAQ l. We denote by $D_l(t)$ the delay of MAQ for link l at time t. The delay of a MAQ intuitively means that how long the MAQ has not been served. In Optimal CSMA theory (or its queue-length based MAC precursors, e.g., Max-Weight), it is important to maintain the quantified information about supply-demand differential, based on which each link

[3]Strictly speaking, TXQ exists in the memory of the host system, from which the 802.11 chipset performs DMA (Direct Memory Access) for preparing actual transmissions.



(a) FC (b) FIM (c) HT (d) IA (e) CH (f) GW

Fig. 2. Atomic topologies: Dotted lines and directed solid arrows indicate mutual interference and traffic flows, respectively. (a) FC (fully-connected), (b) FIM (flow-in-the-middle), (c) HT (hidden terminal), (d) IA (information asymmetry), (e) CH (chain), and (f) GW (gateway)

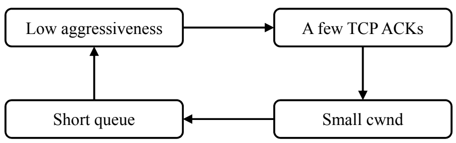


Fig. 3. Vicious cycle: Coupling between TCP's ACK-clocking and MAC queue length

has different intensity of accessing the media. In A-DCF we use delay as supply-demand differential, and claim that it is much better when coupled with TCP. We separately explain the components, DRR and aggressiveness control in A-DCF, where delay-based control is importantly used.

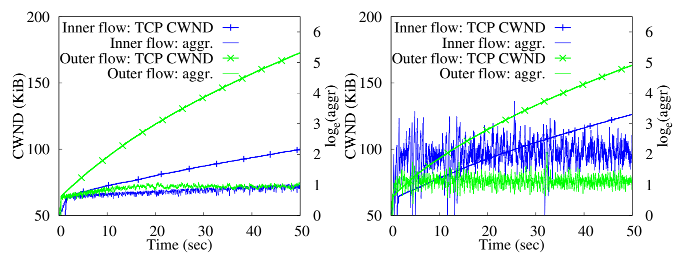
Demand rate regulator (DRR): First, DRR controls the growth rate of a MAQ, for which MAQ's delay is used as follows[4]:

$$\text{Rate from CQ to MAQ for link } l = \frac{V}{d_l(t)}, \tag{2}$$

where $d_l(t) = bD_l(t)$, and b and V are some constants. Intuitively, DRR decreases the rate for the MAQ with long sojourn time of its HOL frame, and increases the rate when the MAQ has been well served so far. As well-known from the theory of Optimal CSMA, b is a small constant value that corresponds to a step size, being responsible for slowing down the variations of D_l, and V is the constant that controls the sensitivity of transfer rate from CQ to MAQ.

Aggressiveness Control: As mentioned earlier, access aggressiveness for link l is decided as $\exp(d_l)$. Thus, when a HOL frame of some MAQ has not been served for a long time, then it has high access aggressiveness by having either small CW size and/or long transmission length. However, note again that it remains to design a mechanism on how to choose the combination of access probability and transmission length. The new features of A-DCF focus on choosing the initial CW size (i.e., step **(1)** in the previous section), which will be described in Section III-C.

Rationale: The main reason of using delay as a measure of supply-demand differential is that conventional queue-based adaptive CSMA, e.g., O-DCF, performs badly due to the so-called *vicious cycle*, which is generated by coupling between TCP's ACK-clocking and the queues at MAC (see Fig. 3). When a link l is ill-served due to heavier interference, a TCP flow over link l is less transmitted and thus gets fewer TCP ACKs, which leads to sluggish growth of CWND (and thus highly low performance of the flow over l). Note that queue length is bounded by TCP's CWND. A small CWND results in low access aggressiveness in the queue-based CSMA. Thus, a TCP-flow link with heavier interference is difficult to have sufficient queue length to escape from this vicious cycle. Delay-based aggressive control as well as demand rate regulation enables *decoupling* between TCP and MAC because even small MAC queue is capable of attaining high access aggressiveness without involving TCP's frame injection. Fig. 4 shows the simulation results for the FIM topology (Fig. 2(b)) with two outer flows, where we observe that the inner flow experiences sluggish CWND increase, and the inner flow's aggressiveness stays low. However, in a delay-based method the inner flow's CWND and aggressiveness grows faster and stays high, respectively.



(a) Queue-based (O-DCF) (b) Delay-based

Fig. 4. Impact of delay-based control: FIM with two outer flows

C. Initial CW Selection: Jointly Using Delay and Queue

In the previous section, we described delay-based control as a basic principle of access aggressiveness control. We now elaborate its realization in A-DCF's first step: **initial access probability**, for which delay and queue are jointly used. We first present the algorithm for the initial access probability selection, and then explain its rationale.

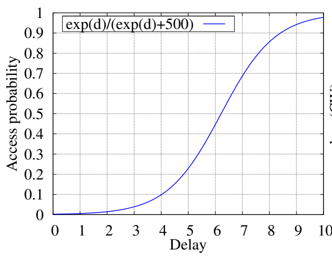
Initial CW Selection

1: $d_l \leftarrow [b \times D_l]_{d_{\min}}$
2: $\text{CW}_{\min} \leftarrow \text{sigmoid}(d_l)$
3:
4: **if** $Q_l \leq Q_{boost}$ **then**
5: $\quad \text{CW}_{\min} \leftarrow \frac{\text{CW}_{\min}}{\text{CW}_{\text{boost}}}$
6: **end if**

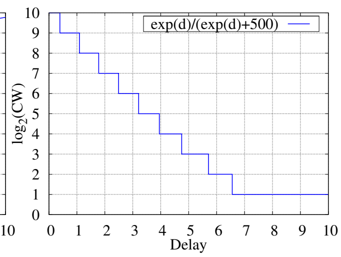
Delay-based access differentiation: Recall that in Section III-A we initially choose the initial $\text{CW}_{\min}$ for the medium access based on the MAQ delay. To this end, we first adopt a sigmoid-type function for *access differentiation* (Lines 1-2), where Fig. 5 shows the sigmoid-type function in A-DCF. Employing a sigmoid-type function is first proposed in O-DCF, where the major goal was to assign different access probability to the flows experiencing different contention levels. For example, as opposed to the symmetric contentions such as the FC (Fig. 2(a)), in asymmetric contentions such as FIM-like topologies, almost no collisions occur and thus BEB

[4]The form of $1/d_l(t)$ comes from Proportional Fairness in theory, where $1/x$ is the inverse of the utility function $U(x) = \log(x)$.

rarely operates, and in this case the inner flow's starvation is the major issue. To tackle this issues, we require that the CW size (or the access probability) of link flow l that contends with many outer flows should be *small (or high)* and thus *prioritized* enough that the link l avoids rare channel access opportunity. We refer the readers to O-DCF [10] to the mathematical discussion on the necessity of this sigmoid-type function. The difference in A-DCF from O-DCF lies in using delay rather than queue-length, which helps to avoid the vicious cycle, as mentioned in the previous subsection. To avoid the situation that a link has too low access probability, we have the minimum access probability floor, $d_{\min}$.



(a) Delay vs. p

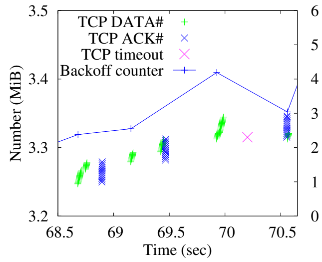


(b) Delay vs. CW

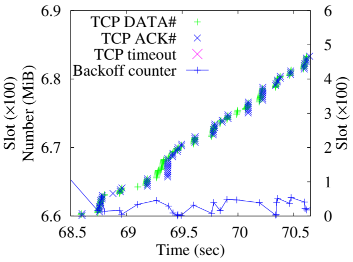
Fig. 5. Delay-based access differentiation: 10 in x-axis corresponds to 1 sec. ($d = bD$, where $b = 10^{-5}/\mu$sec).

Queue-based access probability boosting: In addition to delay-based access differentiation, we also apply the mechanism that in case when MAQ's queue-length is very small, the access probability is boosted up (i.e., $\text{CW}_{\min}$ is reduced by a multiplicative factor), which we call *access probability boosting* (Lines 4-6).

Rationale of Boosting: Delay-based aggressiveness control and access differentiation greatly contribute to TCP's performance in terms of mitigating the vicious cycle and contention level-dependent medium access. However, they still suffer from the following *TCP ACK starvation:* Consider a FIM-like asymmetric contention scenarios and let v and w be the TCP source and sink of the inner flow, respectively. Assume that at some time v has a large MAQ length due to large CWND. In this case, the sink w's TCP ACKs may not sometimes be well transmitted to the source v because (i) the frame containing TCP ACK can be assigned a large backoff counter due to a random selection of between 0 and $\text{CW}_{\min}$, and more importantly (ii) the source v has even higher access probability than w since the forward TCP DATA flow and the reverse TCP ACK flow contend with each other and v has been backlogged longer than w (i.e., delay-based control). Rare but extremely late TCP ACK transmissions naturally lead to TCP timeouts. Fig. 6(a) demonstrates our explanation, where we sometimes have very large CW values. Unfortunately, one of such CW spikes (the third point in Fig. 6(a)) results in the TCP timeout. Even if large CW values of TCP ACK does not always lead to timeout, as shown in the figure, RTT becomes high long, which can be a source of the performance drop. To solve this problem, we allow a node to indirectly infer this situation by investigating the current queue length, where when the queue length is below some threshold Q_{boost}, we boost up the access probability. Note that we carefully choose this threshold Q_{boost} (just 100 bytes in the current A-DCF) low enough to differentiate from other usual cases. Fig. 6(b) shows that our boosting scheme assigns reasonably low CW size to the frame encapsulating TCP ACK, and thus CWND grows fast without timeout.



(a) Delay-based: Without boosting



(b) Delay-based: With boosting

Fig. 6. Impact of queue-based access probability boosting: FIM with three outer flows.

D. Queue-based Reliability and Adaptive DRR

We now describe the design components for "extreme" asymmetric contention cases such as IA (see Fig. 2(d)), where the flow A has significant advantage over the flow B. In this case, delay-based control and access probability boosting do not perfectly work, but instead we choose another components: (i) reliability adaptation and (ii) DRR adaptation. Note that in O-DCF, just RTS/CTS in conjunction with queue-based access differentiation was satisfactory to this case due to its focus only on MAC-level performance, which, however, does not hold in presence of TCP. We start by explaining how practical wireless MACs, e.g., 802.11, handle frame losses.

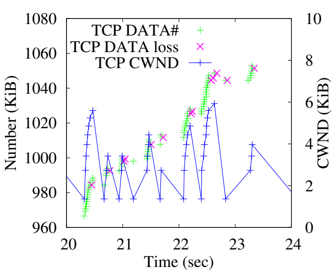
Retry counter and limit: In 802.11 DCF, error detection and correction are up to the station that begins frame transmissions, where when an error is detected, the source station must resend the frame. In many cases, the sender infers frame loss by the lack of a positive acknowledgment from the receiver, e.g., CTS loss for RTS, or ACK loss for MAC DATA, for which each sender maintains a notion of *retry counter* (for each MAC frame). A retry counter for a frame is incremented up to the *retry limit,* whenever the frame is retransmitted. Two kinds of retry counters exist: long counter for normal MAC frames and short counter for RTS/CTS frames, where their default retry limits are four and seven in 802.11, respectively. Unless confusion arises, we just use 'retry limit' for the short retry limit because we use RTS/CTS for the hidden node situations such as IA and HT.

Reliability Control and DRR Adaptation: Consider the IA topology, where suppose that the disadvantaged flow (DF) is temporarily starved for a short duration due to the advantaged flows' (AF) aggressive transmissions. In this case, even short starvation results in a series of RTS failures (because of no CTSs) due to AFs' aggressive channel reservations, which is regarded as frame loss as mentioned earlier. This results in TCP CWND reduction (via timeout or fast recovery), which in turn decreases the MAQ length, and thus has to take some time in the MAQ for medium access in A-DCF, i.e., significant TCP performance drop in the DF. This can be regarded as another kind of vicious cycle, but more serious than that in Section III-B because it is explicitly generated by the MAC-level frame loss. This serious vicious cycle may not be escaped from by just a delay-based control.

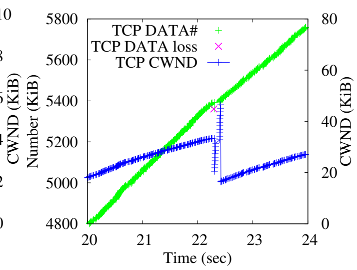
To tackle this problem, we take a more explicit approach (than just adjusting access probabilities) which enforces DFs to persist more in trying to access the medium prior to notifying "loss" signal to TCP, and AFs to reduce their demand rate (thus, lowering access aggressiveness). In what follows, we formally state such mechanisms:

- *Queue-based retry limit:* For each link l, whenever $Q_l \leq Q_{robust}$ for some threshold Q_{robust}, the retry limit is set to R_{robust}, and the default value (i.e., seven in 802.11) otherwise.
- *Adaptive DRR:* In DRR, each link l has its own $V_l = V$ as a default in (2), which is reduced to $V_l = V_{low}$whenever only CTS is detected without its preceding RTS.

Note that the case when only CTS is received by a node gives us the hint that the node is the source of the AF. Fig. 7 shows the microbenchmark of the DF in the IA, where without adaptive control many TCP DATA losses occur (due to reaching the retry limit, 7) and CWND does not grow consistently, whereas with adaptive control, the TCP flow experiences a single TCP DATA loss, growing CWND safely.



(a) Non-adaptive



(b) Adaptive

Fig. 7. Impact of queue based retry limit and adaptive DRR: IA

E. Binary Exponential Backoff and Transmission Length

Once the initial CW size is chosen as a function of MAQ's delay, the actual medium access is attempted, allowing BEB (Binary Exponential Backoff) to occur, which corresponds to a distributed search of the actual access probability. Once the actual CW is obtained after BEB, it is converted to an access probability, and then the transmission length is determined from (1) by considering the corresponding MAQ's delay and the maximum transmission length specified in the legacy 802.11 chip. The above two steps are also employed in O-DCF. Thus, we omit their explanations here due to space limitation.

F. Design Compatibility and Parameter Choice

Design compatibility: First, access probability boosting may cause significant collisions, e.g., in the FC with many competing flows. A-DCF has BEB as a second step to find the actual contention aggressiveness, evading from heavy collisions quickly (see Section V for a validation). Also, boosting is not abused since it is enabled only when MAQ length is very short. Second, retry limit trade offs between robustness and responsiveness, i.e., a node can wait for a long time in spite of the real frame loss. Our design tries to choose a good trade-off point by adaptively tuning the retry limit only for the links with small queue lengths, and thus dynamically adjusting A-DCF to the case when robustness dominates responsiveness, and vice and versa, in terms of TCP performance.

Parameter choice: b and V are the popular parameters in all queue based MAC from optimization theory. The step size b does not sensitively affect the performance as long as it is reasonably small as in the classical optimization algorithms. We choose V so that the maximum demand rate, $\frac{V}{d_{min}}$, largely exceeds the link capacity to prevent under-utilization. Threshold Q_{boost} in access probability boosting is set to be smaller than even just a single data packet, but two or three times larger than TCP ACK size, in order to selectively apply boosting, but not to be played by other normal flows. Our extensive simulations guide us that Q_{robust} should be configured to a value which effectively reduces TCP timeouts but does not increase collisions significantly, where we observe that the packet drops at MAQ backlogged less than 15 packets tend to eventually lead to TCP timeouts. We set R_{robust} to 21. The value larger than 21 still works, but its gain was marginal. Table I presents our parameter choices in all of our performance evaluations in Section V.

TABLE I
PARAMETERS FOR A-DCF

V, V_{low}	400, 100 kB/sec	b	$10^{-5}/\mu$sec
Q_{boost}	100 B	$\mathrm{CW_{boost}}$	8
Q_{robust}	15 packets	R_{robust}	21

IV. IMPLEMENTATION

Setup: We implement A-DCF by modifying 802.11 DCF of the open-source ath9k driver on the Linux kernel 2.6.32-38, where we use the NICs with AR9280 chipset. The ath9k package is the collection of 802.11 drivers for the atheros chipset family. Due to space limitation, we omit the implementation details. Our driver can be downloaded in [11], which includes the user and implementation manuals to help other people to use and modify our source code.

Queue management: In A-DCF, a station maintains CQ and MAQ, both of which are constructed for each neighboring link. They are implemented in the ath9k module (thus existing in the kernel space). Note that the memory overhead for CQ and MAQ are highly marginal since they mainly keep just the pointers (i.e., `struct sk_buff *`) to MAC frames. When upper-layers transfer the packets to the MAC-layer, its destination address is analyzed and enqueued at CQ, during which the conversion from packet to MAC frame occurs. When the frames from CQ are moved to MAQ, each frame is tagged with the enqueue time. MAQ additionally keeps track of the total bytes (for access probability boosting, see Section III-C) and the total number of packets (for queue-based retry limit, see Section III-D). The frames in each MAQ are transferred to TXQ buffers in the memory, where the default TXQ size is 512 frames. To minimize the temporal gap between the service from MAQ and the actual transmission, we reduce TXQ size to two frames through a device driver modification[5].

[5]Whenever 802.11 chipset finishes the actual frame transmission, TX interrupt occurs, which enables the frame transfer from a MAQ to TXQ. Setting TXQ size to one frame causes inefficiency due to TX interrupt delay.

Demand rate regulation: In our implementation, each MAQ maintains an injection timer, which is called every 4 ms. When the timer for a link, say link l, expires, the amount of $\frac{V}{d_l} \times \Delta t$ bytes are moved from CQ to MAQ, where Δt is the elapsed time since the previous expiration. Note that since the Linux kernel timer does not always provide the exact timer ticks, this adjustment is necessary for the accurate implementation. To deal with the case that only the part of the calculated injected bytes is inserted due to packetization, A-DCF maintains a deficit counter to store the remaining bytes of insertion in the next injection instance.

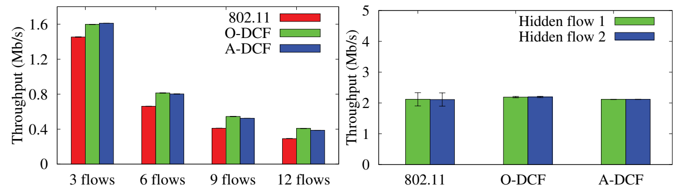
$\mathbf{CW_{\min}}$ selection and frame aggregation: In theory, access probability is used rather than contention window for 802.11. To handle this, we use the relation that captures the average behavior, $p_l = \frac{2}{\mathrm{CW}_{\min}+2}$. Since the value of $\mathrm{CW}_{\min}$ is restricted to $2^n - 1$, we choose $\mathrm{CW}_{\min}$ closest to p_l satisfying the restriction. To configure $\mathrm{CW}_{\min}$, the value of `DATA_CW_MIN` register is dynamically set. ath9k driver provides the flexible control of TXOP length, through which we are able to back-to-back PPDU transmissions to implement the frame aggregation for transmitting the frames as guided by A-DCF. In ath9k, TXOP can be specified per-frame Tx Control Descriptor: `vmf` field and `burst_duration` field, where except for the last packet of back-to-back transmissions, `vmf` field is set. Then, the amount of `burst_duration` is reserved after completing this transmission (including 802.11 ACK) via NAV setting. We set `burst_duration` to the duration of the next packet transmission including 802.11 ACK, which effectively protects back-to-back transmissions from interfering links' intervention [6].

Adaptive retry limit and hidden node detection: Short retry limit of A-DCF is configured by Tx Control Descriptor: `tx_tries` field. Hidden node detection for adaptive demand rate regulation is required by the advantaged flows, where overhearing control or data frames is necessary. We perform this by configuring `MAC_PCU_RX_FILTER` register, which, if set, allows the overheard frames to be transferred up to the device driver.

V. Performance Evaluation

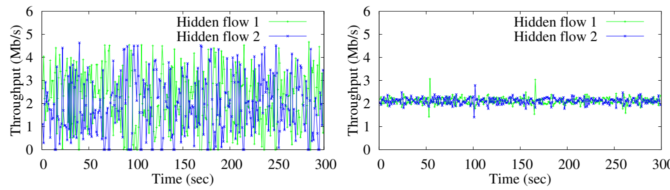
Both simulations on NS-3 and experiments on our 14-node wireless testbed are conducted to evaluate the performance of A-DCF with goal of obtaining both controllability and reality (e.g., channel error, packet capture, and realistic radio propagation, etc). In this paper, using NS-3 simulations we show the performance results for the "basic" topologies as done in other related work [3], [10], [14] (see Section V-A), and some random topology, which is regarded as a collection of such basic topologies, is generated in the experimental evaluations (see Section V-B). Other evaluation environments are presented in Table II. We compare A-DCF with 802.11 DCF and O-DCF [10]. 802.11 DCF is a *de facto* standard MAC, and O-DCF is the recent MAC motivated by Optimal CSMA.

[6]Two other candidate mechanisms exist: frame aggregation by (i) A-MSDU, (ii) A-MPDU. Their limitations are: the TX length of A-MSDU is limited to just 7935 bytes, and it is impossible to dynamically change the length of A-MPDU because the aggregation procedure of A-MPDU is internally processed on the 802.11 DCF chipset.



(a) FC: average per-flow thoughput (b) HT topology

Fig. 8. Simulation results: Average throughput for symmetric contention



(a) 802.11 (b) A-DCF

Fig. 9. Instantaneous throughput in HT

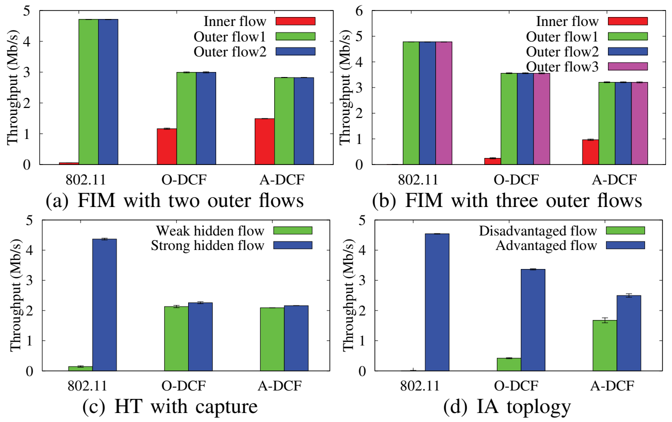
TABLE II
EVALUATION ENVIRONMENT

PHY rate, slot duration	6 Mb/s, 9 μsec
TCP ver., TCP MSS	Reno, 1448 bytes
Time duration	300 sec for sim., 60 sec for exp.
Confidence	20 for sim., 10 for exp. with 99% conf. interval

A. Basic Topologies

(a) Symmetric contention: Two types of topologies are discussed here: FC (Fully Connected, see Fig. 2(a)) and HT (Hidden Terminal, see Fig. 2(c)), whose results are shown in Fig. 8. In HT, we enable RTS/CTS signaling for mitigating the hidden node problem. First, in FC, Fig. 8(a) shows that all three protocols perform similarly with the varying number of contending flows from 3 to 12. This is not surprising because in FC reaching to reasonable access probability is the key to high performance, for which BEB in all three protocols plays the major role. Similarly, in HT, three protocols achieve the similar long-term throughput, where RTS/CTS significantly helps to detect the hidden nodes. One interesting observation is the short-term behaviors in HT as shown in Fig. 9, where high fluctuation exists in 802.11, compared to in A-DCF. This is because 802.11 always starts with its static and low $\mathrm{CW}_{\min}$ for each transmission, often leading to frame drops and thus TCP timeouts of a flow. Then, the other flow occupies the medium for some duration, and this repeatedly occurs between two flows, whereas A-DCF dynamically adjusts its aggressiveness, resulting in more stable short-term behaviors.

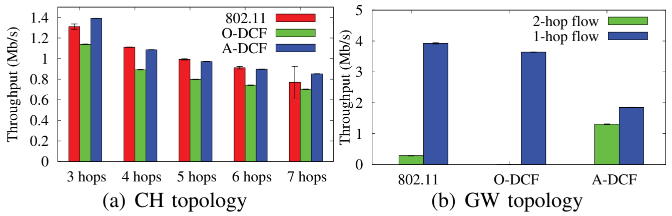
(b) Asymmetric contention: Asymmetric contentions pose major challenges to the performance in wireless networks. First, Figs. 10(a) and 10(b) show the results in the FIM with two and three outer flows, respectively. We observe that serious starvation in 802.11 occurs, as reported in [2], but A-DCF achieves a near-optimal share. O-DCF tries to give more chance to the inner flow, being reported to achieve the near-optimal performance at the MAC-layer, which, however, is still observed to suffer from the inner flow's starvation under TCP. Second, we examine the impact of capture due to RSS (received signal strength) asymmetry, whose results are shown in Fig. 10(c) for the HT with capture (i.e., see



(a) FIM with two outer flows (b) FIM with three outer flows

(c) HT with capture (d) IA toplogy

Fig. 10. Simulation results: Average throughput for asymmetric contention



(a) CH topology (b) GW topology

Fig. 11. Simulation results: Average throughput for multi-hop scenarios

Fig. 2(c), where one strong flow's packets are always decoded even when two flows simultaneously send). We observe that the weak flow is seriously starved in 802.11 because rare BEBs of the strong flow does not often allow the weak flow's access. However, in A-DCF as well as O-DCF, the strong flow's aggressiveness is dynamically reduced when it is well-served, providing access chances to the weak flow. Finally, we consider the IA (Information Asymmetry, see Fig. 2(d)), where severer asymmetry exists. Fig. 10(d) tells us that in 802.11, as well known in [1], the disadvantaged flow (DF) is significantly starved due to high frame losses, whereas O-DCF is just slightly better than 802.11, often failing to recover fair resource allocation since high loss probability leads to the reduction of TCP CWND, and thus the vicious cycle in the DF. A-DCF highly outperforms 802.11 and O-DCF thanks to adaptive control of reliability and demand regulation.

(c) Multi-hop scenarios: We now consider the case when flows consist of multi-hop links, for which two topologies are studied: CH (CHain, see Figs. 2(e)) and GW (GateWay, see 2(f)). First, in CH, we plot the long-term throughput as the hop length varies in Fig. 11(a), where we observe that the performance difference is marginal among three protocols. Note that the performance naturally decreases with increasing hop length due to more collisions and imperfect spatial reuse of CSMA. Second, we discuss GW with a mixture of one-hop and two-hop flows, which is regarded as the basic building block of mesh networks. Fig. 11(b) shows that in 802.11 the two-hop flow is almost starved whose reason is well examined by [3]; frequent collisions between node 1's TCP DATA and node 3's TCP ACK penalize the two-hop flow. A-DCF solves this problem by effectively reducing collisions between nodes 1 and 3 as mentioned earlier in Section V-A-(b), As a result the two-hop flow's performance is about

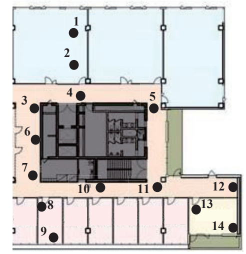


Fig. 12. Our wireless indoor testbed (30 m $\times$ 30 m).

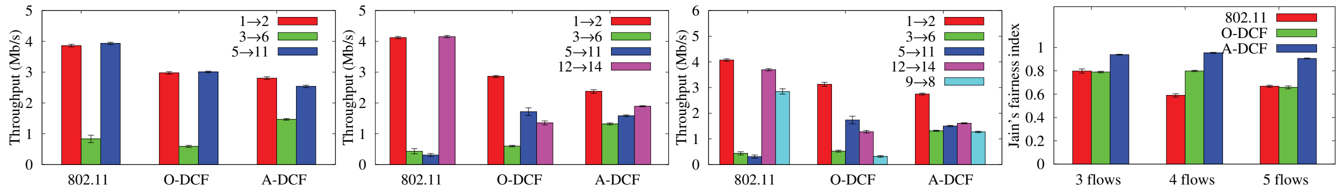
half of that of one-hop flow, which is nearly optimal in this case. Somewhat surprisingly, O-DCF performs even worse than 802.11. This is due to the vicious cycle; Node 1 rarely receives its TCP ACKs. At node 2, O-DCF's queue-length based intra-scheduling prioritizes the link to node 3 since its MAQ is typically longer than MAQ for the link to node 1.

B. *Real Deployment: Practical Scenarios*

Our performance evaluation on a real testbed with 14 nodes is made in a typical office environment with walls and glass meeting booths whose map is depicted in Fig. 12. This real environment reflects the effects of channel error, packet capture, and radio propagation, etc. Each node is a netbook with 1.66 GHz CPU and 1 GB RAM, which runs Linux kernel 2.6.32-38, equipped with a single-chip 802.11a/b/g/n NIC (AR9280 chipset). To enable multi-hop connectivities and suitable spatial reuse despite our dense deployment, we reduce the TX power of each node to 15 dBm. Among all connected links, we selectively use only reliable links whose packet loss ratios are less than 5%. Our experiments are made for two topological scenarios: WLAN and Mesh as detailed next.

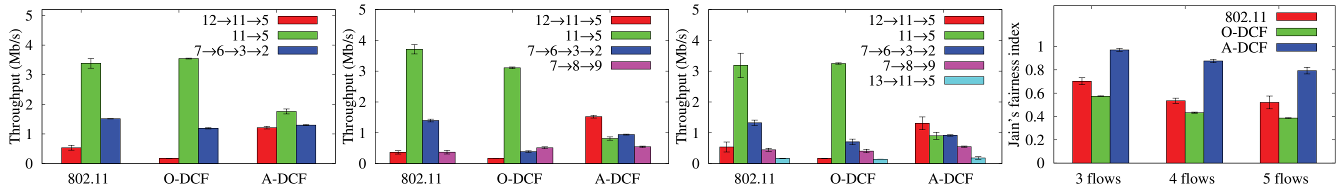
(a) WLAN: We randomly select 10 nodes, 5 nodes as APs and other 5 nodes as stations. Then, each station is associated with one AP that provides the most reliable connection, where we generate 3, 4, and 5 single-hop TCP flows. This random configuration permits us to investigate how A-DCF performs in the mixture of atomic topologies in practice. Fig. 13(d) shows that A-DCF outperforms others in terms of fairness (up to 62.1% over 802.11 and 37.5% over O-DCF in Jain's fairness index). This result is manifested by Figs. 13(a), 13(b) and 13(c), which show how throughputs of all flows are distributed.

(b) Mesh: We randomly choose two nodes as gateways (GWs) and five different nodes as stations. Other unselected nodes act as relays to forward packets between stations and GWs. Each station selects its shortest GW and establishes a multi-hop TCP session, where we statically configure the routes from stations to GWs to purely focus on the impact of MAC and TCP. We vary the number of concurrent TCP flows from three to five, whose results are shown in Fig. 14. Fig. 14(d) reveals that A-DCF outperforms others in terms of fairness (up to 64.3% over 802.11 and 107.9% over O-DCF in Jain's fairness index). The distribution of throughputs of all flows is plotted in Figs. 14(a), 14(b) and 14(c), where we particularly note that the one-hop flow (11$\rightarrow$5) dominates the wireless resource in 802.11 and O-DCF whereas A-DCF gives priority to the multi-hop flows, and thus significantly enhances fairness.



(a) Average throughputs of 3 flows (b) Average throughputs of 4 flows (c) Average throughputs of 5 flows (d) Jain's fairness index

Fig. 13. Experimental results: WLAN



(a) Average throughputs of 3 flows (b) Average throughputs of 4 flows (c) Average throughputs of 5 flows (d) Jain's fairness index

Fig. 14. Experimental results: Mesh

VI. Conclusion

In the theory community, Optimal CSMA was recognized as a valuable piece of work, which is the first fully distributed, no-message passing throughput-, and fairness-optimal MAC scheduling scheme. It remained to show that such a theory can be transferred to practice. O-DCF first showed the possibility of realizing the philosophy of Optimal CSMA in practice, but still with problems when working with TCP. We believe that A-DCF in this paper has taken another important step. As one of our contribution we report our A-DCF device driver in [11], which is expected to motivate other related follow-up work on A-DCF and O-DCF.

References

[1] M. Garetto, T. Salonidis, and E. Knightly, "Modeling Per-flow Throughput and Capturing Starvation in CSMA Multi-Hop Wireless Networks," *IEEE/ACM Trans. Netw*, vol. 16, no. 4, pp. 864–877, 2008.

[2] X. Wang and K. Kar, "Throughput Modelling and Fairness Issues in CSMA/CA Based Ad-Hoc Networks," in *Proc. IEEE INFOCOM*, 2005.

[3] O. Gurewitz, V. Mancuso, J. Shi, and E. Knightly, "Measurement and Modeling of the Origins of Starvation of Congestion-Controlled Flows in Wireless Mesh Networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1832–1845, 2009.

[4] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED," in *Proc. ACM MobiCom*, 2003.

[5] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, Jun. 2010.

[6] S. Rajagopalan, D. Shah, and J. Shin, "Network Adiabatic Theorem: An Efficient Randomized Protocol for Contention Resolution," in *Proc. ACM SIGMETRICS*, 2009.

[7] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, "Towards Utility-Optimal Random Access Without Message Passing," *Wirel. Commu. Mob. Comp.*, vol. 10, no. 1, pp. 115–128, Jan. 2010.

[8] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks," in *Proc. IEEE INFOCOM*, 2010.

[9] P. Marbach and A. Eryilmaz, "A Backlog-Based CSMA Mechanism to Achieve Fairness and Throughput-optimality in Multihop Wireless Networks," in *Proc. Allerton Conf.*, 2008.

[10] J. Lee, H. Lee, Y. Yi, S. Chong, B. Nardelli, and M. Chiang, "Making 802.11 DCF Near-Optimal: Design, Implementation, and Evaluation," in *Proc. IEEE SECON*, 2013.

[11] "Optimal CSMA: From Theory to Practice." [Online]. Available: http://lanada.kaist.ac.kr/ocsma

[12] "ath9k," http://wireless.kernel.org/en/users/Drivers/ath9k.

[13] J. Lee, J. Lee, Y. Yi, S. Chong, A. Proutiere, and M. Chiang, "Implementing Utility-Optimal CSMA," in *Proc. Allerton Conf.*, 2009.

[14] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. Knightly, and M. Chiang, "Experimental Evaluation of Optimal CSMA," in *Proc. IEEE INFOCOM*, 2011.

[15] W. Chen, Y. Wang, M. Chen, and S. C. Liew, "On the Performance of TCP over Throughput-Optimal CSMA," in *Proc. IEEE IWQoS*, 2011.

[16] X. Wang, Z. Li, and J. Wu, "Joint TCP Congestion Control and CSMA Scheduling without Message Passing," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6194–6204, 2013.

[17] B. Ji, C. Joo, and N. B. Shroff, "Delay-Based Back-Pressure Scheduling in Multihop Wireless Networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1539–1552, 2013.

[18] A. Warrier, S. Janakiraman, S. Ha, and I. Rhee, "DiffQ: Practical Differential Backlog Congestion Control for Wireless Networks," in *Proc. IEEE INFOCOM*, 2009.

[19] A. Aziz, D. Starobinski, P. Thiran, and A. Fawal, "EZ-Flow: Removing Turbulence in IEEE 802.11 Wireless Mesh Networks without Message Passing," in *Proc. ACM CoNEXT*, 2009.

[20] R. Laufer, T. Salonidis, H. Lundgren, and P. L. Guyadec, "XPRESS: A Cross-Layer Backpressure Architecture for Wireless Multi-Hop Networks," in *Proc. ACM MobiCom*, 2011.

[21] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," in *Proc. ACM SIGCOMM*, 1994.

[22] S.-J. Han, T. Nandagopal, Y. Bejerano, and H.-G. Choi, "Analysis of Spatial Unfairness in Wireless LANs," in *Proc. IEEE INFOCOM*, 2009.

[23] Y. Jian and S. Chen, "Can CSMA/CA Networks Be Made Fair?" in *Proc. ACM MobiCom*, 2008.

[24] A. Jindal and K. Psounis, "The Achievable Rate Region of 802.11-Scheduled Multihop Networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1118–1131, Aug. 2009.

[25] F. Calì, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 785–799, Dec. 2000.

[26] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs," in *Proc. ACM SIGCOMM*, 2005.

[27] V. A. Siris and G. Stamatakis, "Optimal CWmin Selection for Achieving Proportional Fairness in Multi-Rate 802.11e WLANs: Test-bed Implementation and Evaluation," in *Proc. ACM WinTECH*, 2006.

[28] Y. Grunenberger, M. Heusse, F. Rousseau, and A. Duda, "Experience with an Implementation of the Idle Sense Wireless Access Method," in *Proc. ACM CoNEXT*, 2007.

[29] Y. Kim, S. Choi, K. Jang, and H. Hwang, "Throughput Enhancement of IEEE 802.11 WLAN via Frame Aggregation," in *Proc. VTC-Fall*, 2004.

[30] W. Kim, H. K. Wright, and S. M. Nettles, "Improving the Performance of Multi-hop Wireless Networks using Frame Aggregation and Broadcast for TCP ACKs," in *Proc. ACM CoNEXT*, 2008.

[31] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier sense Multiple Access with Collision Notification," in *Proc. ACM MobiCom*, 2010.

[32] T. Li, M. K. Han, A. Bhartia, L. Qiu, E. Rozner, Y. Zhang, and B. Zarikoff, "CRMA: Collision-Resistant Multiple Access," in *Proc. ACM MobiCom*, 2011.

[33] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?" *IEEE Commun. Mag.*, vol. 39, no. 6, pp. 130–137, 2001.

[34] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756–769, 1997.