Synchronized CSMA Contention: Model, Implementation and Evaluation

Jingpu Shi[1], Ehsan Aryafar[1], Theodoros Salonidis[2], and Edward W. Knightly[1]
[1]Rice University, Houston, TX [2]Thomson, Paris, France

***Abstract*— A class of CSMA protocols used in a broad range of wireless applications uses synchronized contention where nodes periodically contend at intervals of fixed duration. While several models exist for asynchronous CSMA contention used in protocols like IEEE 802.11 MAC, no model exists for synchronized CSMA contention that also incorporates realistic factors like clock drifts. In this paper, we introduce a model that quantifies the interplay of clock drifts with contention window size, control packet size, and carrier sense regulated by usage of guard time. Using an FPGA-based MAC protocol implementation and controlled experiments on a wireless testbed we evaluate the model predictions on the isolated and combined impact of these key performance factors to per-flow throughput and fairness properties in both single-hop and multi-hop networks. Our model and experimental evaluation reveal conditions on protocol parameters under which the throughput of certain flows can exponentially decrease; while at the same time, it enables solutions that can offset such problems in a predictable manner.**

I. Introduction

Synchronized CSMA protocols partition time into periodic cycles of fixed duration. At the beginning of each cycle, nodes contend on a common channel using carrier sense and control packet handshake similar to RTS/CTS; after winning contention, nodes transmit until the end of the cycle. Synchronized CSMA contention has been used in both single-hop and multi-hop wireless networks for a wide variety of tasks. In single-hop networks, the IEEE 802.11 WLAN standard [1] supports synchronized contention in power-saving mode such that hosts wake up periodically and contend within a short window using beacons. In multi-hop networks, synchronized contention has been used for channel selection [18] and per-channel contention [3] in multi-channel networks, for leader election and power saving in sensor networks [7], [20], and for antenna selection in MIMO networks [14].

In practice, clock drifts can range from sub-microseconds to several milli-seconds, depending on the clock synchronization protocol's trade off between clock accuracy and cost or complexity [10], [16]. Drift introduces a clock phase bias in synchronized CSMA protocols because flows starting contention earlier than others may have higher chances to win the medium. In addition, multi-hop systems introduce topological bias where some flows compete with asymmetric channel state. This paper is the first to systematically and comprehensively investigate via modeling and experimental evaluation the joint impact of MAC protocol parameters and both types of bias on the throughput performance and fairness properties of synchronized CSMA systems. Our contributions are as follows.

First, we introduce and implement S-CSMA, a protocol that captures the basic features of synchronized CSMA contention. Our implementation on an FPGA wireless platform enables realistic evaluation of S-CSMA and fine-grained control of the main factors that affect its fairness properties: *contention window size*, *control packet size*, *clock drift* (to control the degree of imperfect synchronization) and *guard time* at the end of each contention period (to control the impact of carrier sensing).

Second, we introduce a Markovian model for throughput prediction in single-hop S-CSMA networks (e.g. synchronized WLANs) that incorporates clock phase drifts. Existing Markovian models for *asynchronous* CSMA protocols such as 802.11 [4], [5], [9], [11], [13], [15], [17], [8] can model stochastic contention instants due to carrier sense but are not suitable for periodic contention. Our model is different in nature and exploits the periodic structure of synchronized contention. Using the model and experimental evaluation, we show that flows with the earliest (latest) clock phases receive maximum (minimum) throughput and that the throughput of latest flows decreases sharply with clock phase drift. Surprisingly, these phenomena occur irrespective of guard time usage.

Third, we consider multi-hop networks where, in addition to their clock phase differences, flows compete with an asymmetric view of channel state and can suffer from lack of transmission opportunities due to carrier sensing. We extend our Markovian model to basic representative multi-hop scenarios. Our model and experimental testbed evaluation show that, unlike the single-hop case, flows with the earliest clocks do not necessarily receive the highest throughput. Instead, they may even receive zero throughput under certain conditions that we identify. For both guard-time and no-guard-time systems, we derive simple relationships among contention window sizes and clock phases that guarantee starvation-free operation. According to these relationships, in no-guard-time systems each flow requires clock phase knowledge of two-hop neighbors; while in guard-time systems, one-hop clock phase knowledge is sufficient. This implies different requirements on the phase bounds provided by the clock synchronization mechanism running in the network.

Finally, we consider arbitrary topologies and introduce an approximate model that lower-bounds throughput for guard-time systems. This model decouples the joint effect of interfering flows and allows to express the throughput of each flow as a simple function of its own contention window size and the *harmonic* mean of the contention window sizes of its one-hop interferers. Using the model and experiments we show that (i) the throughput of a flow with a topology disadvantage *exponentially* decreases with control packet size of its one-hop interferers (ii) the disadvantage of such a flow can only be offset by making one-hop interfering flows less aggressive through increasing the *harmonic mean* of their contention windows and (iii) global fairness objectives such as max-min fairness may require large contention window sizes for all flows, which can lead to increased network delays.

In all scenarios we considered, our experimental evaluation demonstrates that the model is effective in its predictions despite working in real hardware and under the variations of the wireless environment.

The paper is organized as follows. In Section II, we introduce and implement S-CSMA and describe our experimental methodology. In Section III, we introduce the Markovian model for single-hop networks and investigate the impact of imperfect synchronization. In Section IV we extend this model to representative multi-hop scenarios and investigate the impact of carrier sense. In Section V we model arbitrary topologies for guard-time systems, investigate the impact of interference on the resulting throughput approximation and the impact of fairness objectives on contention window adjustment mechanisms. Section VI concludes.

II. Synchronized CSMA (S-CSMA)

The detailed operations of synchronized CSMA protocols are all different from each other and, as a result, there is no single protocol model that characterizes all the operational details of these protocols. To analyze the basic nature of synchronized contention and capture its fundamental fairness properties, in this section we present a simple Synchronized CSMA protocol, which we call S-CSMA. Our protocol is not designed to capture all the details of this protocol family nor for improving performance. It is designed to be simple but characterize the basic use of synchronized contention. The principle and methodology of our analysis can be adapted to the analysis of a particular protocol.

A. Protocol description

S-CSMA is a single-channel synchronized CSMA protocol where time is partitioned into fixed-duration cycles. As shown in Fig. 1, each cycle consists of a contention phase followed by a data transmission phase. At the beginning of each cycle, each node i starts contending for the channel by sensing the medium. Once the medium becomes idle, the node computes a random backoff counter based on an initial contention window W_i and starts counting down mini-slots while sensing the medium. If during this time a transmission is sensed, the node quits contention and waits for the next cycle. If the backoff timer expires, the node transmits a short request (REQ) control packet and waits for a short grant (GNT) control packet. If the GNT is received, the data phase begins *immediately* and the node transmits data until the end of the cycle. The data transmission phase may consist of one or more data-link frames and their corresponding acknowledgments. If GNT is not received, the node doubles its contention window, computes a new random back-off counter and contends again. The node will stop contending either when GNT is received or at the end of the contention phase. If the end of the contention phase is reached, the node resets its contention window to W_i, quits contention and waits for the next contention cycle.

Since the node clocks are not perfectly synchronized, *leading* flows whose transmitters begin contention earlier at each cycle have a *phase advantage* over *lagging flows*. Carrier sensing is known to play a fundamental role in affecting the fairness properties of asynchronous CSMA protocols such as IEEE 802.11 [9]. In synchronized protocols, it is also an important factor that may either alleviate or aggravate the *phase bias* introduced by clock drifts.

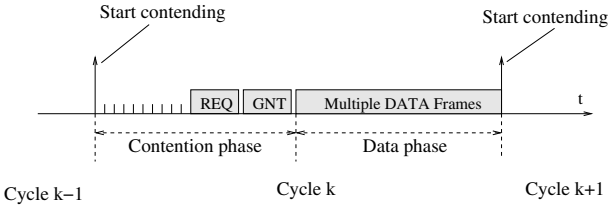


Fig. 1. Basic operations of S-CSMA

In addition to phase bias, a multi-hop S-CSMA network exhibits *topological bias*. More specifically, a flow A has a topological advantage with respect to flow B if the transmitter of A is within range of the receiver of B but not its transmitter. Being a hidden terminal, the advantaged flow can be blocked only when its transmitter senses the GNT packet of the disadvantaged flow receiver. Thus, the disadvantaged flow can win contention only if it completes random backoff count-down plus REQ packet transmission before the advantaged flow completes its random backoff count-down. The degree of topological bias is essentially determined by the duration of the REQ control packet.

A system design alternative is to employ a *guard time* at the end of the data phase. In such a guard-time system, upon finishing data transmission, each flow waits for an additional duration T_{gb} before starting contention for the next cycle. Since T_{gb} typically exceeds the maximum clock drift in the network, a guard-time system may avoid starvation problems due to carrier sensing transmissions of previous cycle. On the other hand, it may eliminate potential opportunities to alleviate the phase bias.

Phase bias, topological bias, carrier sense and the use of guard time are tightly coupled and can dramatically affect the fairness properties of multi-hop synchronized CSMA systems. In the next sections, we introduce models that treat such systems in a unified manner and address critical design issues such as (i) impact of phase bias and topological bias to per-flow throughput (ii) effectiveness of carrier sense in alleviating phase bias (iii) requirements for starvation-free clock synchronization mechanisms (iv) guidelines for contention window adjustment mechanisms to address unfairness due to both phase and topological bias.

B. Protocol implementation

We implemented S-CSMA using the Wireless Open-Access Research Platform (WARP) in Fig. 2. WARP was designed at Rice University[1] and is used for clean-slate MAC/PHY protocol design and implementation. Four components of the WARP platform are of interest to our implementation: (a) Xilinx Virtex-II Pro FPGA: This is the primary communication processor on the board. The FPGA PowerPC cores support a C-based embedded environment that facilitates MAC protocol development. They also interface to the physical layer implementation in the FPGA fabric. (b) Radio board: The RF board consists of interfaces with FPGA, a MAX2829 802.11 transceiver, RF frontend, and clock inputs. Up to four radio boards can be configured on a single platform for supporting applications such as MIMO (c) 10/100 Ethernet port: used to provide traffic source and sink and also offload experiment statistics, allowing for large-scale tests (d) Debug header pins:

[1] http://warp.rice.edu

can be configured as input or output for various applications such as receiving interrupt from an external source or raising an output signal for debugging purposes. We proceed to describe the main components of the S-CSMA implementation.

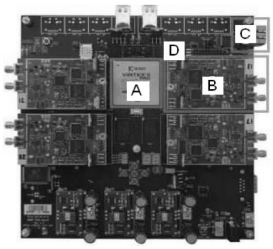


Fig. 2. WARP platform.

Synchronization. Our goal is to investigate the behavior of synchronized CSMA contention over a wide range of clock phase drifts. Hence, instead of implementing a particular wireless synchronization mechanism, we use an implementation that enables control of clock phase drifts at high granularity and over a wide range. In our implementation one node is responsible for announcing the beginning of each cycle to the rest of the nodes in the network. This synchronizer node has several dedicated output debug header pins connected through wire to an input debug header pin of each other node. The synchronizer node generates pulse signals on its output header pins. The rest of the nodes continuously monitor their input header pins and create a high priority interrupt whenever a rising edge is detected on their input header pins. This rising edge announces the beginning of the cycle. The synchronizer node determines the clock phase drift of each node by sending the corresponding pulse signals at different times. Our implementation approach allows to model the drift effect of any desired synchronization mechanism and control the clock phase drift on a per-node basis at $1\mu s$ accuracy.

Carrier Sense. WARP supports timers driven by 40MHz crystal oscillators within the FPGA fabric, which enable small mini-slot durations. We implemented S-CSMA with $20\mu s$ mini-slot duration for carrier sensing, also used by IEEE 802.11 b/g. In our implementation a carrier sense timer accepts a number of mini-slots and counts them down, provided that the medium is idle for the duration of mini-slot. If the medium becomes busy, the timer stops and only resumes counting down after a DIFS period.

Cycle elapsed time. Each node has a timer that computes the elapsed time since the beginning of the cycle. This timer is used for: (i) regulating REQ/GNT transmissions during the contention phase (ii) checking if a packet fits the cycle prior to its transmission and (iii) accounting for guard time duration in a guard-time system.

REQ/GNT handshake. Before transmitting a REQ packet, the contention window is set and the MAC counts down with the carrier sense timer. When this timer expires the node transmits the REQ packet.

Binary Exponential Backoff (BEB). If after a REQ transmission GNT is not received in a timeout, the sender doubles its contention window similar to the IEEE 802.11 BEB rule.

C. Experimental methodology

We conduct our experiments in a nine-node in-lab testbed. Each node consists of a laptop connected through Ethernet to a WARP board, connected to a 3dBi external antenna. The synchronizer node is connected to the rest of nodes using long low voltage wires. The boards operate at 5 GHz band and 12 Mbps data rate. Topology control is performed by adjusting the transmit power and node locations. WARP supports a wide range of transmission power levels where the gain in the transmit path of RF transceiver can be changed over a 60+ db range. Still, it has been challenging to perfectly realize the desired multi-hop topologies due to the in-lab testbed and varying wireless channel conditions.

Unless otherwise specified, the experiments use 200-byte data packets, 24-byte REQ/GNT packets, $30ms$ cycle length and $5ms$ contention phase. In a guard time system, the guard time is set to $1ms$ to account for the wide range of clock drifts investigated in the experiments. The laptops send backlogged UDP traffic using *iperf*. Each data point is an average of five experiments, each lasting for 200 seconds.

In our model and experimental evaluation we use per-flow success probability as throughput performance metric. Success probability is the time fraction a flow successfully reserves the channel at the beginning of each cycle. Since this metric captures fraction of transmission opportunities, it is independent of packet size, data rate, cycle duration and guard time duration.[2] We will use success probability and throughput interchangeably, both referring to the same metric.

III. Modeling single hop networks

We develop a discrete time Markov chain model to predict per-flow success probability in a single-hop network that employs S-CSMA. We account for temporal (dis)advantages introduced by different clock phases in our model, where the system state for a given cycle represents which flow transmits during this cycle. The transition probability is determined by relative clock phases of the flows and other system factors. This allows to relate clock phase to the stationary distribution of the system state. Using this analytical model, we investigate the joint effects of imperfect synchronization, carrier sense and guard time.

A. Model

We define a flow to be a transmitter-receiver pair. We use the terms contention window, backoff counter, and phase for flows instead of their transmitter nodes.

Let $t_i(k)$ be the time instant where the k-th cycle of flow i begins. Let T_c and T_d denote the duration of the contention phase and the data transmission phase, respectively. The fixed duration of a cycle is then given by $T = T_c + T_d + T_{gb}$, where T_{gb} is the duration of guard time. We characterize each flow by its clock phase θ_i with respect to an absolute global clock reference (or alternatively, the earliest clock in the network). Given the phase θ_i, the contention time instant of each node i for cycle k is $\theta_i + kT$. We assume that the clock frequencies of all flows are equal and constant and that the clock phase difference between any two flows can be at most θ_{max}. θ_{max} can be either known (as a maximum error of a clock synchronization mechanism) or unknown to the flows. We assume θ_{max} is much smaller than T, the duration of a cycle, and is smaller than T_{gb} in a guard-time system. We also

[2]These parameters are fixed during the protocol operation and can be used to determine throughput in Mbps given the packet success probability.

denote by $\theta_{ij} = \theta_i - \theta_j$ the relative phase between nodes i and j. All these quantities are expressed in mini-slots.

System state: Consider a fixed number N of contending flows, indexed by $1, 2, \ldots, N$, respectively. We let $b(k)$ denote the index of the flow accessing the channel at cycle k. We model the evolution of stochastic process $b(k)$ by a discrete time Markov chain, in which the state of the system in cycle k is $b(k)$, and the state space is $S = \{1, 2, \ldots, N\}$. The transition probability from state i to state j, denoted by $p_{ij} = P\{b(k) = j | b(k-1) = i\}$, is the probability flow j wins contention at cycle k given that flow i transmitted during cycle $k-1$. Note that the transition probabilities do not depend on k because in a synchronized CSMA system the flows refresh their contention state at the beginning of each cycle. Solving the Markov Chain, we obtain the stationary distribution $\boldsymbol{\pi} = \{\pi_i\}$, $\forall i \in S$, where π_i is the success probability of flow i.

Transition probability: We now compute the transition probabilities p_{ij} for both guard-time and no-guard-time systems. We denote by X_i the random backoff counter computed by each flow i at a contention cycle and use $p_i(x)$ to denote $P(X_i = x)$ and $\Phi_i(x)$ to denote $P(X_i > x)$.

No-guard-time system: Let i be the flow transmitting during cycle $k-1$. We divide all other flows into two sets, the leading set $LD(i) = \{m : \theta_m \leq \theta_i\}$ and the lagging set $LG(i) = \{m : \theta_m > \theta_i\}$. At the beginning of cycle k, flow i will sense idle medium and begin random backoff at its contention instant $t_i(k)$. Any flow m in leading set $LD(i)$ will also begin backoff at $t_i(k)$ because it will sense data transmissions of flow i during cycle $k-1$. On the other hand, any flow m in lagging set $LG(i)$ will start backoff at its later contention instant $t_m(k)$, provided neither flow i nor its leading flows have counted down their backoff counters to zero by that time instant.

Now, let j be the flow that wins contention at cycle k. If flow j is in set $\{i\} \cup LD(i)$, and its backoff counter $X_j = x$, it will win contention if *(i)* all other nodes in $\{i\} \cup LD(i)$ use a random backoff counter greater than x *(ii)* the random backoff counters of all nodes in lagging set $LG(i)$ expire after $t_i(k) + x$. On the other hand, if flow j is in the lagging set $LG(i)$, it will win contention only if the random backoff counters of all other nodes expire after $t_j(k) + x$. After taking expectation with respect to x, the transition probability p_{ij} is given by,

$$p_{ij} = \begin{cases} \sum_{x=0}^{W_j-1} p_j(x) \prod_{m:\theta_m \leq \theta_i, m \neq j} \Phi_m(x) \\ \qquad \times \prod_{m:\theta_m > \theta_i} \Phi_m(\theta_{jm} + x), & \theta_{ji} \leq 0 \\ \sum_{x=0}^{W_j-1} p_j(x) \prod_{m:\theta_m \leq \theta_i} \Phi_m(\theta_{ji} + x) \\ \qquad \times \prod_{m:\theta_m > \theta_i, m \neq j} \Phi_m(\theta_{jm} + x), & \theta_{ji} > 0 \end{cases} \tag{1}$$

Eq. (1) does not handle collisions, resulting in the sum of the transition probabilities out of a system state being slightly less than 1 ($\sum_j p_{ij} < 1$). To handle collisions, we add an additional collision state c to the system state space. The transition probability from state i to state c is $1 - \sum_j p_{ij}$. To compute p_{cj}, we assume the time it takes flows to detect and handle collision at the beginning of the cycle is larger than the clock phase differences. Since clock phases are absorbed, we assume each flow has equal access opportunity after a collision, i.e. p_{cj} is equal for all j. Our experiments have indicated that this simplifying assumption does not compromise the model's accuracy.

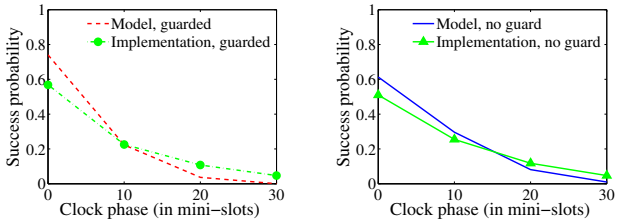
Guard-time system: When guard time is present, no node will sense a busy channel at its contention instant of cycle k and will immediately start random backoff. Since carrier sense at the beginning of each cycle has no effect, the transition probabilities p_{ij} are independent of which node i transmitted during the previous cycle. Therefore, the transition probabilities p_{ij} for the guard time system are given by:

$$p_{ij} = \sum_{x=0}^{W_j-1} p_j(x) \prod_{m:m \neq j} \Phi_m(\theta_{jm} + x). \tag{2}$$

Collisions are handled by adding a collision state, similar to the no-guard-time system.

B. Clock drift experimental investigation

We create a four-flow single-hop network in our testbed where all transmitters are within sensing range. Each flow i uses a contention window of $W_i = 32$ mini-slots and draws its backoff counter uniformly within this window in each contention cycle. The clock phases are fixed to $\theta_i = i \times 10$ mini-slots, $i = 0, .., 3$.



(a) Guard time (b) No guard time

Fig. 3. Model predictions and testbed measurements for a 4-flow single-hop network using S-CSMA.

We first consider a no-guard-time system. Fig. 3 depicts the measured and predicted throughput as a function of the flow clock phases sorted in non-decreasing order. We observe a close match between model and experiments. Also, throughput falls off sharply with the phase distance from the earliest clock, which can be explained by our model: Eq. (1) predicts that when the clock phase of flow j becomes larger, p_{ij} becomes smaller, and at the same time p_{ji} becomes larger for all i, thus rapidly decreasing the stationary probability of state j. The guard-time system, also depicted in Fig. 3, follows a similar trend as the no-guard-time system. Thus, different clock phases can lead to unfairness or even starvation, regardless of guard time usage.

IV. Analyzing the Role of Carrier Sense in S-CSMA Multihop Networks

In CSMA wireless networks, carrier sense can yield unfairness or even starvation when flows sense uncoordinated transmissions in their neighborhood. A representative scenario is Flow-in-the-Middle (FIM) [6], [9], [19], shown in Fig. 4. In asynchronous CSMA protocols like 802.11, the outer flows are not within sensing range of each other, their transmissions are not time-aligned. The middle flow continuously defers due to carrier sensing and can only contend in the small intervals

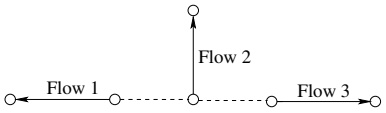


Fig. 4. Flow-in-the-middle (FIM) scenario.

where both outer flows are jointly idle. The result is very few transmission opportunities and very low throughput for the middle flow, even in a collision-free FIM scenario like Fig. 4 where each receiver is only within range of its transmitter. In contrast to asynchronous CSMA, the middle flow throughput under S-CSMA can vary from zero to maximum depending on the relative clock phases of the flows and their interaction with carrier sense. We extend the model of Section III and use the FIM scenario to analyze these interactions and also evaluate the option of disabling carrier sense using guard time.

A. Model

The key observation that simplifies the analysis of the FIM scenario is that when one of the outer flows wins contention in a cycle, the other outer flow will also transmit in the same cycle, while the middle flow will not transmit. On the other hand, when the middle flow wins, both outer flows defer transmission during this cycle. We use two states in the model: state 1 for the two outer flows and state 2 for the middle flow.

Without loss of generality we assume that flow 1 is the earlier outer flow ($\theta_1 \leq \theta_3$). We compute the transition probabilities p_{12} and p_{22} of middle flow 2 winning contention during cycle k, given that either the outer flows or the middle flow itself were transmitting during cycle $k-1$, respectively. Then the other two transition probabilities can be determined as: $p_{11} = 1 - p_{12}$ and $p_{21} = 1 - p_{22}$. Given the transition probabilities, the success probability of the middle flow is given by the stationary probability closed form expression of a two-state Markov chain:

$$\pi_2 = \frac{p_{12}}{1 + p_{12} - p_{22}} \tag{3}$$

while the success probability of each of the outer flows is $\pi_1 = 1 - \pi_2$.

Transition probability of guard-time system. When guard time is used, no flow senses busy carrier at the beginning of each cycle. Thus, in this case, the transition probabilities are computed similarly to the single-hop guard time case:

$$p_{i2} = \sum_{x=0}^{W_2-1} p_2(x)\Phi_1(x+\theta_{21})\Phi_3(x+\theta_{23}), \quad i = 1, 2 \tag{4}$$

Transition probability of no-guard-time system. The transition probabilities depend on how the clock phase of the middle flow is related to the phases of the outer flows. We consider the case where the middle flow 2 is leading both outer flows, i.e., $\theta_2 \leq \theta_1 \leq \theta_3$. To compute p_{12}, the transition probability of the middle flow winning the contention in cycle k given the two outer flows transmitted in cycle $k-1$, we need to determine the time instants when the three flows start their backoff counters. As shown in Fig. 5, the two outer flows sense an idle channel at their contention instants for cycle k and start their backoff counters immediately. However, due to carrier sense, flow 2 starts its backoff counter only when flow 3 finished its transmission for cycle $k-1$. Once the time instant for each flow to start its backoff counter is determined, the transition probability can be computed accordingly. The transition probabilities in different cases are summarized in Table I.

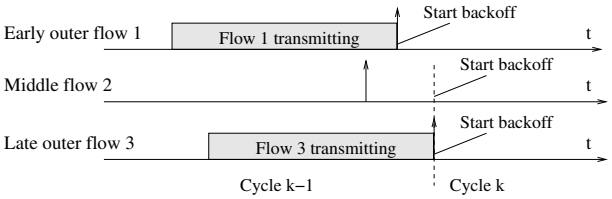


Fig. 5. Computation of p_{12}, the probability of middle flow 2 winning contention given the two outer flows transmitted in previous cycle. Arrows represent cycle boundaries.

B. Clock drift experimental investigation

According to the preceding analysis, the success probabilities depend on the relative phases of the three flows and whether guard time is used or not. In this section, we study the effect of these factors on success probability using the model predictions and experiments in out testbed. Unless otherwise specified, all flows use contention windows $W_1 = W_2 = W_3 = 32$ mini-slots.

Effect of relative phase of the outer flows θ_{31}. We set $\theta_2 = \theta_1 = 0$ and vary the phase θ_3 of the late outer flow 3 from 0 to 30 mini-slots, as shown in Fig. (6).

Guard time system. Fig. 7(a) shows a close match between measurements and model predictions. The middle flow 2 does not carrier sense at the beginning of each cycle and the late outer flow 3 becomes less competitive as its lag from the middle flow 2 increases. Thus, the success probability of the middle flow π_2 increases and stabilizes after the clock phase difference between the two outer flows θ_{31} exceeds 32 mini-slots. After this point, flow 2 only competes with the early outer flow 1; and flow 3 achieves equal throughput to flow 1, not due to its own contention effort, but because it transmits when flow 1 wins the contention.

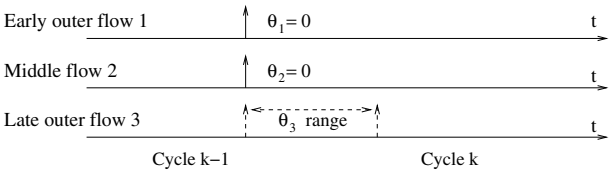


Fig. 6. Clock phase investigation: $\theta_2 = \theta_1 = 0$, $0 \leq \theta_3 \leq 30$, all in mini-slots.

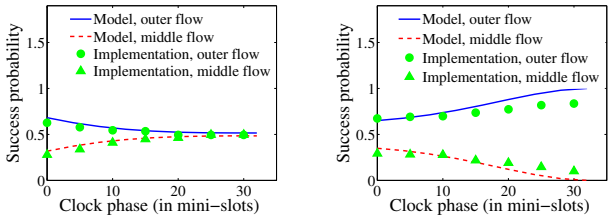


Fig. 7. FIM scenario, flow success probability as a function of θ_3.

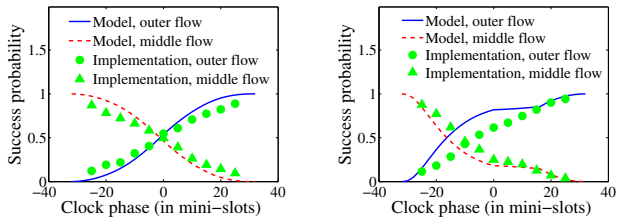
No-guard-time system. Fig. 7(b) shows that the effect of θ_{31} on the middle flow is reversed when guard time is not used.

	p_{12}	p_{22}
$\theta_2 \leq \theta_1$	$\sum_{x=0}^{W_2-1} p_2(x)\Phi_1(x+\theta_{31})\Phi_3(x)$	$\sum_{x=0}^{W_2-1} p_2(x)\Phi_1(x+\theta_{21})\Phi_3(x+\theta_{23})$
$\theta_1 \leq \theta_2 \leq \theta_3$	$\sum_{x=0}^{W_2-1} p_2(x)\Phi_1(x+\theta_{31})\Phi_3(x)$	$\sum_{x=0}^{W_2-1} p_2(x)\Phi_1(x)\Phi_3(x+\theta_{23})$
$\theta_2 \geq \theta_3$	$\sum_{x=0}^{W_2-1} p_2(x)\Phi_1(x+\theta_{21})\Phi_3(x+\theta_{23})$	$\sum_{x=0}^{W_2-1} p_2(x)\Phi_1(x)\Phi_3(x)$

TABLE I

MODEL FOR FIM SCENARIO IN NO GUARD TIME SYSTEM: TRANSITION PROBABILITIES INTO STATE 2.

As the phase θ_3 of the late outer flow 3 increases, the success probability of the middle flow decreases. This is because the middle flow would defer contention due to carrier sensing the data transmission of the late outer flow. At $\theta_3 > 32$ mini-slots, the middle flow receives zero throughput. In this case, flow 1 starts backoff immediately but flow 2 carrier senses the transmission of flow 3 and delays its contention for an amount of time greater than the contention window of flow 1.



(a) Guard time (b) No guard time

Fig. 8. FIM scenario, flow success probability as a function of θ_2.

Effect of middle flow phase θ_2. We now investigate the effect of phase θ_2 of the middle flow for a fixed relative phase θ_{31} of the outer flows. More specifically, contention windows for all flows are set to 32 mini-slots and $\theta_1 = 0, \theta_3 = 16$ mini-slots so that the middle flow starvation condition ($W_1 < \theta_{31}$) does not hold. The phase of the middle flow θ_2 varies from -30 to 30 mini-slots.

Guard-time system: Fig. 8(a) shows that the success probability π_2 of middle flow 2 decreases as its phase θ_2 increases with a smooth curve. Indeed, when guard time is used, the two transition probabilities to state 2 are equal, i.e., $p_{12} = p_{22}$, which implies that $\pi_2 = p_{12}$. Therefore, the two product terms of π_2 that depend on θ_2 appear in the nominator, resulting in a smooth curve.

No-guard-time system: The success probabilities are depicted in Fig. 8(b). We observe that when the middle flow 2 is leading both outer flows by more than W_2 mini-slots, flow 2 receives maximum throughput while the outer flows receive zero throughput. This is verified through experimental results of Fig. 8(b). Indeed, if the middle flow 2 transmits during cycle $k-1$, its backoff counter at cycle k will always expire earlier than the time instants when the outer flows start backoff. Hence, once the middle flow 2 wins contention at a cycle, it will continue to do so at every subsequent cycle. This is confirmed by Table I, where in this case, $p_{21} = 0$ and $p_{22} = 1$–state 2 of the Markov model is an absorbing state.

As the clock phase of the middle flow 2 increases, its success probability decreases rapidly to zero. This is because p_{12} decreases fast and p_{21} increases fast. When the phase of the middle flow 2 is within the phases of the outer flows, i.e., $\theta_1 \leq \theta_2 \leq \theta_3$, the success probability of flow 2 decreases slowly. This is because p_{22} (Table I) has only one product term that depends on θ_2 as opposed to having two terms in other cases. We call this region the *flat region*.

C. Summary and discussion

Overall, the model provides very good predictions of the experimental results. In some cases the model over-estimated the success probability of the middle flow, especially in the no-guard time system (Fig. 8(b) and Fig. 7(b)). We could identify two reasons. First, wireless channel variations occasionally caused the outer flows to interfere with each other which decreased their throughput. Second and most important, our model assumes that a winning flow fills the entire cycle with packets, while in our implementation packets are transmitted only if they fit the remaining part of the cycle. A close inspection of the packet traces showed that the last packets of an outer flow occasionally did not fit the cycle and the middle flow would sense idle channel, increasing its success probability.

Our analysis showed that in a no-guard-time system a (middle) flow can starve even if its clock is the earliest among the clocks within its vicinity. This is because some of its neighbors (outer flows) with very late clocks can significantly delay the time instant this flow starts to contend, resulting in some other neighbors always finishing contention earlier than this flow. In general, a flow will starve if the maximum phase among uncoordinated flows ("outer flows") within its vicinity exceeds the contention window of the early outer flow 1 ($\theta_{31} > W_1$). When this condition holds, this (middle) flow will receive zero throughput if its own contention window W_2 is greater than its relative phase to the early outer flow θ_{12}.

To avoid the above starvation phenomena a contention window adjustment solution should ensure that each node's contention window always exceeds the relative clock phase of both its one-hop and two-hop neighbors. Viewed from another angle, this also imposes stricter requirements for clock synchronization protocols to bound the clock phase within a two-hop instead of one-hop neighborhood of each node.

When guard time is used, the relative phases also play a role in throughput degradation. However, starvation is caused only if a flow is leading or lagging excessively with respect to all its outer flows. In the first case the flow causes starvation to the outer flows while in the second case it starves. This also holds in the no-guard-time system. Although the guard-time system does not inherently offer phase jitter protection,

success probability does not fall off as sharply as in regions other than the flat region of the no guard time system.

V. Modeling arbitrary topologies

Due to spatial reuse in a multi-hop network, several flows can transmit simultaneously during a cycle. Thus, our Markovian model can be naturally extended to the multi-hop case using one state for each such non-interfering set of flows. Then the success probability of each flow can be found by adding the stationary probabilities of the non-interfering flow sets it belongs. However, we do not pursue this approach due to *(i)* complexity of enumerating the independent sets and computing transition probabilities under imperfect synchronization and *(ii)* accurate prediction of per-flow success probability requires global information.

Instead, we introduce a model that computes a *lower bound* on the success probability of S-CSMA systems with guard time, based only on one-hop information. This approach not only simplifies analysis but also enables to directly study the factors that affect fairness properties and ultimately enable distributed algorithms that address unfairness by providing minimum throughput guarantees.

The key observation that enables development of such a model in a S-CSMA system with guard time is the fact that, at the beginning of each cycle, all flows (i) reset their contention windows to minimum, (ii) always sense an idle channel due to the use of guard time which exceeds the maximum clock drift in the network, and (iii) they compute their backoff counters independently. This allows finding a lower bound on the success probability of each flow by expressing the one-hop neighborhood interference of each flow in product form. This interference expression is a lower bound to the success probability of a flow, because in reality its one-hop interferers can *themselves* experience interference by the two-hop neighbors of this flow. The second critical step that reduces the model complexity and leads to a closed form expression is a classification of one-hop interferers according to their topological (dis)advantages.

A. Modeling success probability lower bound.

We use a simple geometric interference model characterized by the transmission range R_T and sensing range R_S. R_T is defined as the maximum distance that allows correct decoding of a packet. R_S ($R_S \geq R_T$) is defined as the maximum distance that triggers carrier sensing as well as the maximum distance that can cause collision due to simultaneous reception of more than one transmission. We define the one-hop interfering set L_i of flow i as the set of flows whose transmitter or receiver is within sensing range of either the transmitter or receiver of flow i.

We now consider a flow i in isolation and derive a lower bound b_i on its success probability π_i as a function of the flows in L_i. We divide the one-hop interfering set L_i of flow i in three subsets:

F_i: set of equivalent neighbor flows of flow i: Any flow j in this set would have equal success probability with flow i had they been contending in isolation using perfectly synchronized S-CSMA and equal contention windows. This set includes any flow j in L_i that has *(i)* a common transmitter or receiver node with flow i, *(ii)* a transmitter within sensing range of the flow i transmitter, *(iii)* a transmitter not in sensing range of the flow i transmitter but a receiver within sensing range of the flow i receiver, or *(iv)* a transmitter only within range of the flow i receiver and a receiver only within range of the flow i transmitter. We note that set F is symmetric, i.e. for any flow $j \in F_i$ it also holds that $i \in F_j$.

A_i: set of advantaged neighbor flows of flow i: Any flow j in this set would have higher success probability than flow i due to the REQ packet duration, had they been contending in isolation using perfectly synchronized S-CSMA and equal contention windows. Formally, this set includes all flows j in L_i for which neither transmitter nor receiver are within sensing range of the transmitter of flow i and whose transmitter is within range of the receiver of flow i. **D_i: set of disadvantaged neighbor flows of flow i**: For any flow j in this set, flow i is in A_j.

To compute b_i we assume flow i only contends with its one-hop neighbors. This is a pessimistic assumption for the success probability of flow i, because its one-hop neighbors can themselves experience interference from the two-hop neighbors of flow i (namely flows in L_j that are not in L_i). This can only increase the success probability of flow i.

At the beginning of each cycle, the backoff counters computed by different flows are independent with each other. Let the backoff counter of flow i at the beginning of a cycle be x_i mini-slots. With respect to the set F_i, flow i will win the contention only if all the equivalent flows' back-off counters are greater than x_i. With respect to A_i, flow i will win contention only if all the advantaged flows' back-off counters are greater than x_i plus REQ packet duration. Finally, with respect to D_i, flow i will win contention only if all the disadvantaged flows' back-off counters plus REQ packet duration are greater than x_i. Removing the condition on $X_i = x_i$ and incorporating relative phases, we reach the following expression for b_i:

$$b_i = \sum_{x_i=0}^{W_i-1} p_i(x_i) \prod_{f \in F_i} \Phi_f(x_i + \theta_{if}) \times \prod_{a \in A_i} \Phi_a(x_i + R + \theta_{ia}) \times \prod_{d \in D_i} \Phi_d(x_i - R + \theta_{id}) \quad (5)$$

Derived under a pessimistic assumption, b_i is a lower bound to the success probability π_i of each flow i. To investigate how b_i compares to π_i, we performed *ns* simulations using the same parameters as our implementation in 15-flow multi-hop arbitrary topologies deployed in an area of 1000m x 1000m. Fig. 9 depicts the results of a representative example, which plots π_i and b_i for each flow i, sorted in decreasing order of π_i. The lower bound curve b_i is below and follows the decreasing trend of the throughput π_i curve. In addition, the lower bound becomes tighter for flows with lower throughput. Thus, Eq. (5) reflects congestion conditions in the network and provides a better throughput approximation in highly congested regions.

B. Impact of REQ packet duration

Here we derive an approximate closed form of Eq. (5) to investigate the impact of REQ packet size on the success probability lower bound. Starting from Eq. (5), we first set all phase terms to zero to isolate the effect of REQ packet duration. We then consider an approximate backoff model where each backoff counter X_i is geometrically distributed

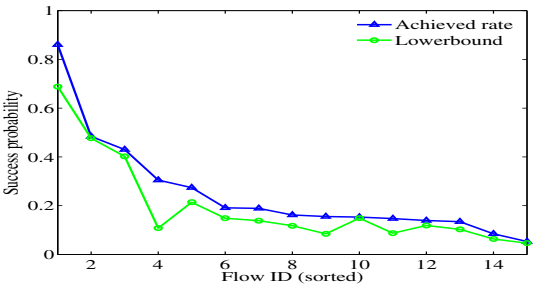


Fig. 9. Lower bound validation.

with parameter q_i ($= \frac{2}{W_i}$), instead of being uniformly distributed within [0,$W_i - 1$]. This approximation has also been successfully used in asynchronous CSMA protocol models [12], [17]. In the geometric backoff model, q_i is the probability that flow i transmits in the next mini-slot if its current window is W_i. Next, we further approximate Eq. (5) by its continuous form:

$$b_i = \int_{x=0}^{\infty} p_i^c(x_i) \prod_{f \in F_i} \Phi_f^c(x_i) \times \prod_{a \in A_i} \Phi_a^c(x_i + R) \times \prod_{d \in D_i} \Phi_d^c(x_i - R) dx_i, \quad (6)$$

where $p_i^c()$, $\Phi_i^c()$ are the continuous counterparts of $p_i()$,$\Phi_i()$, respectively.

Since X_i in discrete form is geometrically distributed with parameter $q_i = 2/W_i$, in continuous form it is exponentially distributed with parameter $\lambda_i = 2/W_i$. Substituting $p_i^c()$ and $\Phi_i^c()$ corresponding to this distribution in Eq. (6) and after algebraic manipulations, we reach the following closed form expression for the lower bound of flow i:

$$b_i = \frac{\lambda_i e^{-R(C_a - C_d)}}{\lambda_i + C_f + C_a + C_d} \quad (7)$$

where $C_f = \sum_{f \in F_i} \lambda_f$, $C_a = \sum_{a \in A_i} \lambda_a$, and $C_d = \sum_{d \in D_i} \lambda_d$. According to Eq. (7), b_i is jointly determined by the one-hop interferers of flow i in the set L_i, yet each subset F_i, A_i and D_i of L_i contributes differently. Next, we investigate the contribution of each individual subset.

Impact of equivalent interfering flows F_i: The expression for b_i considering only the flows in F_i is obtained by setting $D_i = \emptyset$,$A_i = \emptyset$ in Eq. (7):

$$b_i = \frac{\lambda_i}{C_f + \lambda_i} = \frac{1}{\frac{W_i |F_i|}{\widetilde{W}_f} + 1}. \quad (8)$$

Here, $|F_i|$ is the number of flows in F_i and $\widetilde{W}_f$ is the harmonic mean of the contention windows of the flows in F_i. [3]

Eq. (8) reveals that when flow i contends only with equivalent flows, b_i does not depend on the REQ packet duration. Eq. (8) also shows that the window ratio $W_i/\widetilde{W}_f$ can be used to control the throughput ratio between flow i and its interfering flows. To illustrate application to fairness concepts, we use a simple scenario where all interferers of flow i are *not* within range of each other. The interference relationships can be represented by a star-shaped flow contention graph centered at flow i (Fig. 10), where vertices correspond to flows and edges correspond to pair-wise interference among flows. If all flows are backlogged, the sum of their success probabilities within each clique of the flow contention graph should equal one.[4] Also, in this scenario, the success probability of each flow equals its lower bound. Therefore, $b_f = 1 - b_i$ for every interfering flow f of flow i.

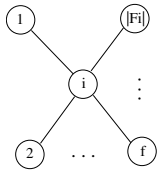


Fig. 10. Flow contention graph of flow i and $|F_i|$ independent interferers.

It is straightforward to show that a max-min fair allocation in the scenario of Fig. 10 would be $b_i = b_f = 1/2$. This can be achieved by $W_i/\widetilde{W}_f = 1/|F_i|$ in Eq. (8). On the other hand, under a proportionally fair allocation, flow i should receive $b_i = \frac{1}{|F_i|+1}$ and each interferer f should receive $b_f = \frac{|F_i|}{|F_i|+1}$. This can be achieved by $W_i/\widetilde{W}_f = 1$ in Eq. (8).

Impact of advantaged interfering flows A_i: The expression for b_i considering only the flows in A_i is obtained by setting $F_i = \emptyset$, $D_i = \emptyset$ in Eq. (7):

$$b_i = \frac{\lambda_i e^{-RC_a}}{C_a + \lambda_i} = \frac{e^{-\frac{2R|A_i|}{\widetilde{W}_a}}}{\frac{W_i |A_i|}{\widetilde{W}_a} + 1} \quad (9)$$

where $|A_i|$ is the number of flows in A_i and $\widetilde{W}_a$ is the harmonic mean of the contention windows of the flows in A_i. Similar to the case of set F_i, Eq. (9) shows that b_i depends on the ratio $W_i/\widetilde{W}_a$. However, b_i decreases exponentially with REQ packet duration R. The disadvantage due to the REQ packet duration cannot be addressed by flow i decreasing its contention window W_i (the exponential term persists even if W_i is zero). It can only be addressed by increasing $\widetilde{W}_a$.

We now rearrange terms in Eq. (9) and express W_i as a function of $\widetilde{W}_a$, R and b_i:

$$W_i = \frac{\widetilde{W}_a}{|A_i| b_i} (e^{-\frac{2R|A_i|}{\widetilde{W}_a}} - b_i) \quad (10)$$

Eq. (10) gives the contention window pairs $(W_i, \widetilde{W}_a)$ that achieve allocation b_i subject to REQ packet duration R and number of interferers $|A_i|$.

We experimentally validate equation (10) in our testbed, using a three-flow topology with flow i and two advantaged interfering flows ($|A_i| = 2$). We perform five experiments. In each experiment a window pair $(W_1, \widetilde{W}_a)$ is set to values predicted by Eq. (10) to achieve max-min fair b_i using $R = 3.2$ mini-slots (REQ size in our implementation). We then measure the success probabilities π_i and compute Jain's Fairness Index

[3] $\widetilde{W}_f = (\sum_{f \in F_i} |F_i|^{-1} W_f^{-1})^{-1}$.

[4] This holds because success probabilities reflect the number of flow transmission opportunities. In contrast, the sum of normalized throughputs would be less than one due to contention overhead.

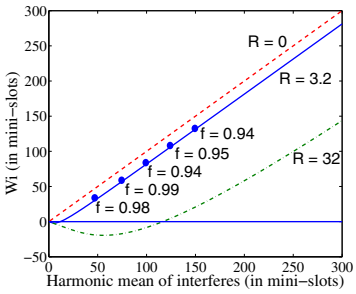


Fig. 11. Contention windows yielding max-min fair transmission for various values of REQ packet size. Jain's fairness index (f) of experimental results is calculated for the circled data points.

($f = \frac{(\sum_{i=1}^{i=n} p_i)^2}{n \times \sum_{i=1}^{i=n} (p_i)^2}$). The data points on the R=3.2 line in Fig. 11 show that, in all cases, the contention windows predicted by the model achieve throughput allocation very close to a max-min fair allocation (f=1.0).

Fig. 11 also plots Eq. (10) for various REQ packet sizes. For each curve, data points above the $W_i = 0$ line correspond to a feasible pair $(W_i, \widetilde{W}_a)$ that yield max-min fairness. For larger R, the required contention window sizes to achieve max-min fairness are higher. Larger contention windows require a larger cycle to absorb the overhead due to the contention phase; however, a larger cycle yields higher delays in the network.

When the relative phases θ_{if}, θ_{ia}, and θ_{id} are non-zero, the closed form expression of Eq. (5) can be found similar to the perfectly synchronized case. It can be shown that similar to REQ packet size, the *average* relative phase of interfering flows contributes exponentially to per flow throughput. For some cases the average relative phase is added to the REQ packet duration, whereas in others it cancels out the REQ effect. More details can be found in [2].

VI. Conclusions

In this paper, we analyzed the fairness properties of synchronized CSMA contention using an analytical model, an FPGA-based MAC protocol implementation, and experimental evaluation on a wireless testbed.

We showed that in single-hop systems, early flows achieve high throughput whereas the throughput of the late flows decreases sharply with clock drift, regardless of use of guard time. In multi-hop systems where clock phase differences are coupled with carrier sense and topological bias, we showed that carrier sense in no-guard-time systems can act as a protection mechanism against clock drifts. On the other hand, guard-time systems offer more predictability of throughput. Our model for arbitrary topologies in guard-time systems reveals that the throughput of a topologically disadvantaged flow can decrease exponentially with control packet size and average clock drift of its one-hop interferers. We showed that no-guard-time systems impose tighter requirements on the maximum phase provided by the network clock synchronization mechanism. This is because in a no-guard-time system, the contention window size of each flow must be greater than the phases of all its two-hop neighbors, whereas in a guard-time system, this condition is restricted to one-hop neighbors.

Our derived analytical relationships can provide a foundation for future work that targets achieving network-wide fairness through contention window adjustment. For example, a congestion control algorithm can exploit our results that (i) the exponential disadvantage of a flow due to REQ packet duration and average clock drift can only be addressed by increasing the harmonic mean of the contention windows of its one-hop interferers, and not by decreasing its own contention window; and (ii) increasing the *harmonic mean* requires very aggressive window increases for each individual interferer.

VII. Acknowledgement

This research was supported by NSF grants CNS-0751173 and CNS-0721894.

References

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE Std. 802.11-1999 edition*.

[2] E. Aryafar, J. Shi, T. Salonidis, and E. Knightly. Synchronized CSMA Contention: Model, Implementation and Evaluation. In *Technical Report, TREE0902*, Department of Electrical and Computer Engineering, Rice University.

[3] P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proc. ACM MobiCom*, New York, NY, USA, September 2004.

[4] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

[5] M. Carvalho and J. J. Garcia-Luna-Aceves. A scalable model for channel access protocols in multihop ad hoc networks. In *Proc. ACM MobiCom*, Philadelphia, PA, USA, September 2004.

[6] C. Chaudet, I. Guerin-Lassous, and B. Gaujal. Study of the impact of asymmetry and carrier sense mechanism in IEEE 802.11 multi-hops networks through a basic case. In *Proc. ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '04)*, Venice, Italy, 2004.

[7] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proc. ACM MobiCom*, Rome, Italy, July 2001.

[8] M. Durvy and P. Thiran. A packing approach to compare slotted and non-slotted medium access control. In *INFOCOM*, 2006.

[9] M. Garetto, T. Salonidis, and E. Knightly. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. In *Proc. IEEE INFOCOM*, Barcelona, Spain, 2006.

[10] G. Nguyen H.W. So and J. Walrand. Practical synchronization techniques for multi-channel MAC. In *Proc. ACM MobiCom*, Los Angeles, CA, USA, September 2006.

[11] A. Kashyap, S. Ganguly, and S. Das. A Measurement-based Approach to Modeling Link Capacity in 802.11-based Wireless networks. In *Proc. ACM MobiCom*, Montreal, Canada, Oct. 2007. ACM.

[12] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed point analysis of single cell IEEE 802.11 WLANs. In *Proc. IEEE INFOCOM*, Miami, FL, March 2005.

[13] K. Medepalli and F. Tobagi. Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications. In *Proc. IEEE INFOCOM*, Barcelona, Spain, 2006.

[14] M. Park, S. Choi, and S. Nettles. Cross-layer MAC design for wireless networks using MIMO. In *Proceedings of IEEE Globecom*, St. Louis, MO, 2005.

[15] L. Qiu, Y. Zhang, F. Wang, M. Han, and R. Mahajan. A general model of wireless interference. In *Proc. ACM MobiCom*, Montreal, Canada, Oct. 2007. ACM.

[16] K. Romer. Time synchronization in ad hoc networks. In *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 01)*, Long Beach, CA, 2001.

[17] G. Sharma, A. Ganesh, and P. Key. Performance analysis of contention based medium access control protocols. In *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006.

[18] J. So and N. Vaidya. Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *Proc. ACM MobiHoc*, Tokyo, Japan, May 2004.

[19] X. Wang and K. Kar. Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks. In *Proc. IEEE INFOCOM*, Miami,FL,USA, 2005.

[20] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.