PSMAC: Polling Service-based Medium Access Control for Wireless Networks

Yihan Li† Shiwen Mao† Shivendra Panwar‡
†Electrical and Computer Engineering Department, Auburn University, Auburn, AL
‡Electrical and Computer Engineering Department, Polytechnic University, Brooklyn, NY
Email: yli@ieee.org, smao@ieee.org, panwar@catt.poly.edu

***Abstract*— It has been shown in the literature that many MAC protocols for wireless networks, such as the IEEE 802.11 MAC, have a considerable control overhead, which limits their achievable throughput and delay performance. In this paper, we study the problem of improving the efficiency of MAC protocols. We first analyze the popular p-Persistent CSMA scheme which does not achieve 100% throughput. Motivated by insights from polling system theory, we then present three polling service-based MAC schemes, termed PSMAC, for improved performance. The main idea is to serve multiple data frames after a successful contention resolution, thus amortizing the high control overhead and making the protocols more efficient. We present analysis and simulation studies of the proposed schemes. Our results show that the proposed algorithms can effectively improve the throughput and delay performance of p-Persistent CSMA, as well as providing energy savings. The proposed schemes are more efficient for handling bursty traffic typically found in wireless networks. Finally, we observe that the proposed PSMAC schemes significantly outperform p-Persistent CSMA with respect to fairness.**

I. Introduction

Despite the recent advances in wireless technologies, today's wireless networks still cannot offer comparable data rates as their wired counterparts. In such networks, medium access control (MAC) protocols play an important role in coordinating channel access among the wireless terminals. In order to accommodate existing and emerging bandwidth-intensive applications, it is important to improve the efficiency of wireless MAC protocols, while adopting new physical layer technologies to obtain higher channel data rates.

Over the years, many MAC protocols have been proposed for wireless networks, such as ALOHA, Slotted ALOHA, carrier sense multiple access (CSMA), and CSMA with collision avoidance (CSMA/CA). The CSMA/CA-like IEEE 802.11 MAC has become the most popular protocol for single- or multi-hop wireless networks. However, the IEEE 802.11 MAC has a considerable control overhead. For example, Xiao and Rosdahl [1] show that the maximum achievable throughput for IEEE 802.11a is 24.7 Mb/s, which is about 45.7% of the nominal link capacity. In [2], Woo and Culler find that the RTS-CTS-DATA-ACK handshake required in transmitting a packet can constitute up to 40% overhead in a sensor platform. When used in a multi-hop environment, the problem gets even worse due to the carrier sensing and spatial reuse issues, making it hard to provide sufficient end-to-end throughput for paths with a large number of hops [3]. In addition, the interesting study in [1] shows that by simply increasing the data rate without reducing overhead, the enhanced performance is bounded even when the data rate goes to infinity. It is therefore crucial to reduce the control overhead of such wireless MAC protocols.

In this paper, we study the problem of improving the efficiency of wireless MAC protocols. For simplicity, we consider a single-hop ad hoc network, where all nodes can hear and directly communicate with each other.[1] We first examine the popular *reservation-based p-Persistent CSMA scheme* (called p-Persistent CSMA in this paper), which uses RTS/CTS for contention resolution and the p-Persistent scheme when sending RTS frames. This scheme differs from the standard IEEE 802.11 protocol only in the selection of backoff interval. Instead of the *binary exponential backoff* used in the standard, a backoff interval sampled from a geometric distribution with parameter p is used. Its performance has also been shown to closely approximate the standard protocol if the average backoff intervals are the same (at least from the perspective of protocol capacity) [4], [5]. We show that this scheme uses a limited-1 polling service [6] (since after each contention period, only one data frame is served) which does not achieve 100% throughput for the network.

We propose to use gated or exhaustive polling service for medium access. Polling is a general way of multiplexing service requests for a single server from multiple stations [6]. In a polling system, incoming requests are buffered at each station and are served by the server according to certain order (e.g., cyclic or random). There are three types of service policies for a polling system: (i) *exhaustive policy*, where the server serves a station until its buffer is emptied; (ii) *gated policy*, where the server serves for a station only those requests which are already buffered in the station when this service period begins; and (iii) *limited-k policy*, within which a station is served until either the buffer is emptied or the first k buffered requests are served, whichever occurs first. One special case of limited-k service is the limited-1 service, where at most one request is served during each service period (as in most existing MAC protocols). It has been shown that both exhaustive service and gated service are more efficient than limited-k service, and they can guarantee bounded delay as long as the offered load is strictly less than 100% [6], [7].

Motivated by insights from polling system theory [6], we

[1] We discuss how to extend this work to multi-hop wireless networks in Section III-D.

present three polling service-based MAC scheduling algorithms, termed PSMAC, for improved performance. The main idea is to serve multiple data frames after a successful contention resolution, thus amortizing the high control overhead over multiple data frames and making the protocols more efficient. Specifically, PSMAC Algorithm 1 uses the same p-Persistent strategy in sending RTS frames for contention resolution, but a winning node will use a gated service or exhaustive service to serve its queue. An improvement of Algorithm 1, PSMAC Algorithm 2, maintains multiple virtual queues, one for each of its neighbors, and the gated service or exhaustive service is used for one of the non-empty virtual queues at a winning node. In this way, those nodes that are not involved in the current service can be scheduled to sleep, thus achieving energy savings. A further improvement, PSMAC Algorithm 3, combines the strengths of the first two algorithms. It also maintains multiple virtual queues as in Algorithm 2, but when a node wins the channel, it will use gated or exhaustive service to serve all its non-empty virtual queues, one at a time. Thus it has the high efficiency of Algorithm 1, and the capability of sleep-scheduling for energy savings as in Algorithm 2.

We provide a random polling-based analysis of Algorithm 1 that provides a tight estimate for the achievable average delay. We also present extensive simulation studies of the proposed algorithms under various traffic models. Our analysis and simulation results show that all the three proposed algorithms achieve considerable throughput and delay improvements over p-Persistent CSMA. In addition, Algorithms 2 and 3 can achieve significant energy savings by allowing node sleep-scheduling. We also find that the proposed schemes are more efficient for handling bursty traffic, which are typical in wireless networks [8]. Specifically, when traffic gets burstier, all the three PSMAC algorithms achieve a similar delay performance, and the gains over p-Persistent CSMA is larger than that under the i.i.d. Bernoulli traffic. Another interesting observation from our simulation results is that, surprisingly, such performance gains can be achieved without sacrificing fairness performance. All the three PSMAC schemes achieve better fairness performance than p-Persistent MAC. Finally, we show that the proposed algorithms can be extended to multi-channel wireless networks and Algorithm 2 can be used in a multi-hop wireless network. We expect similar performance gains in these environments.

The remainder of this paper is organized as follows. In section II, we provide an throughput analysis for the p-Persistent CSMA scheme. We then present three novel polling service-based MAC schemes in Section III and our simulation and analysis performance studies in IV. We discuss related work in Section V and Section VI concludes this paper.

II. Throughput Analysis of p-Persistent CSMA

In this section, we provide an analysis of the throughput performance of p-Persistent CSMA (due to its similarity to the IEEE 802.11 MAC [4], [5]). The purpose is to provide a proper benchmark for the performance of the proposed schemes.

We consider a slotted-time system throughout this paper, where each time slot is the combined transmission time of an RTS and a CTS frame. Frame transmissions are aligned to the beginning of the time slots. In p-Persistent CSMA, a nonempty node, say Node A, first senses the medium at the beginning of the next time slot. If the medium is idle, Node A will transmit an RTS with probability p in the first half of the time slot. In the RTS, node A specifies the destination of its head of line frame, say Node B. If this is the only RTS sent in that time slot, there is no collision and Node B will reply with a CTS in the second half of the time slot; otherwise, there is collision of multiple RTS's and no CTS will be transmitted. We assume that each frame has the same length of L time slots. If the RTS/CTS dialog is successful, a data frame will be transmitted in the following L time slots, right after the time slot of the successful RTS/CTS dialog. The operation of this mechanism is illustrated in Figure 1.[2] Note that it is similar to the *limited-1 service* in a polling system, since only one frame is transmitted in every service period.

For such random access networks, it is more interesting to study the system under heavy load for performance limits. Therefore, we make the *heavy traffic assumption* in the following analysis. That is, we assume each node has at least one data frame to transmit at any time. Letting Q be the probability that only one RTS is sent in a time slot, we have

$$Q = Np(1-p)^{N-1}. \tag{1}$$

Let S be the time measured from the time when the previous service finishes to the time when one pair of RTS/CTS succeeds (see Figure 1). We have that $Pr(S = k) = Q(1-Q)^{k-1}$, and the average of S is

$$\mathrm{E}(S) = \sum_{k=1}^{\infty} kQ(1-Q)^{k-1} = \frac{1}{Q}. \tag{2}$$

The operation of p-Persistent CSMA can be modeled as a *alternating renewal process* [9]: every contention period of S will be followed by a service period of L, and S is the overhead for transmitting the L time slots of data. Therefore, the average throughput of the system can be calculated by [9]

$$T = \frac{L}{L + E(S)}. \tag{3}$$

For a given network of N nodes, T is a function of the transmission probability p. To achieve a high throughput, it is important to select a proper value for p. We can set the first derivative of T with respect to p to 0, and obtain the optimal value as

$$p^* = \frac{1}{N}. \tag{4}$$

With some algebra, it can be verified that p^* is a maximizer by checking the the second derivative of T with respect to p. By setting $p^* = 1/N$, we can derive $T^*(N)$, the maximum throughput for each N as

$$T^*(N) = \frac{L(1-\frac{1}{N})^{N-1}}{L(1-\frac{1}{N})^{N-1} + 1}. \tag{5}$$

[2]For simplicity, we ignore the protocol components such as Inter-Frame-Spaces (IFS) and ACK frames that are used as in the IEEE 802.11 MAC. However, these components can be modeled as a fixed amount of overhead and can be easily incorporated into the model (e.g., adding an ACK time slot after each data transmission in Figure 1).

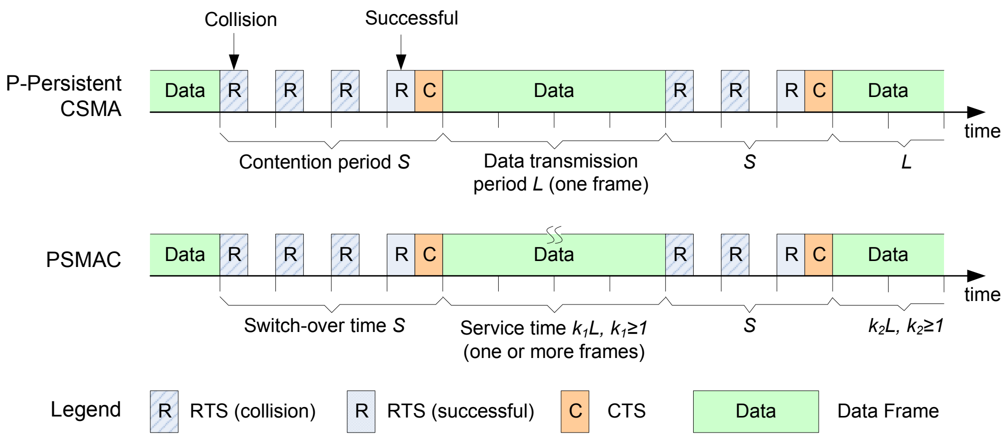


Fig. 1. Time-line illustration of p-Persistent CSMA and the proposed algorithms.

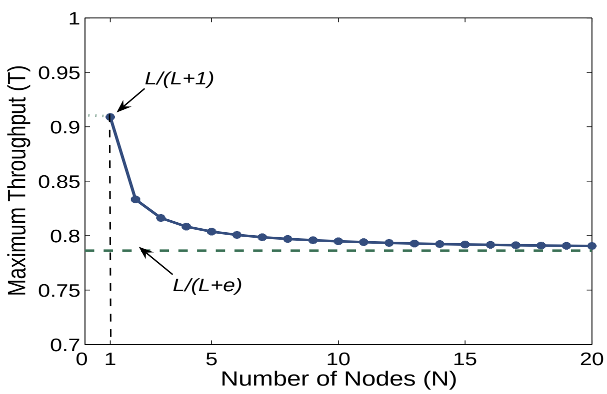


Fig. 2. The maximum throughput of p-Persistent CSMA ($L = 10$).

We plot $T^*(N)$ versus N in Figure 2 for the case $L = 10$. It can be seen that, in the trivial case of a single user network (all frames will be lost since there is no receiver), every RTS will succeed and the throughput achieves its maximum of $T^*(1) = L/(L+1)$. As the number of nodes increases, the maximum throughput decreases due to the higher chance of collision. When $N \to \infty$, the maximum throughput approaches to a limit of $T^*(\infty) = L/(L+e)$ (from above), where e is Euler's number. For example, if $L = 10$ and $N = 20$, the maximum throughput is 79%; when $N \to \infty$, the throughput reaches its limit of 78.6%. The p-Persistent CSMA scheme clearly does not achieve 100% throughput.

Equations (4) and (5) can also be interpreted as follows. Letting M be the number of nodes sending RTS in a time slot, we have $\Pr(M = m) = C_n^k p^m (1-p)^{N-m}$, and the average of M is $\mathrm{E}(M) = \sum_{m=1}^{N} m C_n^k p^m (1-p)^{N-m} = Np$, where C_n^k is the *choose function* n choose k. The optimal value p^* ensures that, on average, there is only one transmitter in each competition slot (i.e., $\mathrm{E}(M) = 1$), which leads to the highest throughput. We will use $p^* = 1/N$ throughout this paper.

It is worth noting that the maximum throughput (5) is also a function of frame length L. It is possible to achieve a reasonably high throughput with p-Persistent CSMA by increasing the frame length. For example, the sum of RTS and CTS is 34 Bytes in the IEEE 802.11 MAC. If a data frame is 1000 Bytes long, we have that $L \approx 30$ and $T^* \in [91.7\%, 96.8\%]$. However, dynamic rate adaptation is usually used in IEEE 802.11 [1], where data frames are transmitted at a high bit rate (e.g., 11 Mb/s) and control frames are transmitted at a low bit rate (e.g., 1 Mb/s). In this case, we have that $L \approx 3$ and $T^* \in [52.4\%, 75\%]$. Another question is, can we reduce the protocol overhead by disabling the RTS/CTS mechanism? If data frames are directly used in contention resolution, the system reduces to Slotted ALOHA. Its throughput is Q as given in (1). It can be verified that $p^* = 1/N$ is a maximizer of Q, and the the optimal throughput is $T^* = 1/e = 36.8\%$ as $N \to \infty$, the well-known upper bound for Slotted ALOHA, which may be even lower than (5).

III. Polling Service-Based MAC Algorithms

It is well-known that in a polling system, exhaustive service has the highest efficiency, limited-1 service has the lowest efficiency, and the efficiency of gated service lies in between [6]. The p-Persistent CSMA scheme uses limited-1 service and has a limited throughput. We will show in the following that when exhaustive service or gated service is used, the throughput can be significantly improved. This is also intuitive, since if more than one frames are served continuously after a successful RTS/CTS dialogue, the overhead of the contention period can be amortized and the system will be more efficient.

In this section, we present three polling service-based MAC schemes that can improve the throughput, delay, and energy performance for p-Persistent CSMA-like networks. In Section IV, we will show that these improvements are achieved without sacrificing the fairness performance.

A. PSMAC Algorithm 1

The first polling service-based algorithm, termed PSMAC Algorithm 1, incorporates a *gated service* for frame scheduling [6]. More specifically, all arriving frames (transit or locally generated) are queued in a common transmission buffer (or, the *global queue*). Nodes send out RTS/CTS as in p-Persistent CSMA. In the RTS, the source node specifies the destination MAC address of its *head-of-line* frame, say Node B. If this is the only RTS in the time slot, Node B will return a CTS. In the RTS, Node A will also specify how many packets will be transmitted, so that all other nodes will get this information. However, instead of sending one data frame after a successful RTS/CTS pair, all frames that have arrived at the source node before the RTS transmission, will be served back-to-back in the following slots. Other nodes will start a new round of competition (using the p-persistent method) when the current sequence of frame transmissions is over. The operation of Algorithm 1 is illustrated in Figure 1.

Alternatively, *exhaustive service* can also be incorporated into this algorithm [6], which is generally more efficient than gated service in a polling system. However, with exhaustive service, the source node does not know when the corresponding transmission will be over when it sends an RTS, since there may be new frame arrivals after the transmission starts. According to the exhaustive service policy, such frames will also be served during this service period. When the source node clears its buffer, it should send a special control frame in the following time slot to notify all its neighbors, which will start a new round of competition in the next time slot. We found that gated service and exhaustive service achieve very similar performance. Therefore we only present the analysis and simulations for gated service in this paper for brevity.

1) Delay Performance: With Algorithm 1, the network can be modeled as a *gated service random polling system*. The average delay of Algorithm 1 under uniform i.i.d. traffic can be analyzed as follows. By abuse of notation, we also let S denote the *switch-over time*, which is the time between two consecutive service periods in a polling system. It is measured from the time when the previous service is over, to the time when the next service starts. Since every node sends RTS/CTS as in p-Persistent CSMA (see Section II), the average switch-over time is identical to the average contention period in p-Persistent CSMA, with a geometric distribution. Its mean is given in (2), and its second moment is

$$E(S^2) = \sum_{k=1}^{\infty} k^2 Q(1-Q)^{k-1} = \frac{2-Q}{Q^2}. \qquad (6)$$

The average delay of a fully symmetric random gated service polling system is

$$\mathrm{E}(D) = \frac{1}{2}\left[\frac{\delta^2}{r} + \frac{\sigma^2 + Nr\mu(1+\mu) + (N-1)r\mu}{(1-N\mu)\mu}\right], \qquad (7)$$

where μ is the arrival rate to a node, σ^2 is the variance of the arrival process for a node, $r = \mathrm{E}(S)$, and $\delta^2 = \mathrm{Var}(S) = \mathrm{E}(S^2) - \mathrm{E}^2(S)$ [7]. In a symmetric system with i.i.d. Bernoulli traffic, we have that $\mu = \rho/N$ and $\sigma^2 = \rho/N - (\rho/N)^2$, where ρ is the total arrival rate to the system. We will show in Section IV that (7) provides a very good approximation for the average delay when PSMAC Algorithm 1 is used.

2) Throughput Performance: In polling systems, throughput is closely related to the notion of *stability* [10], [11]. A scheme is said to stabilize the system, or achieves 100% throughput, if it can guarantee bounded delay as long as the offered load is strictly less than 100% (i.e., $\rho + \epsilon = 1$, for $0 < \epsilon \ll 1$) [6], [7].

As discussed, p-Persistent CSMA achieves a maximum throughput which is strictly less than 100% (see (5)). On the other hand, prior work on polling systems has shown that both exhaustive service and gated service can serve any offered load less than 100% with bounded delay [6], [7]. Therefore, the throughput of PSMAC Algorithm 1 should be very close to 100% even when the RTS/CTS overhead is taken into account. We will demonstrate this point in Section IV.

B. PSMAC Algorithm 2

For wireless networks, it is generally crucial to conserve energy (e.g., for disposable sensor nodes). It has been shown that a node in the sleep state consumes far less energy than in the idle, transmit, or receive state. It is therefore desirable to schedule nodes to sleep whenever possible, be it a Wireless LAN [12], a wireless sensor network [2], [13] or an ad hoc network [14]. We can modify PSMAC Algorithm 1 to enable such sleep scheduling for energy savings.

In PSMAC Algorithm 2, each node maintains $N-1$ *virtual queues*, one for each of its neighbors. If there are one or more non-empty virtual queues, the node first selects one of them. The selection strategy can be round robin, uniform, or by following a priority order (e.g., longest-queue-first). The node then attempts to transmit RTS as in p-Persistent CSMA to contend for service. In the RTS, it specifies K, the number of frames backlogged in the selected virtual queue at this time, and the ID of Node B, the destination node corresponding to the selected virtual queue. If the RTS succeeds, Node B will return a CTS and *gated service will be used for this virtual queue* (i.e., K frames will be transmitted back-to-back to Node B). All other nodes which are not involved in this transmission, can be scheduled to sleep during this period and wake up when the K frame transmissions are over.[3]

We expect that the throughput of PSMAC Algorithm 2 is not as high as Algorithm 1, since it uses gated service only for one virtual queue instead of the global queue at a node. However, it has its advantage if energy conservation is a major consideration. Furthermore, our simulation results show that the PSMAC 2 throughput is very close to 100%. Under bursty traffic patterns, its delay performance is also found to be very close to that of PSMAC 1 (see Section IV).

C. PSMAC Algorithm 3

For the two PSMAC algorithms, Algorithm 1 is more efficient in bandwidth utilization, while Algorithm 2 is more efficient for energy conservation. Motivated by these observations, we further extend the algorithms to obtain both advantages.

In Algorithm 3, each node maintains $N-1$ virtual queues, and nodes compete for the channel by sending RTS as in Algorithm 2. When a sender successfully wins the channel, it first broadcasts an *announcement frame*. The announcement frame notifies its neighbors the lengths of all its non-empty virtual queues, as well as the order in which the virtual queues will be served. That is, each destination node will know how

[3]Note that we can set a threshold K_{th} and schedule a node to sleep only if the expect sleep period is longer than K_{th}, in order to avoid frequently switching between sleep and awake modes with very short periods.

many frames it will receive, as well as the staring and ending times for reception, after receiving the announcement frame. The sender then starts data transmission, clearing the virtual queues one at a time in the same order as specified in the announcement frame. All other nodes, except the corresponding destination of the virtual queue currently being served, can be scheduled to sleep and to wake up when it is its turn to receive the frames. If a node finds out that it is not one of the announced destinations, it can go to sleep and wake up when all the virtual queues at the source node are cleared.

This way, we obtain a similar service as in Algorithm 1, as well as the energy conservation capability as in Algorithm 2. There is only one additional frame (the announcement frame) as extra overhead for each burst of data transmission, as compared to Algorithm 1. Thus Algorithm 3 achieves approximately the same delay and throughput performance as Algorithm 1, and approximately the same energy savings as Algorithm 2.

D. Extension to Multi-Channel and Multi-hop Networks

So far we have considered a single hop ad hoc network, where all the nodes can hear each other and share a common wireless channel. It would be interesting to consider the cases of multi-hop wireless networks and multi-channel wireless networks where multiple orthogonal channels are used. Can we use the three algorithms in such environments?

Although the use of multiple channels offers great potential for higher throughput, it also brings about challenging scheduling problems, since now connectivity also depends on channel assignment, in addition to mobility/distance and channel dynamics [15], [16]. An effective solutions to the above problem is to use a common control channel along with multiple data channels [17], [18]. With PSMAC, we can have nodes compete in the control channel for gated service of their backlogged frames in the data channels. We expect similar performance gains over traditional limited-1 type services as in [17], [18].

Extension to multi-hop wireless networks is a more complicated issue, since each node now sees a different set of neighbors and the inherent hidden terminal and exposed terminal problems should be carefully addressed. Among the three schemes, it is relatively easier to adopt PSMAC Algorithm 2 for a multi-hop wireless network. Its operation is similar to that of IEEE 802.11: each time one receiver is reserved by a successful RTS/CTS dialog, but a gated service is used for the virtual queue corresponding to the target receiver. A question to ask is "how about its performance?" As will be demonstrated in Section IV, PSMAC Algorithm 2's performance on throughput, delay, and fairness approaches to those of Algorithms 1 and 3 as traffic gets bursty, in addition to its capability of energy savings. We conjecture that these trends will also hold true in the multi-hop wireless environment, where traffic may be even burstier than its wired counterparts due to large variations on wireless channel capacity [8].

PSMAC Algorithms 1 and 3 require reserve one or more receivers with a successful RTS/CTS dialog. In a multihop environment, this requires an additional three-way handshake. In addition, there is an interesting scheduling problem when only part of the target receivers are reserved, or a reserved receiver will be available for receiving only part of the frames in the corresponding virtual queue at the source node (since some of them may be involved in an ongoing transmission two hops away). We are working on these issues and will report our results in a sequel of this paper.

IV. Performance Evaluation

In this section, we present our performance study for the proposed algorithms. The delay, throughput, energy consumption and fairness performance of the three PSMAC algorithms are compared with that of p-Persistent CSMA under the following three types of traffic:

- *i.i.d. Bernoulli traffic*: a frame arrives in each time slot with a certain probability.
- *On-off bursty traffic*: frames are generated according to an on-off model with geometrically distributed on and off periods. The average on and off periods are 5 for the results reported in this section.
- *Long-range-dependent (LRD) traffic*: frames are generated according to an on-off model with (truncated) Pareto distributed on and off periods. This type of traffic is much more bursty than the other two. For results reported in this section the average on and off periods are 26.7 and the Hurst parameter is $H = 0.7$.

The *uniform* traffic pattern is used in most of the simulations, i.e., an arriving frame or burst is equal likely to be destined to each of the neighbors. In Section IV-C, we also use a non-uniform traffic pattern for fairness performance study, where a specific neighbor has a much higher load than all other nodes. We set $L = 10$ and $N = 20$ for most of the simulations and analysis, except that $N = 6$ for simulations with LRD traffic.

The three algorithms and p-Persistent CSMA are implemented using the C language. Each experiment is repeated 10 times with different random seeds and each point in the figures is the average of the 10 samples. We also compute 95% confidence intervals for the simulation results. Since they are generally very small, we only show the confidence intervals in the first few figures, but omit them in the following figures for clarity.

A. Delay and Throughput Performance

We first examine the delay of the proposed algorithms. In Figure 3, we plot the simulated average delay of Algorithm 1 and Algorithm 3, along with the analysis using (7), under the i.i.d. Bernoulli uniform traffic. We find that the analysis is quite accurate: the analytical curves matches the simulation curves, especially when the system is heavily loaded. The gap between the two curves, when the load is light, is due to the small discrepancy between our system and a random polling system. In a random polling system, after one station is served, the server may switch to an empty station, resulting in a service period of zero, followed by a new switch-over period. In our system, only non-empty nodes will be served, since an empty node will not send RTS to compete for the channel. Therefore, a switch-over time is *always* followed by a non-zero service

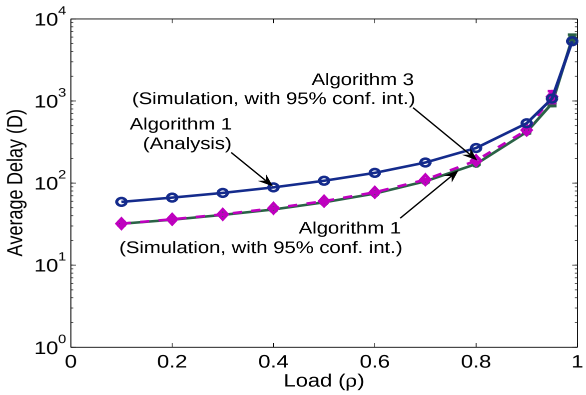


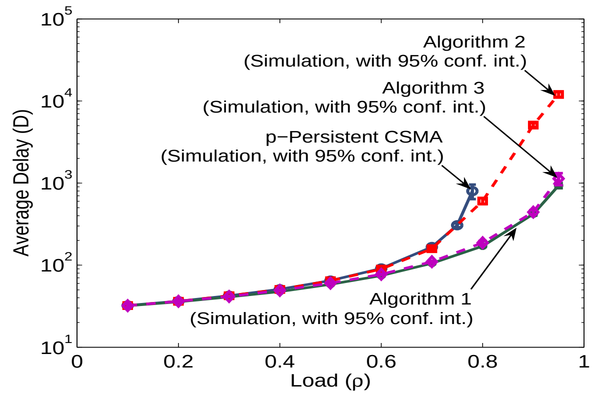
Fig. 3. Average delays under the uniform i.i.d. Bernoulli traffic.

period in our system. By using the less efficient random polling system model, the analysis is actually an *upper bound* of the delay when the load is light. When the load is heavy, nodes are less likely to be empty and the two systems behave more like each other. As a result, the analysis and simulation curves converge.

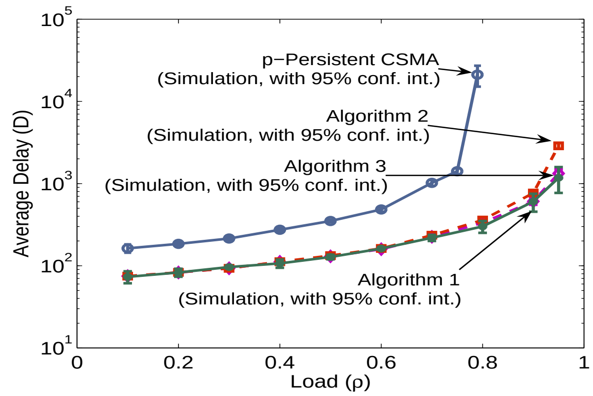
The average delay of the three proposed schemes are plotted in Figure 4 for different traffic models. It can be seen from Figure 4 that Algorithm 1 and Algorithm 3 have the same delay performance under the entire range of load examined and under the three traffic models. These are expected results, since from the descriptions in Section III, both algorithms use gated service for the global queue at a winning node, while the only difference is the order at which frames are served. That is, first-come-first-serve or one virtual queue at a time. We will show that Algorithm 3 achieves more energy savings than Algorithm 1 in Section IV-B.

We also plot the delays achieved by p-Persistent CSMA in Figures 4(a)-4(c), where the traffic is getting increasingly bursty. In all the figures, the PSMAC schemes outperform p-Persistent CSMA in delay performance, and the improvements are bigger when traffic is burstier. Specifically, the p-Persistent CSMA curve diverges when ρ is close to 79%, while the PSMAC delays are all bounded for all the traffic models and all the loads examined.

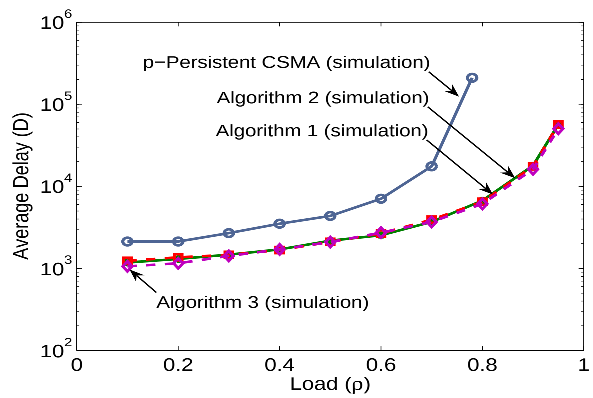
From Figure 4(a), we also find that all the four schemes have similar average delay in the low load region. This is because under a light load, the queues or virtual queues are less likely to build up. Although gated service is used in the three proposed algorithms, it usually serves queues with a single frame and thus reduces to limited-1 service as in p-Persistent CSMA. However, for the more interesting heavy load region, the Algorithms 1 and 3 delays are significantly smaller than that of p-Persistent CSMA. This is because under a heavy load, the queues are more likely to build up and the gated service will be more efficient than the limited-1 service. For Algorithm 2, although load is higher, the average rate to each virtual queue, $\mu = \rho/[N(N-1)]$, is still not big enough to build up large backlogs. Thus a gated service for a virtual queue is still more like a limited-1 service, and its delay curve remains close to that of p-Persistent CSMA. Nevertheless, it still achieves a higher throughput than p-Persistent CSMA.



(a) i.i.d. Bernoulli traffic



(b) On-off bursty traffic



(c) LRD traffic

Fig. 4. Average delays under various traffic models.

When traffic gets bursty, we find in Figures 4(b) and 4(c) that all the three PSMAC algorithms achieve significant delay improvement over p-Persistent CSMA. Under a bursty traffic, the backlogs of the global queues or virtual queues are more unevenly distributed. There is a high chance for the proposed algorithms to find a queue (or a virtual queue) with a large number of backlogged frames, and gated service will achieve a much better delay performance in these scenarios. A somewhat counter-intuitive observation is that, as traffic gets more bursty, the Algorithm 2 delay curve almost completely overlaps with the Algorithm 1 curve, although it only uses the gated service

TABLE I
ENERGY CONSUMPTION MODEL [19]

-	*Transmit*	*Receive*	*Idle*	*Sleep*
Power	1400mw	1000mw	830mw	130mw
Normalized	1.4	1.0	0.83	0.13

for a chosen virtual queue (see Figures 4(b) and 4(c)). This implies that under highly bursty traffic, the frames backlogged at a node are more concentrated in a small number of virtual queues (destined to a small subset of neighbors). In these cases, gated service for the global queue rather than a heavily loaded virtual queue does not make much difference.

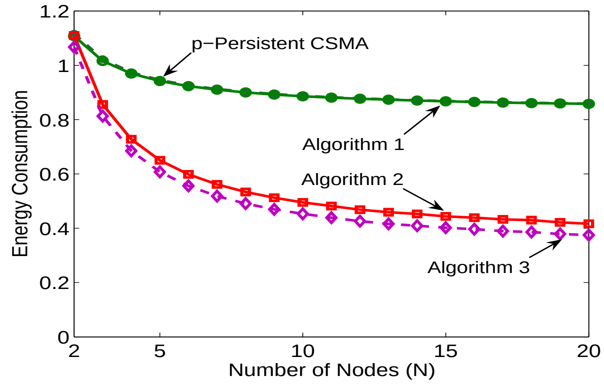
For a polling system, throughput is closely related to the notion of stability [10], [11]. From Figures 4, we find the p-Persistent CSMA curve diverges when ρ is close to 79% for all the traffic models. That is, when the load is close to 79%, the average delay becomes unbounded (i.e., goes to ∞). This verifies our analysis in Section II, since according to (5), $T^*(N = 20, L = 10) = 79\%$. For the proposed schemes, however, bounded delays are achieved even when the load is very close to 100%. Therefore, PSMAC can stablize the system [6] and achieve significant improvement in throughput over p-Persistent CSMA.

B. Energy Savings

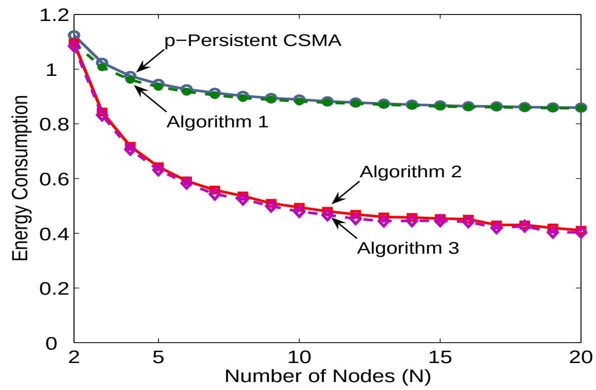
For wireless networks, it is very important to conserve battery power (e.g., for disposable sensor nodes). It has been shown in prior work that the most effective means of conserving energy is to schedule nodes to sleep whenever possible [13], [14]. As discussed, Algorithms 2 and 3 allow such sleep-scheduling due to the use of virtual queues. We examine the achievable energy savings in this section.

In the simulations, we use the power consumption model from [19], which is given in Table I. Note that we use a time slot as unit of time, and the normalized power (in units per time slot) is used in our simulations. In Figure 5, we plot the simulated average energy consumption, i.e., the average normalized energy consumed per node per time slot, for all the four schemes under the i.i.d. Bernoulli traffic, the on-off bursty traffic, and the LRD traffic. The load is $\rho = 0.7$ for the first two and 0.4 for the LRD traffic simulations. We vary the number of nodes N from 2 to 20.

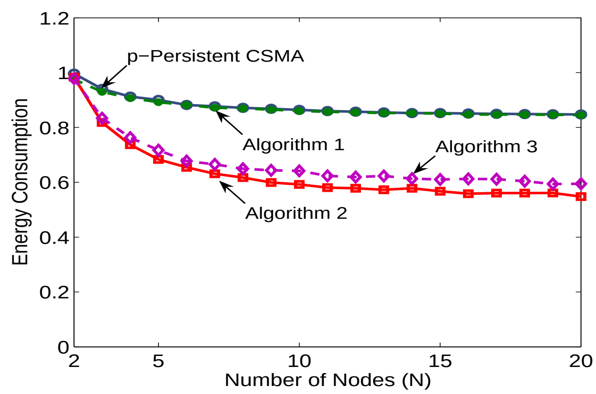
It can be seen that p-Persistent CSMA and Algorithm 1 consume almost the same amount of energy per node per time slot, while Algorithm 2 and 3 are much more energy efficient. For example, when $N = 20$, the average energy consumption of p-Persistent CSMA is 0.8590, while the average energy consumption of Algorithm 2 is 0.4166. The normalized reduction is $(0.8590 - 0.4166)/0.8590 = 51.5\%$. Note that when $N = 2$, all the schemes have similar energy consumptions. This is because there is no way to schedule nodes to sleep when $N = 2$; when one node is transmitting, the other node must be receiving.



(a) i.i.d. Bernoulli traffic ($\rho = 0.7$)



(b) On-off bursty traffic ($\rho = 0.7$)



(c) LRD traffic ($\rho = 0.4$)

Fig. 5. Energy consumption under various traffic models.

C. Fairness Performance

One general concern of using the gated or exhaustive service is fairness performance. Usually, compared to limited-1 service, these two service disciplines favor heavily loaded users. In this section, we examine the fairness performance using the fairness index defined as follows [20],

$$f(D_1, D_2, \cdots, D_N) = \frac{(\sum_{i=1}^{N} D_i)^2}{N \sum_{i=1}^{N} D_i^2}, \tag{8}$$

where D_i is the average delay of node i, $i \in [1, 2, \cdots, N]$. This fairness index generally varies from 0 to 1. When all nodes have the same average delay, we have that $f = 1$ and the system is 100% fair. As the disparity increases, fairness decreases for schemes which favor only a selected few nodes. For example, when the delay of one node is dominant (i.e., $D_1 \gg D_i, \forall i \neq 1$), the fairness index is $f \approx 1/N$ (and $\lim_{N\to\infty} f = 0$).

Under the uniform traffic pattern, our simulation results show that all the schemes have a similar fairness index. We omit these results for brevity. Now let's consider a *nonuniform* traffic pattern as follows. Recall that ρ is the arrival rate to the system and μ_i the arrival rate to Node i. The arrival rates to the nodes are determined as

$$\mu_i = \begin{cases} \frac{\rho}{2}, & i = 1 \\ \frac{\rho}{2(N-1)}, & 2 \leq i \leq N \end{cases} \quad \text{and} \quad \sum_{i=1}^{N} \mu_i = \rho. \qquad (9)$$

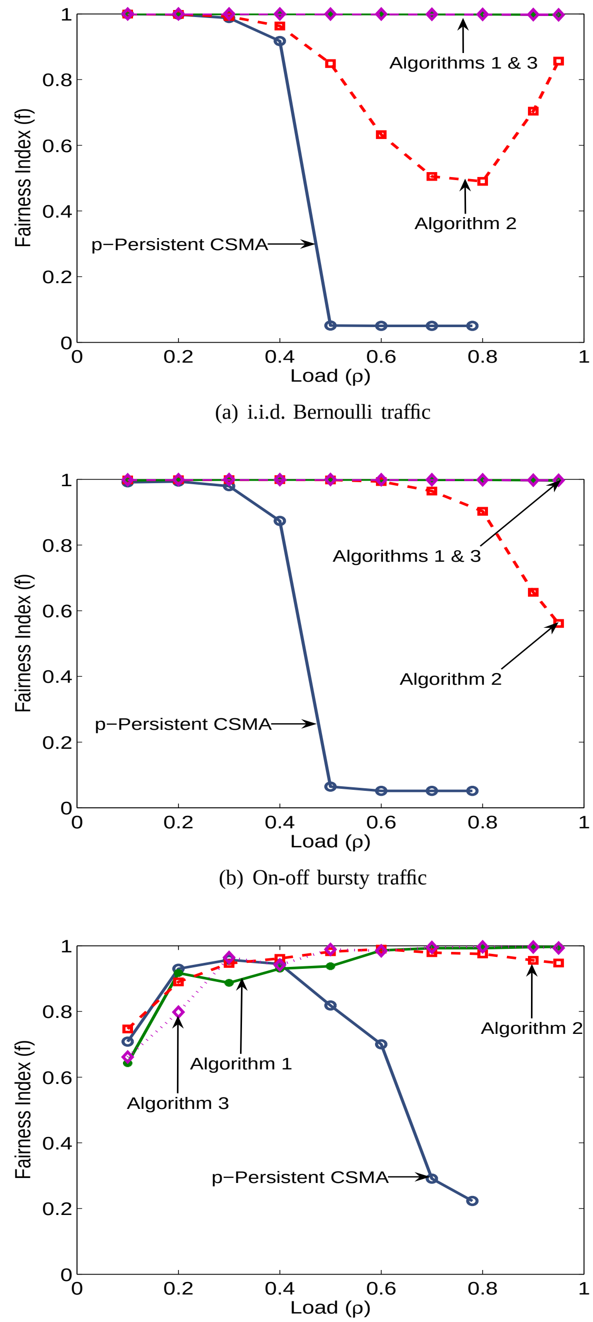
With this non-uniform traffic pattern, Node 1 is heavily loaded, while all the other nodes are lightly and equally loaded.

Figure 6 shows the fairness index achieved by the four schemes under the i.i.d. Bernoulli, the on-off bursty and the LRD traffic. It can be seen that when load ρ is low, the fairness indices of all the schemes are almost the same. For increased ρ, the p-Persistent CSMA fairness index drops quickly, while the fairness curves of Algorithm 1 and 3 remain at high values close to 1. Specifically, under i.i.d. Bernoulli traffic and the on-off bursty traffic, the p-Persistent CSMA fairness index drops to $f = 1/N = 5\%$, when ρ is beyond 50%. This indicates that one of the nodes, i.e., Node 1, has a large average delay that dominates the delays of all other nodes. This is illustrated in Figure 7 for the case of $\rho = 0.7$ under the on-off bursty traffic. When p-Persistent CSMA is used, the Node 1 delay is 18147.5, which is much larger than the delays of other nodes (ranging from 320 to 350). We also find that the average node delays achived by the three PSMAC algorithms are all lower than the corresponding p-Persistent CSMA node delay. Furthermore, the Node 1 delay is slightly lower than those of all the other nodes when Algorithms 1 and 3 are used, and slightly higher than those of all the other nodes when Algorithm 2 is used. The PSMAC fairness indices are all much larger than that of p-Persistent CSMA.

These are quite *counter-intuitive results*, since, contrary to the common belief, the use of gated (or exhaustive) service does not result in poor fairness performance. Rather, the three PSMAC schemes achieve much better fairness performance than p-Persistent CSMA. This is largely due to the high efficiency of the polling service-based schemes; *all* the queues are efficiently served and the delays of those lightly loaded nodes are only slightly increased (due to the heavily loaded node). Thus the benefit introduced by gated service to a heavily loaded node does not seriously increase the delay of other nodes. Fairness is not sacrificed for improved delay and throughput performance.

V. Related Work

Efficient MAC schemes have been the subject of intensive research for years. There have been a large number of MAC schemes proposed in the literature for wired and wireless networks, such as ALOHA, Slotted ALOHA, CSMA, and CSMA/CA. This research regained considerable interests recently, largely due to the dominant adoption of IEEE 802.11 family protocols [12], [21] and Bluetooth [22] for wireless LANs as well as multi-hop wireless networks [1], [2], [4], [5], [13]–[19], [23], [24]. To the best of our knowledge, none of the schemes use a gated or exhaustive service for a winning node. Using such polling services is analogous to the move from Stop-and-Wait flow control to Go-Back-N: by allowing a larger transmission window, a higher throughput can be achieved. In the proposed algorithms, by using the gated or exhaustive service, more frames are served for a winning node, thus the



(a) i.i.d. Bernoulli traffic

(b) On-off bursty traffic

(c) LRD traffic

Fig. 6. Fairness indices under various traffic models.

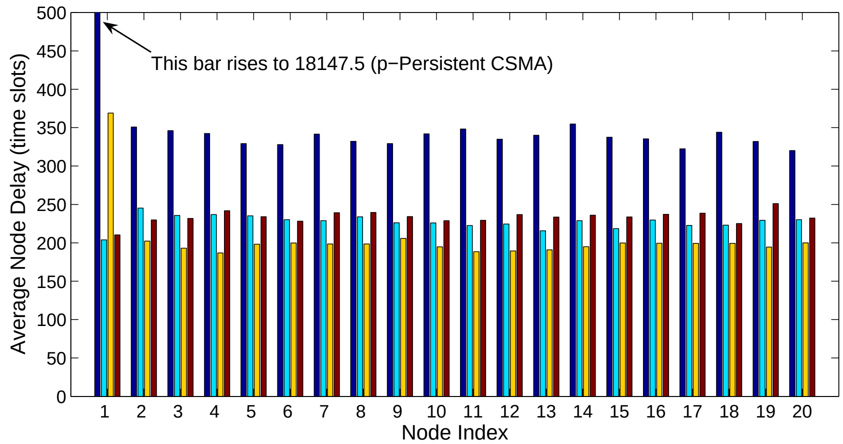


Fig. 7. The average delay for each node achieved by the four schemes under non-uniform on-off bursty traffic with $\rho = 0.7$. The bars in each group, from left to right, are for p-Persistent CSMA, Algorithm 1, Algorithm 2, and Algorithm 3, respectively.

channel contention overhead can be amortized over the frames and higher efficiency can be achieved.

The master-driven architecture of Bluetooth piconets provides an ideal setting for applying polling-based scheduling. In fact, polling is adopted in Bluetooth piconets for access control, although the actual scheduling policy has not been prescribed in the current standard [22], [23]. In addition, a polling mechanism has been incorporated in the recent IEEE 802.11e Hybrid Coordination Function (HCF) [21]. In the HCF Controlled Channel Access (HCCA) mode, the hybrid coordinator (HC) (co-located with the QoS Access Point (QAP)) controls the polling mechanism, to assign transmission opportunity (TXOP) to QoS enhanced stations (QSTA), which is a bounded time interval in which a QSTA is allowed to transmit one or more frames. Again, the scheduling policy is not specified. In both cases, a centralized controller is required to poll the secondary nodes according to some predefined policy, which is different from the random access and fully distributed approach taken in this research.

An analysis is presented in [1] on the throughput and delay performance bounds for the IEEE 802.11 protocols. The authors show that by simply increasing the data rate without reducing overhead, the enhanced performance, in terms of throughput and delay, is moderate even when the data rate goes to infinity. This interesting work provides a motivation for reducing control overhead in IEEE 802.11-like wireless networks.

In two recent papers [17], [18], the authors analyze the split channel MAC schemes that are based on the RTS/CTS dialogue and the pure-ALOHA or p-Persistent CSMA for contention resolution. The shared channel is split, either in time or frequency, into multiple channels: one is used for control and the rest for data. An interesting observation is that, under certain conditions, the maximum achievable throughput of the split-channel MAC schemes is lower than that of the corresponding single-channel MAC schemes. Note that the limited-1 service is used in [17], [18]. As discussed, our scheme can be adapted to the multi-channel case for improved performance.

VI. Conclusions

In this paper, we presented three polling service-based MAC schemes (termed PSMAC) for reducing control overhead and achieving performance gains. We presented analytical and simulation studies of a p-Persistent CSMA reservation-based scheme and the three proposed schemes, under various traffic models. The proposed PSMAC algorithms achieve significant gains on throughput, delay, and energy consumption over p-Persistent CSMA. In addition, we found PSMAC can effectively handle bursty traffic typically found in wireless networks. Our simulation results also show that due to the high efficiency of the proposed schemes, the performance gains can be achieved without hurting the fairness performance. We discussed strategies to adopt the proposed schemes for multi-channel or multi-hop wireless networks, which will be reported in a sequel to this paper.

Acknowledgments

This work has been supported in part by the National Science Foundation under Grants CNS-0520054, CNS-0435303, CNS-0435228 and WICAT, and by the New York State Office of Science, Technology and Academic Research (NYSTAR) through the Center for Advanced Technology in Telecommunications (CATT) at Polytechnic University.

References

[1] Y. Xiao and J. Rosdahl, “Throughput and delay limits of ieee 802.11,” *IEEE Commun. Letters*, vol. 6, no. 8, pp. 355–357, Aug. 2002.

[2] A. Woo and D. Culler, “A transmission control scheme for media access in sensor networks,” in *Proc. ACM MOBICOM’01*, Rome, Italy, July 2001, pp. 221–235.

[3] J. Li, C. Blake, D. Coute, H. Lee, and R. Morris, “Capacity of ad hoc wireless networks,” in *Proc. ACM MobiCom’01*, Rome, Italy, July 2001, pp. 61–69.

[4] R. Bruno, M. Conti, and E. Gregori, “Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless lans,” *IEEE Trans. Mobile Computing*, vol. 1, no. 1, pp. 10–31, Jan./Mar. 2002.

[5] F. Cali, M. Conti, and E. Gregori, “Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit,” *IEEE/ACM Trans. Networking*, vol. 8, no. 6, pp. 785–799, Dec. 2000.

[6] H. Takagi and L. Kleinrock, "A tutorial on the analysis of polling systems," Computer Science Department, UCLA Report No. 850005, Tech. Rep., Feb. 1985.

[7] L. Kleinrock and H. Levy, "The analysis of random polling systems," *Operations Research*, vol. 36, no. 5, pp. 716–732, Sept./Oct. 1988.

[8] I. Lee and A. Fapojuwo, "Estimating heavy-tails in long-range dependent wireless traffic," in *Proc. IEEE VTC 2005-Spring*, Stockholm, Sweden, June 2005, pp. 2132–2136.

[9] S. Ross, *Applied Probability Models with Optimization Applications*. New York, NY: Dover Publications, 1970.

[10] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, no. 12, pp. 1936–1949, Dec. 1992.

[11] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[12] IEEE, "Wireless LAN media access control (MAC) and physical layer (PHY) specifications," 1999, [Online]. Available: http://standards.ieee.org/getieee802.

[13] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated, adaptive sleeping for wireless sensor networks," *ACM/IEEE Trans. Networking*, vol. 12, no. 3, pp. 493–506, June 2004.

[14] R. Zheng and R. Kravet, "On-demand power management for ad hoc networks," *Elsevier Ad Hoc Networks J.*, vol. 3, no. 3, pp. 51–68, Jan. 2005.

[15] P. Kyasanur and N. Vaidya, "Capacity of multichannel wireless networks: Impact of channels, interfaces and interface switching delay," University of Illinois at Urbana-Champaign, Tech. Rep., Oct. 2006.

[16] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceive," in *Proc. ACM MOBIHOC'04*, Roppongi, Japan, May 2004, pp. 222–233.

[17] J. Deng, Y. Han, and Z. Haas, "Analyzing split channel medium access control schemes," *IEEE Trans. Wireless Commun.*, vol. 5, no. 5, pp. 967–971, May 2006.

[18] Y. Han, J. Deng, and Z. Haas, "Analyzing multi-channel medium access control schemes with ALOHA reservation," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 2143–2152, Aug. 2006.

[19] 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'01*, Rome, Italy, July 2001, pp. 85–96.

[20] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," Feb. 1999, ATM Forum Document Number: ATM_Forum/99-0045.

[21] IEEE, "Part 11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," July 2003, ANSI/IEEE Std 802.11e, Draft 5.0.

[22] Bluetooth Special Interest Group, "Specification of the Bluetooth system," Nov. 2003, version 1.2. [online]. Available: http://www.bluetooth.com.

[23] J. Misic and V. Misic, "Bridges of Bluetooth county: Topologies, scheduling, and performance," *IEEE J. Select. Areas Commun.*, vol. 21, no. 2, pp. 240–258, Feb. 2003.

[24] X. Qin and R. Berry, "Distributed approaches for exploiting multiuser diversity in wireless networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 392–413, Feb. 2006.