Learning Automata-Based Polling Protocols for Wireless LANs

P. Nicopolitidis, G. I. Papadimitriou, *Senior Member, IEEE*, and A. S. Pomportsis, *Member, IEEE*

***Abstract*—A learning automata-based polling (LEAP) protocol for wireless LANs, capable of operating efficiently under bursty traffic conditions, is introduced. We consider an infrastructure wireless LAN, where the access point (AP) is located at the center of a cell which comprises a number of mobile stations. According to the proposed protocol, the mobile station that is granted permission to transmit is selected by the AP by means of a learning automaton. The learning automaton takes into account the network feedback information in order to update the choice probability of each mobile station. It is proved that the learning algorithm asymptotically tends to assign to each station a portion of the bandwidth proportional to the station's needs. LEAP is compared to the randomly addressed polling and group randomly addressed polling protocols and is shown to exhibit superior performance under bursty traffic.**

***Index Terms*—Bursty traffic, dynamic bandwidth allocation, learning automata-based polling (LEAP), wireless LANs.**

I. Introduction

THERE are fundamental differences between wireless and wired LANs that pose difficulties in the design of medium-access control (MAC) protocols for wireless LANs (WLANs) [1]–[3]. WLANs, as the name suggests, utilize wireless transmission for information exchange. The wireless medium is characterized by bit-error rates (BER) having an order of magnitude even up to ten orders of magnitude of a LAN cable's BER. The primary reasons for the increased BER are atmospheric noise, physical obstructions found in the signal's path, multipath propagation, interference from other systems, and terminal mobility. Furthermore, in WLANs, errors occur in bursts, whereas in traditional wired systems, errors appear randomly. Finally, a fully connected topology between the nodes of a WLAN cannot be assumed. Rather, the logical topology of a WLAN tends to change as users move from one position to another. As a result, WLANs are characterized by unreliable links between nodes, resulting in bursts of errors and dynamically changing network topologies.

Modern WLAN MAC protocols should be able to efficiently handle the bursty traffic that is expected to be generated by WLAN applications (such as client/server and file transfer applications between WLAN nodes). In this paper, we propose learning automata-based polling (LEAP), a new polling protocol designed for bursty traffic WLANs. Learning automata are efficient structures that can provide adaptation to systems operating in changing and/or unknown environments [4], [5]. We consider an infrastructure WLAN where an access point (AP) is located at the center of a cell which contains a number of mobile stations. According to the proposed protocol, the mobile station that grants permission to transmit is selected by the AP by means of a learning automaton. The learning automaton takes into account the network feedback information in order to update the choice probability of each mobile station. The network feedback conveys information both on the traffic pattern of the network and the condition of the wireless link between the AP and the mobiles. It is proved that the learning algorithm asymptotically tends to assign to each station a portion of the bandwidth proportional to the station's needs.

This paper is organized as follows. Section II discusses work related to the subject of the paper. The LEAP protocol is presented in Section III, and an analysis of the asymptotic behavior of the system, which consists of the automaton and the network, is presented in Section IV. Simulation results comparing the performance of the proposed protocol and the performance of a family of proposed polling protocols for WLANs, randomly addressed polling (RAP) and group randomly addressed polling (GRAP), are presented in Section V. Those results reveal the performance superiority of LEAP under bursty traffic. This superiority of LEAP becomes even more important due to its simpler hardware implementation. Those, along with other concluding remarks, are discussed in Section VI.

II. Related Work

Polling is an appealing MAC option for a WLAN [7] since it offers centralized supervision of the network nodes. However, constant monitoring of all nodes is required, which is not feasible in the harsh fading environment of a WLAN. A good summary of wireless MAC protocols, including polling ones, can be found in [6]. Below, we summarize some representative polling protocols. Of these, the first two will be used later on in the paper as a metric for the performance achieved by LEAP.

An effort to alleviate the above-mentioned problem of constant monitoring of all nodes is made by the RAP protocol [7]–[10]. RAP is designed to work, not with all the nodes contained in a cell, but only with the active ones seeking uplink communication. The RAP protocol assumes an infrastructure cellular topology. Within each cell, multiple mobile nodes exist that, when active, compete for access to the wireless medium. For a network of N active mobile stations under the coverage of an AP, the stages of the protocol are as follows.

- *Contention invitation stage*: Whenever the AP is ready to collect packets from the mobile nodes, it transmits a

Paper approved by Y. Fang, the Editor for Wireless Networks of the IEEE Communications Society. Manuscript received June 4, 2002; revised September 10, 2002.

The authors are with the Department of Informatics, Aristotle University, 54006 Thessaloniki, Greece (e-mail: g.i.papadimitriou@ieee.org).

Digital Object Identifier 10.1109/TCOMM.2003.809788

READY message, which may be piggy-backed in a previous downlink transmission.

- *Contention stage*: Each active mobile node generates a random number R, ranging from 0 to $P-1$. All active nodes transmit their random numbers simultaneously to the AP using a form of orthogonal transmission, such as code-division multiple access (CDMA) or frequency-division multiple access (FDMA). The number transmitted by each station identifies this station during the current cycle and is known as its random address. To combat the medium's fading characteristics, a station may transmit their generated random address up to q times in a single contention stage. When an error-free transmission is assumed, $q=1$ suffices. Optionally, the contention stage may be repeated L times. Each time, each active station generates and transmits a (possibly different) random address, as described above.
- *Polling stage*: Suppose that at the lth stage $(1 \le l \le L)$, the AP received the largest number of distinct addresses and these are, in ascending order, $R_1, R_2, \ldots, R_n$. The AP polls the mobile nodes using those numbers. When the AP polls mobile nodes with R_k, nodes that transmitted R_k as their random address at the lth stage transmit packets to the AP. Obviously, if two or more nodes transmitted the same random address at the lth stage, a collision will occur. If $n=N$ however, no collision occurs.
- If the AP successfully receives a packet from a mobile node, it sends a positive acknowledgment (ACK). ACK packets are transmitted right before polling the next mobile node. If a mobile node receives an ACK, it assumes correct delivery of its packet, otherwise, it retries during the next polling cycle.

GRAP is a modification of RAP. It adopts the super-frame structure, consisting of $P+1$ frames, and divides active nodes to groups. At the beginning of each frame, only the AP is allowed to transmit. After the AP's transmissions, the polling procedure begins. However, GRAP does not allow all active nodes to compete in a single contention period; rather, all nodes that successfully transmitted during the previous polling cycles maintain their random addresses and form the groups from 0 to $P-1$. A mobile station joins group j if the random address for its previously successful transmission was j. Newly active stations form the Pth group. All mobile stations that have time-bounded packets can join any group for contention. After the formation of groups, the polling procedure, according to RAP, begins for all stations inside each group.

A complete description of RAP and GRAP is provided in [8] and [9]. Numerical results show that increasing values of L yield better throughput results, however, the performance gain with $L>2$ is very small. As a result, a value of two for L seems to be a good choice. Orthogonal signaling can be implemented using CDMA, transmission in adequate time slots, etc. However, use of large values of P lead to increased circuit complexity. As a result, values of P around five are suggested in [7].

A number of other polling protocols have appeared in the literature. One such protocol is proposed in [11]. In this approach, the AP polls all mobile nodes for transmission requests in a round-robin fashion. If a mobile node has a packet to transmit, it responds with a request message, otherwise with a KEEP_ALIVE one. The poll-request handshake ensures a good communication channel between the AP and the mobile nodes. The AP polls nodes for data according to the requests received and all stations must be polled within a time period T equal to the coherence time of the wireless channel.

A variation of [11] is the disposable token MAC protocol (DTMP) [12]. DTMP substitutes the poll-request–poll-data packet exchange sequence with a poll-data one. Thus, the need to poll all stations in a period less than the coherence time of the channel is eliminated. In DTMP, when the AP polls a mobile node, it also indicates if it has buffered packets for this node. If the AP has no packets for the mobile node and the node has no packets to send, the mobile node remains silent. However, if the AP has packets for the mobile node, then the latter sends a short message in order to invite the AP to send its packets. If the mobile node has packets, it sends them in response to the AP's poll.

III. LEAP Protocol

According to the LEAP protocol, the AP is equipped with a learning automaton [13] which contains the choice probability for each mobile station under its coordination. Before discussing the way choice probabilities are used, we present the operation of LEAP. Each polling cycle of LEAP consists of a sequence of packet exchanges between the AP, the mobile which is granted permission to transmit, and a destination station (if the selected mobile station has a packet to transmit). The protocol uses four control packets, POLL, NO_DATA, BUFF_DATA, and ACK, whose duration is t_{POLL}, $t_{\mathrm{NO\_DATA}}$, $t_{\mathrm{BUFF\_DATA}}$, and t_{ACK}, respectively. Assuming that the AP polls mobile station k at time position t which marks the beginning of polling cycle j, the propagation delay is $t_{\mathrm{PROP\_DELAY}}$, and a data packet transmission takes t_{DATA} time to complete, a number of events are possible. These are schematically depicted in the message exchange sketch shown in Fig. 1 and are summarized below.

1) The poll is received at station k at time $t + t_{\mathrm{POLL}} + t_{\mathrm{PROP\_DELAY}}$.
   (a) If station k does not have a buffered packet, it immediately responds to the AP with a NO_DATA packet. If the AP correctly receives the NO_DATA packet [Fig. 1(a)], it lowers the choice probability of station k and immediately proceeds to poll the next station. This poll is initiated at time $t + t_{\mathrm{POLL}} + 2 * t_{\mathrm{PROP\_DELAY}} + t_{\mathrm{NO\_DATA}}$. In case of no reception at the AP (Fig. 1(b), marked NO_DATA packet), the choice probability of station k is lowered and the next poll begins at time $t + t_{\mathrm{POLL}} + 4 * t_{\mathrm{PROP\_DELAY}} + t_{\mathrm{BUFF\_DATA}} + t_{\mathrm{DATA}} + t_{\mathrm{ACK}}$.
   (b) If station k has a buffered DATA packet, it responds to the AP with a BUFF_DATA packet, transmits the DATA packet to its destination, and waits for an ACK packet [Fig. 1(c)]. The AP monitors the wireless medium for a time interval equal to $t_{\mathrm{BUFF\_DATA}} + t_{\mathrm{DATA}} + t_{\mathrm{ACK}} + 3 * t_{\mathrm{PROP\_DELAY}}$. If it correctly receives one or more of the three

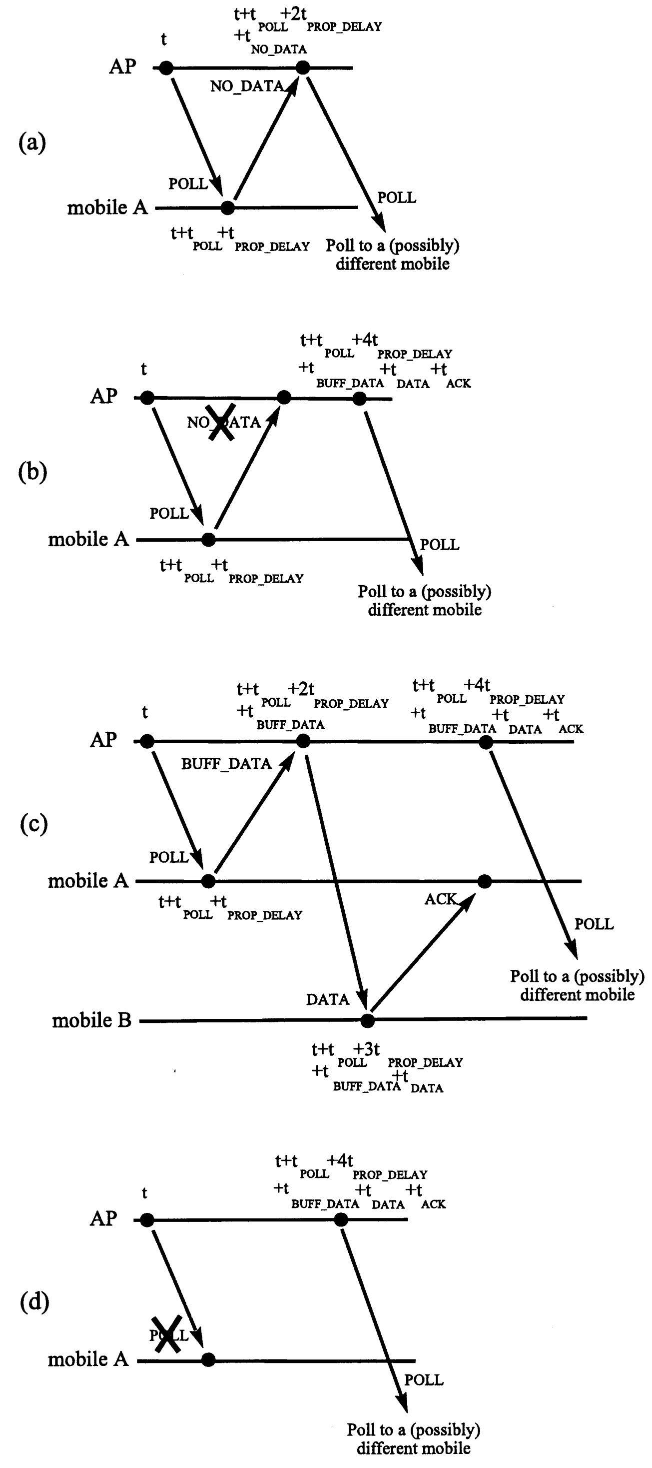


Fig. 1. Message-exchange sketch and timeline for the LEAP algorithm.

packets, it concludes that station k received the poll and has one or more buffered data packets. Thus, it raises station's k choice probability. On the other hand, if the AP does not receive reliable feedback, it concludes that it cannot communicate with k, lowers the choice probability of k, and proceeds with the next poll at time $t + t_{\rm POLL} + 4 * t_{\rm PROP\_DELAY} + t_{\rm BUFF\_DATA} + t_{\rm DATA} + t_{\rm ACK}$.

2) The poll is not received at station k (Fig. 1(d), marked $\rm POLL$ packet), k does not respond to the AP, and the latter proceeds to poll the next station at time $t + t_{\rm POLL} + 4 * t_{\rm PROP\_DELAY} + t_{\rm BUFF\_DATA} + t_{\rm DATA} + t_{\rm ACK}$.

From the above discussion, its is obvious that the learning algorithm takes into account both the bursty nature of the traffic and the bursty appearance of errors over the wireless medium. These types of information are used by the learning automaton at the AP in order to determine which mobile station will be polled (equivalently, granted permission to transmit). Upon conclusion of a polling cycle j, the AP examines the network feedback information in order to update the choice probability of the mobile station k that was polled at cycle j. If the feedback information indicates that k performed a data packet transmission in cycle j, the basic choice probability P_k for the next polling cycle $j+1$ increases to become $P_k(j+1) = P_k(j) + L(1 - P_k(j))$. This means that the AP correctly received one or more of the $\rm BUFF\_DATA$, DATA, and possibly $\rm ACK$, packets exchanged due to k's transmission. On the other hand, if either the AP concludes due to reception of station's k $\rm NO\_DATA$ packet that k did not perform a data packet transmission, or the AP failed to receive feedback about k's transmission state at cycle j (either to the erroneous reception, or to the reception of no packets at all), the choice probability P_k of k for the next polling cycle $j+1$ decreases to $P_k(j+1) = P_k(j) - L(P_k(j) - a)$.

It should be noted that the AP only waits for a shorter time period $(t_{\rm poll} + 2 * t_{\rm prop\_delay} + t_{\rm no\_data})$ only after a correct reception of a $\rm NO\_DATA$ packet. The AP does not need to distinguish between an erroneous reception of the $\rm BUFF\_DATA$ packet and an erroneous reception of a $\rm NO\_DATA$ packet. If either the $\rm NO\_DATA$ or the $\rm BUFF\_DATA$ packet is erroneously received or not received at all at the AP, the AP waits for a time period equal to $t_{\rm poll} + 4 * t_{\rm prop\_delay} + t_{\rm buff\_data} + t_{\rm data} + t_{\rm ack}$, which is enough for a polled mobile to send a data packet and receive the acknowledgment. Thus, the AP always waits for the above time period after an erroneous reception of either a $\rm BUFF\_DATA$ or a $\rm NO\_DATA$ packet. Thus, after an erroneous reception of either a $\rm BUFF\_DATA$ or a $\rm NO\_DATA$ packet, the next poll packet never hits a station's data packet or the corresponding ACK.

At each polling cycle j, the basic choice probabilities P_k for each mobile station k are normalized in the following way: $\Pi_k(j) = P_k(j) / \sum_{i=1}^{N} P_i(j)$. Clearly, $\sum_{k=1}^{N} \Pi_k(j) = 1$, where N is the number of mobile stations under the coverage of the AP. In the beginning of each polling cycle, the AP polls mobile stations according to the normalized probabilities $\Pi_k(j)$.

For all j, it holds that L, $a \in (0,1)$, and $P_k(j) \in (a, 1)$. The role of parameters L and a is described below and will be made more clear in the next section.

- L governs the speed of the automaton convergence. The selection procedure for a value of L reflects the classic speed versus accuracy problem. The lower the value of L, the more accurate the estimation made by the automaton, a fact, however, that comes at expense over convergence speed.
- The role of parameter a is to enhance the adaptivity of the protocol. This is because when the choice probability of a station approaches zero, then this station is not selected for

a long period of time. During this period, it is probable that the station transits from idle to busy state. The same holds for the status of the link between the mobile station and the AP, after a period of time, it is probable that the link changes state. However, since the mobile station does not grant permission to transmit, the automaton is not capable of "sensing" those transitions. Thus, the use of a nonzero value for parameter a prevents the choice probabilities of the stations from taking values in the neighborhood of zero and increases the adaptivity of the protocol. The network capacity is limited by $1/a$ by the algorithm. This is an argument in favor of a small value for a. An alternative could be the use of a very small value for a and implementation of at the AP of another mechanism for stations newly waking up to get the AP's attention, such as through contention slots. However, such an approach would raise the complexity of LEAP, which in its present form demands at the AP only a processor that implements the learning algorithm.

Since the offered traffic is of bursty nature, when the AP realizes that the selected station had a packet to transmit, it is probable that the selected station will also have packets to transmit in the near future. Thus, its choice probability is increased. On the other hand, if the selected station notifies that it does not have buffered packets, its choice probability is reduced, since it is likely to remain in this state in the near future.

In general, the background noise and interference at the AP will be the same, if not lower, than that at a mobile node. When the AP fails to receive feedback about the selected mobile's state, the mobile is probably experiencing a relatively high level of background noise. In other words, it is "hearing" the AP over a link with a high BER. Since in wireless communications errors appear in bursts, the link is likely to remain in this state for the near future. Thus, the choice probability of the selected station is lowered in order to reduce the chance of a futile poll.

IV. Asymptotic Analysis

The LEAP protocol updates the choice probabilities of mobile stations according to the network feedback information. In this section, we will prove that the choice probability of each mobile station converges to the probability that this station is ready to transmit. Thus, it has a nonempty queue and it is capable of communicating successfully with the AP. The following theorem (presented in [14]) is needed to carry out the asymptotic analysis.

Lemma: Let $x(n)_{n\geq 0}$ be a stationary Markov process dependent on a constant parameter $\theta \in [0,1]$. Each $x(n) \in I$, where I is a subset of the real line. Let $\delta x(n) = x(n+1) - x(n)$. The following are assumed to hold:

(1) I is compact;
(2) $E[\delta x(n)|x(n) = y] = \theta\, \omega(y) + O(\theta^2)$;
(3) $E[|\delta x(n)|^2 \ | \ x(n) = y] = \theta^2\, b(y) + o(\theta^2)$;
(4) $E[|\delta x(n)|^3 \ | \ x(n) = y] = O(\theta^3)$, where

$$\sup_{y\in I} \frac{O\left(\theta^k\right)}{\theta^k} < \infty \text{ for } k = 2,3 \text{ and } \sup_{y\in I} \frac{o(\theta^2)}{\theta^2} \to 0 \text{ as } \theta \to 0;$$

(5) $\omega(y)$ has a Lipschitz derivative in I;
(6) $b(y)$ is Lipschitz in I.

If assumptions (1)–(6) hold, $\omega(y)$ has a unique root y^* in I, and $d\omega/dy|_{y=y^*} < 0$, then:

(1) $var[x(n)|x(0) = x] = O(\theta)$ uniformly for all $x \in I$ and $n \geq 0$;
(2) for any $x \in I$ the differential equation $(dy(\tau)/d\tau) = \omega(y(n))$ has a unique solution $y(\tau) = y(\tau, x)$ with $y(0) = x$ and $E[x(n)|x(0) = x] = y(n\theta) + O(\theta)$ uniformly for all $x \in I$ and $n \geq 0$;
(3) $(x(n) - y(n\theta))/\sqrt{\theta}$ has a normal distribution with zero mean and finite variance as $\theta \to 0$ and $n\theta \to \infty$.

Theorem 1: Under the LEAP protocol, the choice probability of a mobile station k_i converges to the probability that station k_i is ready to transmit. If the learning algorithm (2) is used and d_i is the probability that station k_i is ready (for $i = 1, \ldots, N$), then for any station k_i

$$\lim_{n\to\infty, L\to 0, a\to 0} P_i(n) = d_i.$$

Proof: We use the *Lemma* to prove the current theorem. Here, we have to identify $x(n)$ (of the *Lemma*) with $P_i(n)$,θ (of the *Lemma*) with L, and I (of the *Lemma*) with $(a, 1)$. The probabilities d_i are assumed to be constant. Therefore, $P_i(n)$ is a stationary Markov process, because the probability of transition to a state depends only on the previous state and not on the time variable n. In practice, the probabilities d_i may change after a time interval, but here we want to study how the automaton reacts to a given set of probabilities d_i. We will prove that at any time instant, the automaton tends to satisfy the relation $P_i = d_i$. Thus, we have

$$\begin{aligned} &E\left[\delta P_i(n)|P_i(n) = P_i\right] \\ &\quad = \frac{P_i}{\sum_{k=1}^N P_k}\left(d_i L(1-P_i) - (1-d_i)L(P_i - a)\right) \\ &\quad = L\frac{P_i}{\sum_{k=1}^N P_k}\left(-P_i + d_i + a(1-d_i)\right) \\ &\quad = L\omega(P_i) \end{aligned} \tag{1}$$

$$\begin{aligned} &E\left[|\delta P_i(n)|^2\,|P_i(n) = P_i\right] \\ &\quad = L^2\frac{P_i}{\sum_{k=1}^N P_k}\left(d_i(1-P_i)^2 + (1-d_i)(P_i - a)^2\right) \\ &\quad = L^2 b\left(P_i\right) \end{aligned} \tag{2}$$

$$\begin{aligned} &E\left[|\delta P_i(n)|^3\,|P_i(n) = P_i\right] \\ &\quad = L^3\frac{P_i}{\sum_{k=1}^N P_k}\left(d_i(1-P_i)^3 + (1-d_i)(P_i - a)^3\right) \\ &\quad = O(L^3). \end{aligned} \tag{3}$$

The functions $\omega(P_i)$ and $b(P_i)$ are defined as follows:

$$\omega(P_i) = \frac{P_i}{\sum_{k=1}^N P_k}\left(-P_i + d_i + a(1-d_i)\right) \tag{4}$$

$$b(P_i) = \frac{P_i}{\sum_{k=1}^N P_k}\left(d_i(1-P_i)^2 + (1-d_i)\left(P_i - a\right)^2\right). \tag{5}$$

It is immediately seen that assumptions (1)–(6) are satisfied. It can also be proved that $b(P_i)$ and $\omega'(P_i)$ are Lipschitz in (a,1) by showing that their first derivatives ($b'(P_i)$ and $\omega''(P_i)$, correspondingly) are bounded [15] for $P_i \in (a, 1)$.

It remains to show that $\omega(P_i)$ has a unique root P_i^r near the point $P_i^* = d_i$ and that $d\omega(P_i)/dP_i|_{P_i=P_i^r} < 0$. It is immediately seen that $\omega(P_i)$ has a unique root at the point $P_i^r = d_i + a(1-d_i)$. Since a can be arbitrarily small, it follows that P_i^r is in the neighborhood of the point $P_i^* = d_i$. The derivative of $\omega(P_i)$ at this point is

$$\frac{d\omega(P_i)}{dP_i}|_{P_i} = P_i^r = \frac{d\left(\frac{P_i}{\sum_{k=1}^{N} P_k}(-P_i+d_i+a(1-d_i))\right)}{dP_i}|_{P_i=P_i^r}$$
$$= \frac{1}{1+\frac{\sum_{k=1,k\neq i}^{N} P_k}{P_i^r}} < 0. \quad (6)$$

It has been shown that $\omega(P_i)$ has a unique root P_i^r in the neighborhood of the point $P_i^* = d_i$ and that the derivative of $\omega(P_i)$ at this point is negative.

If we set $P_i(\tau) = P_i^r$, the differential equation $(dP_i(\tau))/(d\tau) = \omega(P_i(\tau))$ is satisfied $(0 = 0)$. Thus, $P_i(\tau) = P_i^r$ is a solution of the above differential equation. From *Theorem 1*, it is derived that this solution is unique, thus, all the solutions starting in $(a, 1)$ of the differential equation $(dP_i(\tau))/(d\tau) = \omega(P_i(\tau))$ converge to the point $P_i(\tau) = P_i^r \simeq P_i^* = d_i$. According to *Theorem 1*, we have

$$\lim_{n\to\infty, a\to 0} E\left[P_i(n)\right] = P_i^* + O(L)$$
$$var\left[P_i(n)\right] = O(L) \text{ for all } n.$$

Consequently,

$$\lim_{n\to\infty, L\to 0, a\to 0} P_i(n) = d_i \quad \text{Q.E.D.} \quad (7)$$

According to *Theorem 1*, for any two mobile stations k_i and k_j (with $d_j \neq 0$), the LEAP protocol asymptotically tends to satisfy the relation

$$\frac{P_i}{P_j} = \frac{d_i}{d_j}. \quad (8)$$

This relation also holds for the normalized choice probabilities Π_i and Π_j

$$\frac{\Pi_i}{\Pi_j} = \frac{\frac{P_i}{\sum_{k=1}^{N} P_k}}{\frac{P_j}{\sum_{k=1}^{N} P_k}} = \frac{P_i}{P_j} = \frac{d_i}{d_j}. \quad (9)$$

In order to obtain a better understanding of the claim of (8), we performed a simulation study for a LEAP WLAN of 10 mobile stations, from which only stations 1 and 2 are active, with $d_1 = 0.8$ and $d_2 = 0.4$. The result of this experiment, which can be seen in Fig. 2, shows that the claim of (8) indeed stands. The automaton estimates of the basic choice probabilities P_1 and P_2 converge to $d_1 = 0.8$ and $d_2 = 0.4$, respectively. The same stands for the normalize choice probabilities Π_1 and Π_2, which converge to 2/3 and 1/3, respectively. Thus, the claim of (9) is also verified, since $(P_1/P_2) = (d_1/d_2) = (\Pi_1/\Pi_2) = 2$.

Fig. 3 shows the effect of the choice for parameters a and L on the convergence of the automaton. In this experiment, we

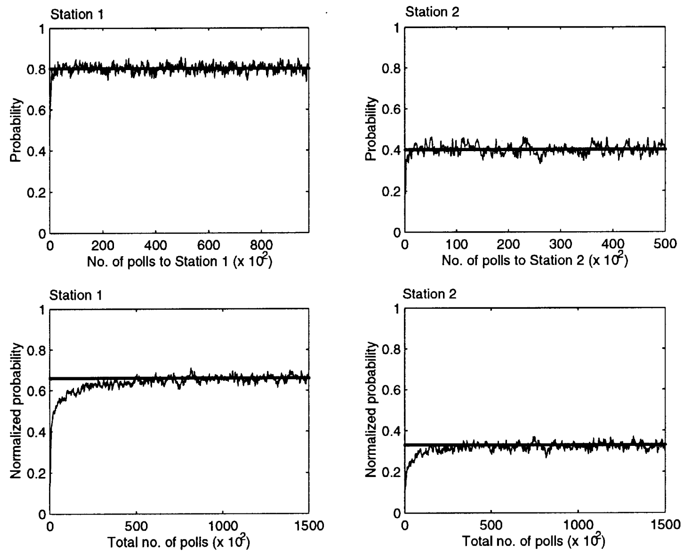


Fig. 2. Convergence of basic and normalized choice probabilities for stations 1 and 2.

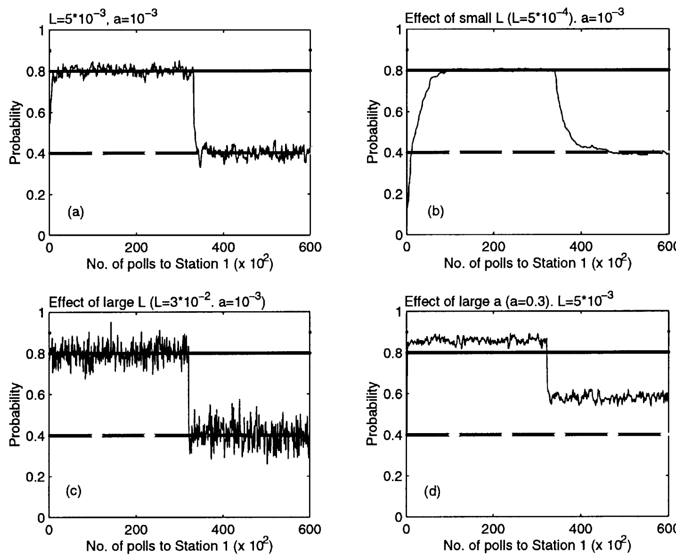


Fig. 3. Effects on convergence of basic choice probabilities by variation of parameters L and a.

assume a station i with $d_i = 0.8$. However, d_i is not constant during the entire duration of the simulation; rather, it changes from its initial value (solid line, $d_i = 0.8$) to a new one (dashed line, $d_i = 0.4$) after about 30 000 polls to this station. The conclusions that can be drawn from this figure are as follows.

- The choice for a value of L reflects the classic speed versus accuracy problem. As can be seen from Fig. 3(b), where $L = 5 * 10^{-4}$, small values of L provide a higher convergence accuracy. This, however, happens at the expense of convergence speed, as can be seen from the slow convergence, both at the beginning of the simulation and at the point where d_i changes [Fig. 3(b)]. On the other hand, increasing the value of L leads to a faster convergence, as can be seen from Fig. 3(c), where $L = 3 * 10^{-2}$. However, it can be seen that this increase in convergence speed comes at the expense of convergence accuracy.

- It was mentioned above that the basic choice probability P_i of a station i converges to $d_i + a(1 - d_i)$. Thus, convergence accuracy also depends on the value of a, with smaller values of a giving better convergence of P_i to d_i. Increased values of a will obviously make P_i converge to a point higher than d_i. Fig. 3(d) supports this fact.

When the environment is slowly switching or when the environmental responses have a high variance, a and L must be very close to 0 in order to guarantee a high accuracy. On the other hand, in a rapidly switching environment or when the variance of the environmental responses is low, higher values of a and L can be used, in order to increase the adaptivity of the protocol. Thus, when the burst length is high or the queue length is low or channel errors occur in large bursts, then small values of a and L must be selected. On the other hand, when the burst length is low or when the queue length is high or channel errors occur in small bursts, then a and L can be much higher.

V. Performance Evaluation

A. Simulation Environment

Using simulation, we compared the proposed protocol against RAP and GRAP. The bursty traffic was modeled in the following way. We define "time slot" as the time duration required for a data packet to be transmitted over the wireless link. Each source node can be in one of two states, S_0 and S_1. When a source node is in state S_0, then it has no packet arrivals. When a source node is in state S_1, then, at each time slot, it has a packet arrival with probability Z. Given a station is in state S_0 at time slot t, the probability that this station will transit to state S_1 at the next time slot is P_{01}. The transition probability from state S_1 to state S_0 is P_{10}. It can be shown that, when the load offered to the network is R packets/slot and the mean burst length is B slots, then the transition probabilities are $P_{01} = (R)/(B(NZ - R))$ and $P_{10} = (1/B)$. Each station uses a buffer to store the arriving packets. The buffer length is assumed to be equal to Q packets. Any packets arriving to find the buffer full are dropped.

In our simulation model, the condition of any wireless link was modeled using a finite-state machine with three states [16], [17]. The model comprises three states.

- Stage G denotes that the wireless link is in a relatively "clean" condition and is characterized by a small BER, which is given by the parameter G_BER.
- Stage B denotes that the wireless link is in a condition characterized by increased BER, which is given by the parameter B_BER.
- Stage U denotes that the pair of communicating stations is out of range of one another.

We assume that the background noise is the same for all stations, and thus, the principle of reciprocity stands for the condition of any wireless link. Therefore, for any two stations A and B, the BER of the link from A to B and the BER of the link from B to A are the same. The time spent by a link in states G, B, and H are exponentially distributed, but with different average values, given by the parameters TG, TB, and TH, respectively. The status of a link probabilistically changes between the three states. When a link is in state G and its status is about to change, the link transits either to stage U, with probability given by the parameter P_h, or to stage B, with transition probability $1 - P_h$. When a link is in state B and its status is about to change, the link transits either to stage U, with probability given by the parameter P_h, or to stage G, with transition probability $1 - P_h$. Finally, when a link spent its time in state U, it transits either to state G or B, with the same probability (0.5). It can be easily seen that by setting the parameter P_h to zero, a fully connected network topology can be assumed, whereas for values of P_h greater than zero, the effect of the well-known "hidden terminal" problem on protocol performance can be studied. For example, for $P_h = 0.1$, there is a 10% probability that two stations A and B are out of range of one another. Thus, for a third station C in range both of A and B, A and B are hidden nodes for transmissions from B to C and A to C, respectively. By changing the values for the various parameter of the above-described model, the protocols can be simulated for a variety of physical environments.

In the process of delivering our simulation results, we made the following assumptions.

1) If the AP was to send data packets to the mobile nodes in the same way for all three protocols, this would not make a difference as it would pose the same overhead for every protocol. Thus, in order to simplify the simulation model, we limit the role of the AP to be only the means of executing the polling algorithms. Therefore, no data packets are exchanged between the AP and the mobiles, and upon being polled, a mobile station can initiate a data packet transmission only with another mobile as its destination.
2) We did not include the effect of adding a physical layer preamble in our simulations. This turns out to be a conservative choice (albeit of a negligible impact) for LEAP, as it would slightly increase its superiority over RAP and GRAP. As will be explained later, this is due to the fact that RAP and GRAP require more control overhead per delivered data packet than LEAP does.
3) No error correction is used and we did not account for the possibility of packet capturing for RAP and GRAP. Whenever two packets collide, they are assumed lost. The capturing effect does not affect LEAP, since it is collision free (at each polling cycle only one station can be polled).

We employed the broadly used throughput versus offered load and delay versus throughput performance metrics in order to compare the three protocols. We simulated the protocols for two different sets of the following parameters. The number of mobile stations N, the buffer size Q, the mean burst length B, the packet arrival probability of each active mobile station Z, and the parameters B_BER and P_h. The simulation parameters are summarized in Table I.

The variable R_LIM sets the maximum number of transmission attempts per packet. If the number of retransmissions of a packet exceeds this value (either due to collisions or channel errors), the packet is dropped. At the MAC layer, the size of all control packets for the three protocols is set to 160 bits, the data packet size is set to 6400 bits, and the overhead Ovh for the orthogonal transmission of the random addresses in RAP and GRAP is set to five times the size of the poll packet, as in [8].

TABLE I
SIMULATION PARAMETERS

G_BER	0
TG	3 sec
TB	1 sec
TH	0.5 sec
L_{RAP}	2
P	5
q	1
N	10
Q	50 data packets
B	10 data packets
Z	1
R_LIM	6
L_{LEAP}	0.1
α_{LEAP}	0.03
Medium rate	1 Mbps
t_{PROP_DELAY}	0.0005 msec
Data packet size	6400 bits
Control packet size	160 bits
Ovh	5 x poll duration
B_BER in N_1	10^{-6}
B_BER in N_2	10^{-4}
P_h in N_1	0.0
P_h in N_2	0.1

The wireless medium bit rate was set to 1 Mb/s. The propagation delay between any two stations was set to 0.0005 ms, corresponding to interstation distances of 150 m.

The choice for values of TG, TB, and TH being 3, 1, and 0.5 s, respectively, provides a fading environment which can be effectively handled by LEAP. The reason for this fact can be understood by observing the performance superiority of LEAP as it appears in the figures and will be discussed later on. Specifically, since LEAP is able to adapt to the changing pattern of the traffic, it will also be able to do the same for the changing condition of the wireless links. This is because the choice for a mean burst length B of 10 data packets, a data packet size of 6400 bits, and a 1 Mb/s transmission speed produces average changes in the traffic pattern every 64 ms. Since LEAP is able to adapt to this changing environment, it will also be able to adapt to the wireless environment which, based on the above values of TG, TB, and TH, changes significantly more slowly ($TG = 3$ s, $TB = 1$ s, $TH = 0.5$ s).

Finally, as far as the parameters of LEAP, L, and a are concerned, we set them to 0.1 and 0.03, respectively, since the use of $B = 10$ provides a relatively fast changing environment, and, as mentioned in Section IV, in such environments, convergence speed is of more importance than convergence accuracy. In networks with $B \gg 10$, L and a can be much lower as the traffic environment will change more slowly.

The two network configurations in which the three protocols were simulated are the following.

1) Network N_1: $N = 10$, $Q = 50$, $B = 10$, $Z = 1.0$, $B\_\mathrm{BER} = 10^{-6}$, $P_h = 0.0$.
2) Network N_2: $N = 10$, $Q = 50$, $B = 10$, $Z = 1.0$, $B\_\mathrm{BER} = 10^{-4}$, $P_h = 0.1$.

Finally, we mention that in order to validate the LEAP simulator, we set B_BER, P_h, $t_{\mathrm{PROP\_DELAY}}$, and the size of the control packets to zero, and slightly altered the code so that the time elapsed after a poll to a station that does not have a buffered packet equals a time slot. Thus, the resulting simulator simulates the protocol proposed in [18]. We run the simulator for the same parameter values that give the results of [18], and the results we obtained coincided with those of [18]. Moreover, the RAP and GRAP simulators occurred with small programming changes from that of LEAP.

B. Simulation Results

Figs. 4–10 display a number of simulation results that reveal the performance superiority of LEAP over RAP and GRAP. For each point of the curves, the simulation was carried out until 400 000 packets were correctly received. Therefore, the confidence level of the simulation results is high. For example, the throughput of LEAP for an offered load of 1 packet/slot has the following 95% confidence intervals: 0.9135$\pm$ 0.0011 for Network N_1 and 0.6745 $\pm$ 0.0022 for Network N_2.

Following are the contributions of the figures.

- For Networks N_1 and N_2, Figs. 4 and 5, respectively, display the throughput versus offered load and delay versus throughput characteristics. The throughput of LEAP outperforms that of RAP and GRAP for medium and high loads, since LEAP is collision free and the per-DATA packet transmission overhead of the protocol is less. LEAP requires an overhead of three control packets per DATA packet (POLL, BUFF_DATA, ACK). RAP and GRAP, on the other hand, can transmit, at most, five DATA packets per polling cycle for $P = 5$, assuming no collisions occur, with an overhead of sixteen control packets, for $L = 1$, (READY, orthogonal transmission of random addresses which is equal to five times the

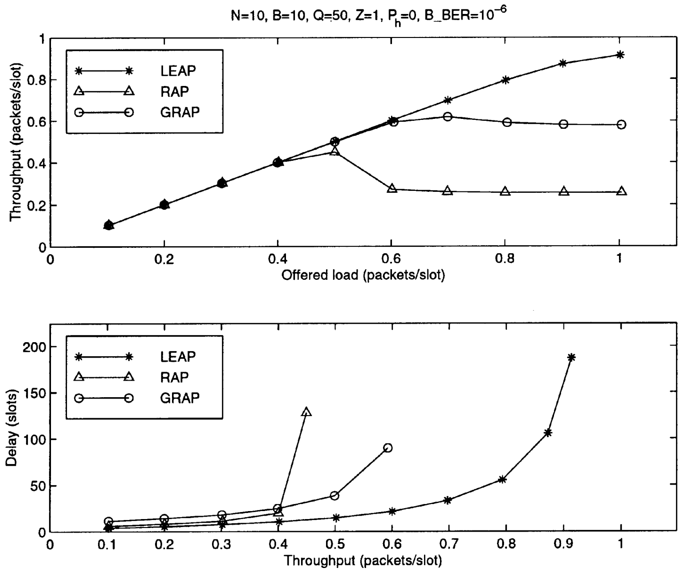


Fig. 4. Throughput versus offered load, and delay versus throughput characteristics of LEAP, RAP, GRAP, for Network N_1. The delay versus throughput characteristics of the protocol are plotted for packet loss rates lower than 10%.

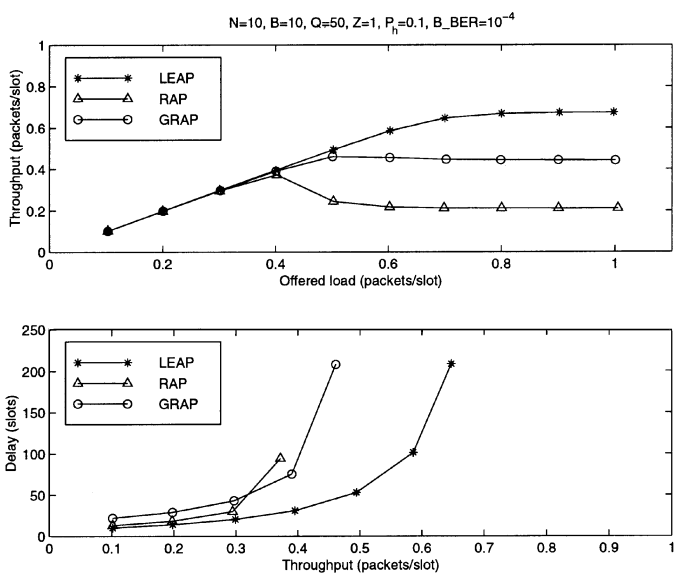


Fig. 5. Throughput versus offered load and delay versus throughput characteristics of LEAP, RAP, GRAP, for Network N_2. The delay versus throughput characteristics of the protocol are plotted for packet loss rates lower than 10%.

duration of a control packet, five POLL packets, five ACK packets) yielding an overhead of 3.2 control packets per DATA packet. However, this scenario rarely occurs in practice, due to the increased number of occurring collisions and the resulting instability of RAP for medium and high loads, where the number of active stations per polling cycle approaches the number of random addresses P. GRAP is not subject to the increased performance degradation of RAP, since the use of the superframe reduces collisions, as not all stations are allowed to compete at the same polling cycle. The relative behavior

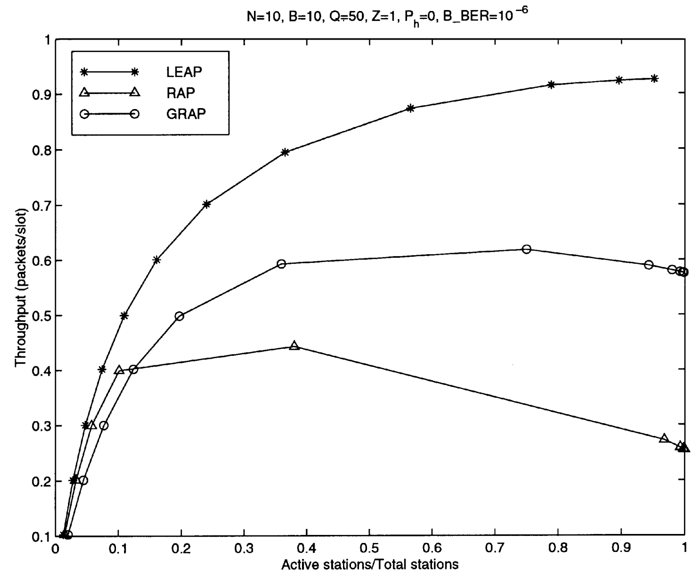


Fig. 6. Throughput LEAP, RAP, GRAP versus ratio of number of active stations to total number of stations for Network N_1.

of the protocols in Network N_2 (Fig. 5) is similar to that of Network N_1 (Fig. 4), with the difference that their performance degrades due to the increased unreliability of the wireless channel.

- Similar to the observation for the throughput versus offered load characteristics, we observe that the delay versus throughput characteristics of LEAP are superior to those of RAP and GRAP. The delay versus throughput characteristics plotted in Figs. 4 and 5 correspond to packet loss rates lower or equal to 10%. Our simulation results have shown that the superiority of LEAP also holds for higher packet loss rates.
- Fig. 6 displays the throughput of LEAP, RAP, and GRAP for Network N_1 for various values of the ratio r_1 of active stations to total number of stations. We note here that by the term "active station," we define a station that has a packet to transmit. It can be seen that the superiority of LEAP increases for increasing values of r_1. This is due to the fact that while LEAP is collision free, and thus, not affected by the increased number of active stations N_{active}, GRAP, and especially, RAP experience a high number of collisions for large values of r_1 where the number of active stations is bigger than the five available random addresses. Especially for $r_1 = 1$, $N_{\text{active}} = N = 10$, which is significantly higher than $P = 5$. It can be realized that the superiority of LEAP over RAP and GRAP for high values of r_1 will be even higher for networks with $N > 10$.
- Fig. 7 displays the throughput of LEAP, RAP, and GRAP for various values of the ratio r_2 of slot time to propagation delay. These results have been obtained by varying the size of a data packet from 500 to 5000 bits. For all values of r_2, the offered load to the network was 1 packet/slot and the rest of the simulation parameters are the same as those of Fig. 4 for Network N_1. As can be observed from the figure, LEAP outperforms both RAP and GRAP for the various values of r_2.

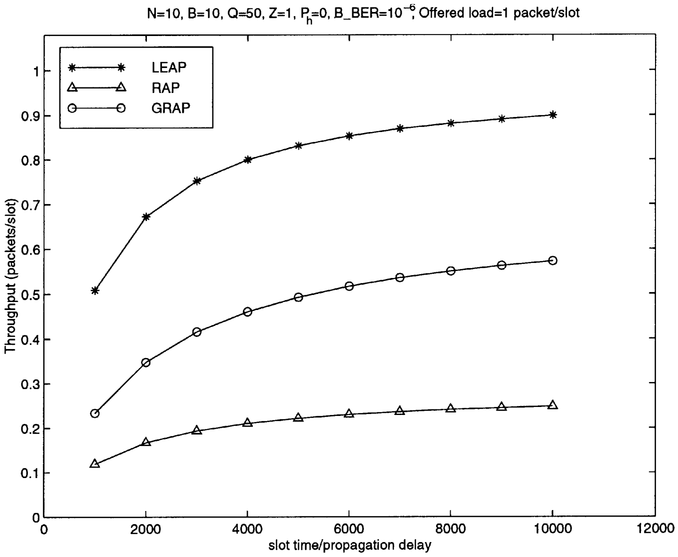


Fig. 7. Throughput LEAP, RAP, GRAP versus ratio of slot time to propagation delay for an offered load of 1 packet/slot.

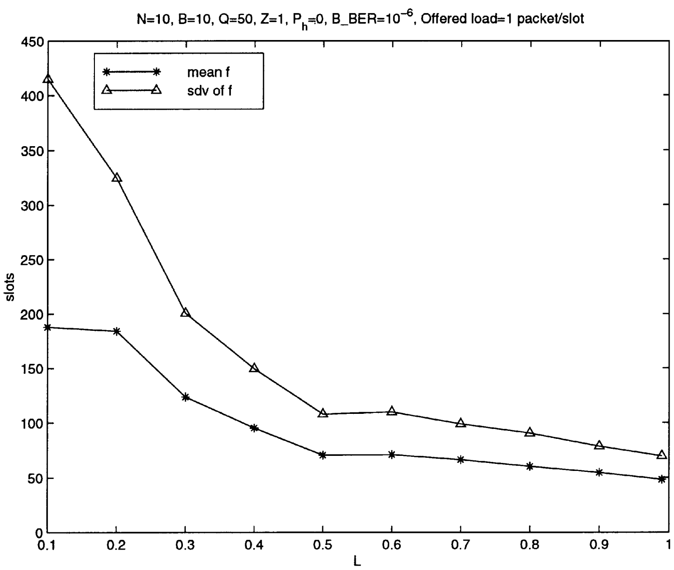


Fig. 8. Mean and standard deviation for the number of slots that have elapsed until the first transmission of a station that exits power-saving mode for various values of L.

- Fig. 8 displays the transient behavior of LEAP. In this experiment, we set 30% of the client population to have the ability to enter power-saving mode. The time spent by stations in this mode is given by an exponential distribution with mean 1 s. In our simulations, under the mentioned parameter values, we observed that a mean power-save mode time of 1 s is enough to lower the transmit probability of a node to the minimum value. Thus, we selected a mean power-save mode time of 1 s. While in this mode, stations do not respond to AP polls, thus, their choice probabilities remain very low. For various values of L, Fig. 8 plots the mean and standard deviation for the number of slots f that have elapsed until the first transmission of a station that exits power-saving mode. It can be seen from the figure that increased values of L provide lower waiting times,

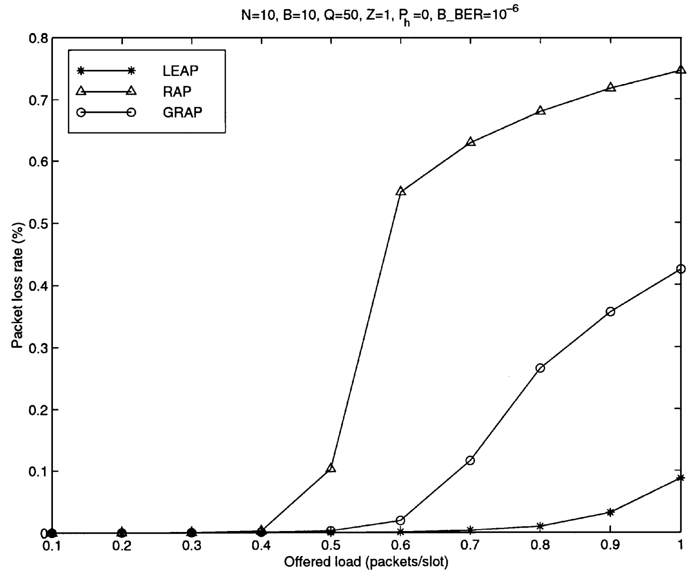


Fig. 9. Percentage of packet loss rate for LEAP, RAP, GRAP versus offered load for Network N_1.

which is to be expected, since, as mentioned in Section III, an increased L provides faster reaction of the automaton to environmental changes. However, it has also been mentioned that these higher reaction speeds come at an expense over accuracy. Thus, the result that can be drawn from this figure is that by increasing L, the performance for power-saving stations can be increased, at an expense, however, of convergence accuracy of the automaton and thus, protocol fairness.

- Fig. 9 displays the percentage of packets that were dropped for LEAP, RAP, and GRAP for Network N_1 either due to the finite capacity of station queues, or to the number of transmission attempts per data packet exceeding R_LIM. This percentage is calculated as the ratio of lost packets to the total number of packet arrivals. It can be seen that the number of dropped data packets for RAP and GRAP rise significantly at medium and high offered loads. This is mainly due to increased contention, which causes the number of transmission attempts per data packet to exceed R_LIM. This takes place as the number of active stations approaches and exceeds the five available random addresses. The percentage of data packets that are dropped by LEAP, however, remain under 10% even at high loads, and is due to the finite capacity of the stations' queues. The lower loss rate for LEAP implies that the protocol adapts to the changing state of stations more quickly than RAP and GRAP, as the lost of fewer data packets means faster adaptation to the changing environment.
- In order to assess the effect of data packet retransmissions, we plot in the top of Fig. 10 the percentage of data packets lost due to the number of transmissions attempts per data packet exceeding R_LIM (this percentage is again calculated as the ratio of these lost packets to the total number of packet arrivals), and the additional pressure caused on the request queues by data packet retransmissions. In order to

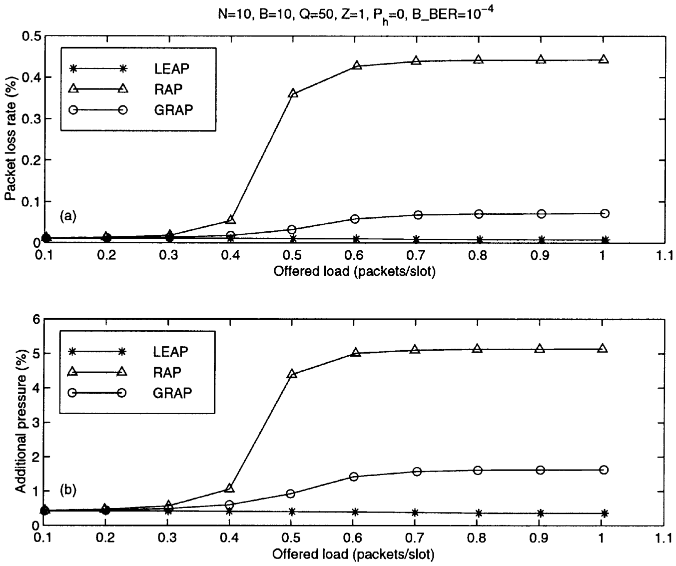


Fig. 10. Percentage of dropped data packets due to number of transmission attempts reaching R_LIM (top). Additional pressure on station queues due to data packet retransmissions (bottom).

provide a more fair comparison, this scenario was simulated in an unreliable environment with $B\_\mathrm{BER} = 10^{-4}$ and the rest of the simulation parameters remain similar to those of Network N_1. It can be seen that the number of data packets that exceed the maximum number of transmission attempts (and are thus dropped) rises significantly for RAP and GRAP for medium and high loads due to increased occurrence of collisions (top of Fig. 10), while this percentage for LEAP remains very low and is mainly due to channel unreliability. Finally, the increased number of transmission attempts per data packet in RAP and GRAP at medium and high loads due to collisions causes an additional pressure on station queues as backlogged packets remain in the queues waiting for retransmission. This additional pressure due to backlogged packets is shown in Fig. 10. It can be seen that the additional pressure for LEAP is very small, since it is collision free and retransmission attempts are only due to reception errors.

Finally, it is interesting to note that the performance superiority of LEAP over RAP and GRAP will also hold for the case of traffic characterized by homogeneous packet arrivals rather than bursty ones. This is due to the same reasons that were mentioned above: an increased number of occurring collisions when the number of active stations per polling cycle approaches the number of random addresses, P in RAP; and the fact that LEAP is collision free and requires less control overhead than RAP-GRAP.

VI. Conclusion

This paper proposed the LEAP protocol designed for bursty traffic wireless LANs. The protocol is able to achieve significantly higher throughput and lower delay values compared to the RAP and GRAP polling protocols under bursty traffic conditions. The main characteristics of the proposed protocol are: a) it achieves a high performance, even when the offered traffic is bursty; b) it is self-adaptive [19]; and c) it is easier to implement than RAP and GRAP. Each station takes a fraction of the bandwidth in proportion to its needs. The only requirement is the existence of a processor at the AP which implements the learning algorithm. On the other hand, the RAP and GRAP protocol implementation demands extra hardware for the CDMA-based contention stage.

Acknowledgment

The authors wish to thank Prof. Y. Fang and the three anonymous reviewers for their insightful and useful comments.

References

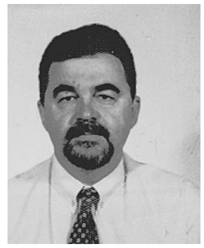
[1] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, *Wireless Networks*. New York: Wiley, 2003.
[2] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Design alternatives for wireless local area networks," *Int. J. Commun. Syst.*, vol. 14, pp. 1–42, Feb. 2001.
[3] A. C. V. Gummalla and J. O. Limb, "Design of an access mechanism for a high speed distributed wireless LAN," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1740–1750, Sept. 2000.
[4] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Using learning automata for adaptive push-based data broadcasting in asymmetric wireless environments," *IEEE Trans. Veh. Technol.*, to be published.
[5] G. I. Papadimitriou and D. G. Maritsas, "Learning automata-based receiver conflict avoidance algorithms for WDM broadcast-and-select star networks," *IEEE/ACM Trans. Networking*, vol. 4, pp. 407–412, June 1996.
[6] A. C. V. Gumalla and J. O. Limb, "Wireless medium access control protocols," *Commun. Surveys*, 2000.
[7] K.-C. Chen, "Medium access control of wireless LANs for mobile computing," *IEEE Network Mag.*, pp. 50–63, Sept./Oct. 1994.
[8] K.-C. Chen and C.-H. Lee, "RAP—a novel medium-access control protocol for wireless data networks," in *Proc. IEEE GLOBECOM*, 1993, pp. 1713–1717.
[9] ——, "Group randomly addressed polling for wireless data networks," in *Proc. IEEE ICC*, 1994, pp. 1713–1717.
[10] J.-W. Dai, "The stability of randomly addressed polling protocol," *IEICE Trans. Commun.*, pp. 1502–1508, Oct. 1997.
[11] Z. Zhang and A. S. Acampora, "Performance of a modified polling strategy for broadband wireless LANs in a harsh fading environment," in *Proc. IEEE GLOBECOM*, 1991, pp. 1141–1146.
[12] R. J. Haines and A. H. Aghvami, "Indoor radio environment considerations in selecting a media access control protocol for wideband radio data communications," in *Proc. IEEE ICC*, 1994, pp. 1306–1311.
[13] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Applications*. New York: Pergamon, 1994.
[14] K. S. Narendra and M. A. L. Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp. 262–269, May 1980.
[15] S. M. Nikolsky, *A Course of Mathematical Analysis*. Moscow, Russia: MIR Publishers, 1977.
[16] E. Gilbert, "Capacity of a burst noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, Sept. 1960.
[17] M. Zorzi, R. R. Rao, and L. B. Milstein, "On the accuracy of a first-order Markov model for data transmission on fading channels," in *Proc. ICUPC*, Tokyo, Japan, Nov. 1995, pp. 211–215.
[18] G. I. Papadimitriou and A. S. Pomportsis, "Self-adaptive TDMA protocols: a learning automata-based approach," in *Proc. IEEE ICON*, 1999.
[19] A. Farago, A. D. Myers, V. R. Syrotiuk, and G. V. Zaruba, "Meta-MAC protocols: automatic combination of MAC protocols to optimize performance for unknown conditions," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1670–1681, Sept. 2000.

P. Nicopolitidis received the B.S. and Ph.D. degrees in computer science from the Department of Informatics, Aristotle University, Thessaloniki, Greece, in 1998 and 2002, respectively.

His research interests are in the areas of wireless local area networks and mobile communications. He is co-author of the book *Wireless Networks* (New York: Wiley, 2003).

G. Papadimitriou (M'89–SM'02) received the Diploma and Ph.D. degrees in computer engineering from the University of Patras, Patras, Greece, in 1989 and 1994, respectively.

From 1989 to 1994, he was a Teaching Assistant at the Department of Computer Engineering of the University of Patras and a Research Scientist at the Computer Technology Institute, Patras, Greece. From 1994 to 1996, he was a Postdoctorate Research Associate at the Computer Technology Institute. From 1997 to 2001, he was a Lecturer at the Department of Informatics, Aristotle University, Thessaloniki, Greece. Since 2001, he has been an Assistant Professor at the Department of Informatics, Aristotle University. His research interests include wireless networks, optical networks and learning automata. He is a member of the Editorial Board of the *International Journal of Communication Systems*. He is also an Associate Editor of the journal *Simulation: Transactions of the Society for Modeling and Simulation International*. He is co-author of the books *Wireless Networks* (New York: Wiley, 2003) and *Multiwavelength Optical LANs* (New York: Wiley, 2003). He is the author of more than 70 refereed journal and conference papers.



A. S. Pomportsis received the B.S. degree in physics and the M.S. degree in electronics and communications, both from the University of Thessaloniki, Thessaloniki, Greece, and the Diploma in electrical engineering from the Technical University of Thessaloniki, Thessaloniki, Greece. In 1987, he received the Ph.D. degree in computer science from the University of Thessaloniki.

Currently, he is a Professor at the Department of Informatics, Aristotle University, Thessaloniki, Greece. He is co-author of the books *Wireless Networks* (New York: Wiley, 2003) and *Multiwavelength Optical LANs* (New York: Wiley, 2003). His research interests include computer networks, learning automata, computer architecture, parallel and distributed computer systems, and multimedia systems.