IEEE MELECON 2004, May 12-15, 2004, Dubrovnik, Croatia

A High Performance QoS Supportive Protocol for Wireless LANs

T. D. Lagkas, G. I. Papadimitriou, *Senior Member, IEEE*, and A. S. Pomportsis

Department of Informatics, Aristotle University, Box 888, 54124 Thessaloniki, Greece.
e-mail: g.i.papadimitriou@ieee.org

Abstract – A QoS supportive Adaptive Polling (QAP) protocol for wireless LANs is introduced. QAP operates under an infrastructure wireless LAN, where an Access Point (AP) polls the wireless nodes in order to grant them permission to transmit. The polled node sends data directly to the destination node. We consider bursty traffic conditions, under which the protocol operates efficiently. The polling scheme is based on an adaptive algorithm according to which it is most likely that an active node is polled. Also, QAP takes into account packet priorities, so it supports QoS by means of the Highest Priority First packet buffer discipline and the priority distinctive polling scheme. Lastly, the protocol combines efficiency and fairness, since it prohibits a singe node to dominate the medium permanently. QAP is compared to the efficient learning automata-based polling (LEAP) protocol, and is shown to have superior performance.

I. INTRODUCTION

Lately, there has been a great interest in the wireless communication networks which support high quality services and combine asynchronous communication, such as file transfer, and time bounded communication, such as streaming video. In general, the wireless networks have some special characteristics, which make the design of an appropriate medium access control protocol rather difficult [1]–[4]. Generally, in a wireless network the links are not reliable, the bit-errors are more often, and the topology changes in a continuous way. Furthermore, a modern wireless network needs QoS support.

In this paper, we propose QoS supportive Adaptive Polling (QAP), a new WLAN protocol designed for bursty traffic that supports QoS. An adaptive polling algorithm tends to poll the nodes, which are actually active, without having direct feedback about their current status [5]. An infrastructure WLAN topology is considered, where there is an access point (AP) that is only responsible for polling the mobile nodes in order to give permission to transmit. The adaptive polling algorithm takes into account the priorities of the data packets that are broadcasted by the mobile stations, in order to decide which node to poll [6]. Furthermore, every node implements a Highest Priority First (HPF) packet buffer discipline, which contributes in the QoS support. It is shown that the introduced protocol manages bandwidth assignment in an effective and fair way.

The paper is organized as follows. Section II discusses other WLAN polling protocols emphasizing on the learning automata-based polling (LEAP) protocol. In Section III, the QAP protocol is analyzed, and specifically the polling scheme is examined, the priority model of QAP is presented, and the node choice mechanism is approached in analytical way. Section IV presents the simulation environment and the simulation results, which show the performance superiority of the QAP protocol, comparing the proposed protocol and the LEAP protocol. Also, the QoS support of QAP is revealed. Section V concludes the paper.

II. WLAN POLLING PROTOCOLS

The polling protocols are popular WLAN MAC protocols for infrastructure networks [7]. The Randomly Addressed Polling (RAP) protocol provides zero wrong polls, but it gives a rather increased overhead and high collision probability [8]. According to it, the AP forwards all the packets to their destinations and CDMA or FDMA transmission is demanded for the node-to-AP communication. Apart from the high collision probability, RAP supports no QoS at all. GRAP is an improvement of RAP [9]. It uses super-frames and divides active nodes to groups. GRAP provides a minimum QoS support by allowing the nodes that carry time bounded packets to join any group for contention. This protocol performs better than the original RAP protocol, but the provided throughput and packet delay are still not satisfactory.

The LEAP protocol is also a wireless polling protocol, but it is based on a different concept [10]. It assumes a cellular topology as it was described above, however it considers direct communication between the mobile nodes (the AP is not a packet forwarder). This protocol defines that the AP chooses the node that will be given permission to transmit by using choice probabilities. Four control packets are used: POLL, NO_DATA, BUFF_DATA, and ACK, with duration t_{POLL}, t_{NO_DATA}, t_{BUFF_DATA}, and t_{ACK}, respectively. The propagation delay is t_{PROP_DELAY}, and a data packet transmission lasts t_{DATA}. According to this polling scheme, the maximum polling cycle duration is $t_{POLL} + t_{BUFF\_DATA} + t_{DATA} + t_{ACK} + 4{*}t_{PROP\_DELAY}$. When the AP detects that the polled node transmits data then it is assumed that it is active, so its choice probability is increased. Respectively, when the polled node responds with a NO_DATA packet or the AP fails to receive feedback, then it is assumed inactive or that there is a bad link, so the node's choice probability is decreased [11]. According then to this protocol, AP examines the feedback that gets during a polling cycle (j) in order to update the choice probabilities and select the node to poll at the next polling cycle ($j + 1$). When the choice probability of node k is increased, it becomes $P_k(j + 1) = P_k(j) + L(1 - P_k(j))$, and when it is decreased it becomes $P_k(j + 1) = P_k(j) - L(P_k(j) - a)$, where L, a are constants. Finally, the choice probabilities are normalized. LEAP is an efficient WLAN protocol and performs clearly better than RAP and GRAP, but the main drawback of the protocol, which is rather important, is that it does not support QoS.

III. THE QAP PROTOCOL

QAP also assumes a cellular topology where the AP polls the nodes in order to give them permission to transmit. The used polling scheme is similar to the polling scheme of LEAP, however it is more efficient due to the lower overhead. The QAP protocol uses the control packets that were mentioned before, except from the BUFF_DATA packet, which schemes to be rather useless. The possible polling events are depicted schematically in Fig. 1.

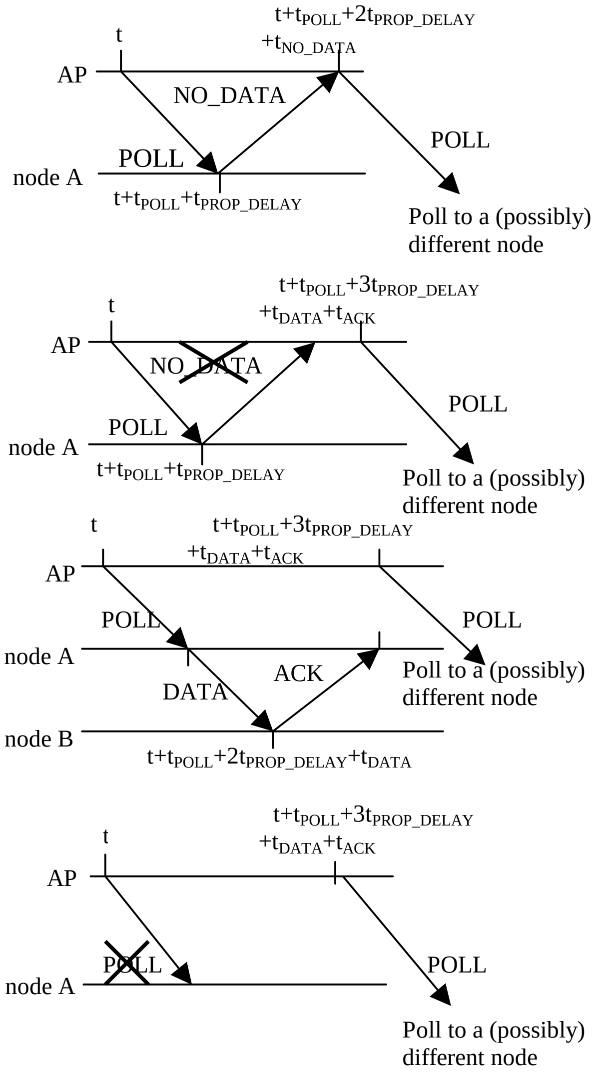

Fig. 1. The polling scheme of the QAP protocol

It is obvious that this polling scheme reduces the overhead, since no BUFF_DATA control packet is considered. This results in shorter waiting times, and finally in a shorter polling cycle. Specifically, the maximum duration of the polling cycle of QAP is $t_{BUFF\_DATA} + t_{PROP\_DELAY}$ shorter than the polling cycle of LEAP. In LEAP, the role of the BUFF_DATA is to inform the AP of the oncoming data transmission, but since the AP is able to detect the polled node's data broadcast, no BUFF_DATA is needed. The above polling scheme, which is collision free, takes into account the bursty nature of the traffic, the bursty appearance of bit-errors, and the need for minimal overhead.

The QAP protocol supports QoS by using packet priorities. The first utilization of the packet priorities takes place in the packet buffer. QAP uses the Highest Priority First (HPF) buffer discipline, according to which the packets that carry the highest priorities are served first. Among the packets of the same priority we use First In First Out (FIFO) buffer discipline, based on the generation time of the packets.

The QAP protocol updates the choice probabilities of the nodes according to their status (active or not) and their priority. According to the "active nodes" concept, it is clearly considered that under bursty traffic conditions it is most probable that a node (k) which transmits a data packet has more packets in the buffer [5]. So, this node is inserted in the set of the active nodes, which are more probable to be polled. If the AP assumes that the polled node transmitted no data, then it consider it to be inactive. The node choice procedure is depicted in Fig. 2, where N is the total number of nodes in the cell and M is the number of active nodes.

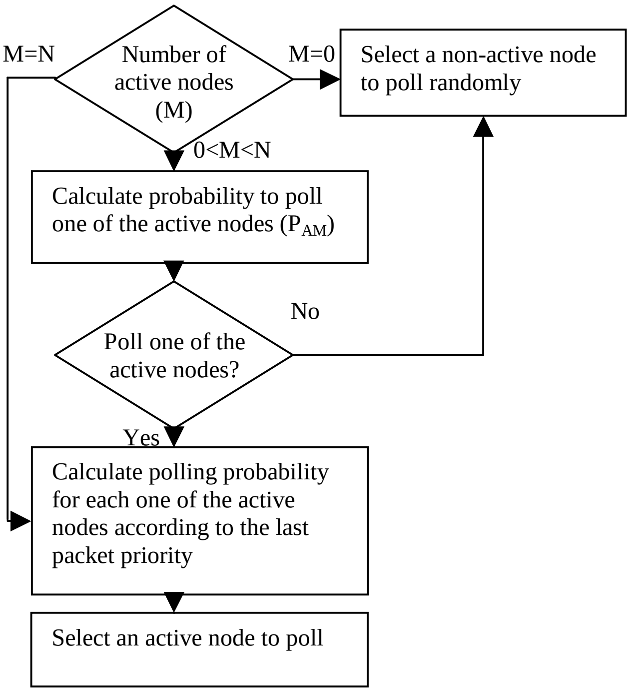

Fig. 2. Overview of the node choice mechanism of QAP

The probability P_{AM} is given by the equation $P_{AM} = P_A + P_Q$. The variable P_A depends on the number of active nodes, and it holds that $P_A = P_{A1} + (M-1)\times(1-P_{A1})/(N-1)$. We define that the probability to choose one of the active nodes when there is only one active node is P_{A1}. In order to provide fairness, we set by default $P_{A1} = 0.9$. The 3D-plot of the two-variable function $P_A(M, P_{A1})$ is shown in Fig. 3, where we assume $N = 10$.

P_Q is the second addendum in the equation that gives P_{AM}. The concept is to increase P_{AM} (positive P_Q) when A_Q is greater than the mean priority level ($Q_{max}/2$), and decrease P_{AM} (negative P_Q) when A_Q is less than the mean priority level. Q_{max} is the highest packet priority and it holds $Q_{max} = PLevels - 1$. It finally holds that $P_Q = P_{Qm}\times(A_Q - Q_{\max}/2)/(Q_{\max}/2)$.The parameter P_{Qm} defines the maximum and minimum values of P_Q, and affects it in a proportional way. It becomes clear that this method enhances QoS support in the node choice mechanism. The 3D-plot of the two-variable function $P_Q(A_Q, P_{Qm})$ is shown in Fig. 4, where we assume $PLevels = 4$. We can see the variation of P_Q, and the way A_Q and P_{Qm} affect the value of P_Q.

So, the function that gives the probability to poll one of the active nodes (P_{AM}), is the following:

$$P_{AM} = P_{A1} + (M-1)\times\frac{1-P_{A1}}{N-1} + P_{Qm}\times\frac{A_Q - \frac{Q_{\max}}{2}}{\frac{Q_{\max}}{2}}$$

Of course, we finally limit P_{AM} between 0 and 1. The 3D-plot of the two-variable function $P_{AM}(M, A_Q)$ is shown in Figure 5, where we consider $N = 10$ and we assume the default values $P_{A1} = 0.9$, $P_{Qm} = 0.03$, and $Q_{max} = 3$.

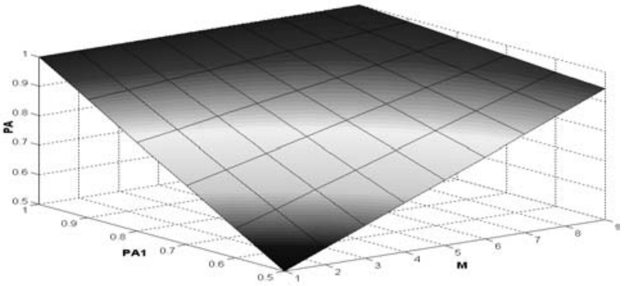

Fig. 3. Probability polling an active node (PA), without taking into account priorities, as a function of the number of active nodes (M) and the probability polling a single active node (PA1)

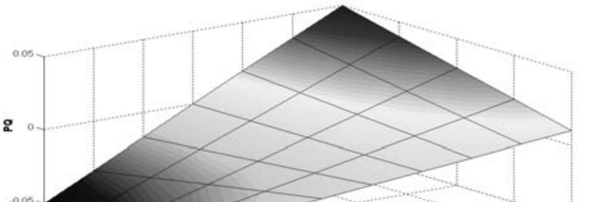

Fig. 4. The variation of the probability polling an active node (PQ), depending on the packet priorities, as a function of the maximum variation (PQm) and the average priority of the active nodes (AQ)

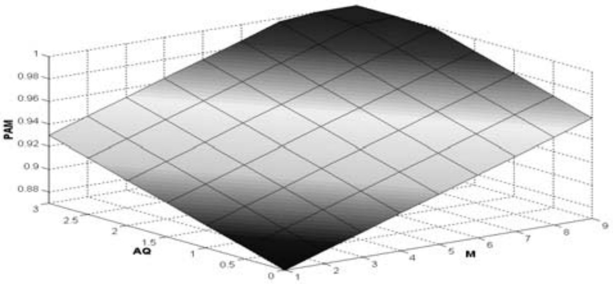

Fig. 5. Probability polling one of the active nodes (PAM) as a function of the number of active nodes (M) and the average priority of the active nodes (AQ)

If an active node is going to be polled, the relative probability of choosing node (k) is given by the equation $P_C(k) = q + 1$, where q is the priority of the specific node. The choice probabilities of all the active nodes are calculated and then are normalized. So, the actual choice probability of node (k) is $\Pi_C(k) = P_C(k)\big/\sum_{i=1}^{N} P_C(i)$.

IV. PERFORMANCE EVALUATION

In order to compare the QAP protocol against LEAP, we developed a simulation program in C++. The bursty traffic was simulated as stated in [11]. The offered load is R packets per slot, and the buffer size is Q packets. When a packet is generated, it is assigned a packet priority (range [0, $PLevels - 1$]). The packets of the same burst are assigned the same random priority.

In the developed simulation environment, the condition of any wireless link was modeled using a finite-state machine with three states [12], [13]. State G is characterized by a small BER, given by G_BER. State B is characterized by increased BER, given by B_BER. State U denotes that the pair of communication nodes is out of range of one another (hidden nodes). The time spent by a link in states G, B and H is exponentially distributed, but with different average values, given by the parameters TG, TB, TH, respectively. The status of a link probabilistically changes between the three states, defined by the parameter P_h. For example, for $P_h = 0.1$, there is a 0.1 probability that two nodes A and B are hidden. Some default values are: $TG = 3$ *sec*, $TB = 1$ *sec*, $TH = 0.5$ *sec*, $G\_BER = 0$, $B\_BER = 10^{-6}$ and $P_h = 0$ for relatively "clean" network conditions, and $B\_BER = 10^{-4}$ and $P_h = 0.1$ for rather not "clean" wireless links. We also considered $N = 10$, $Q = 50$, $L\_LEAP = 0.1$, $a\_LEAP = 0.03$, the control packet size (*cpSize*) equal to *160* bits, the default data packet size (*dpSize*) equal to *6400* bits, $PLevels = 4$, $P_{A1} = 0.9$, and $P_{Qm} = 0.03$. Every simulation was carried out until *400000* data packets where successfully received. The random number generator that is used by the simulator is a classic multiplicative congruential random number generator with period 2^{32} provided by ANSI C. The simulation results presented in this section are produced by a statistical analysis based on the "sequential simulation" method [14]. For this statistical analysis we used 95% confidence intervals. The relative statistical error threshold varies depending on the meaning of the metric and the magnitude of the produced value. However, this threshold was usually assumed to be lower than 2% and never exceeded 5%.

The simulation results have shown that the QAP protocol in comparison to LEAP performs superiorly in any network condition, mainly due to the lower overhead, the optimized polling scheme, and the efficient polling algorithm. In Fig. 6, the simulated network has increased BER, and the "hidden nodes" problem is present. Specifically, we assume $B\_BER = 10^{-4}$ and $P_h = 0.1$. In a rather harsh environment like this, QAP provides packet delays clearly lower than the delays provided by LEAP. We assume that high priority packets are the packets which are assigned a priority higher than $(PLevels - 1)/2$. The corresponding curve is a proof of the QoS support.

When the data packet size gets small compared to the control information, QAP has great advantage, which is shown by the high throughout and the low packet delay. In Fig. 7, it is obvious that, for $dpSize = 800$, QAP provides significantly lower packet delays than LEAP.

In Fig. 8, we assume $B\_BER = 10^{-6}$, $P_h = 0$, and $R = 1$. It is shown that the delay of the high priority packets remains impressively stable, while the overhead alters. Specifically, we plot the average packet delay versus the *dpSize*, while keeping the *cpSize* stable.

Assuming the same network conditions, we plotted the high priority packet delay as percentage of the low priority packet delay. In Fig. 9, it is shown that the high priority packets are favored in a relatively greater degree under harsh network conditions, which means that the QAP protocol ensures QoS support in any case.

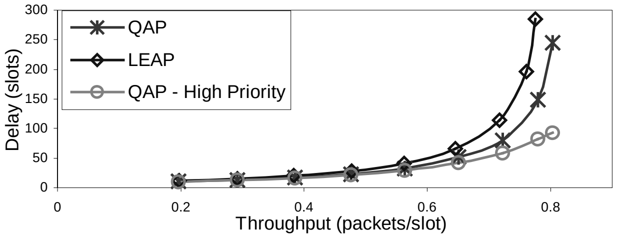

Fig. 6. Average packet delay versus throughput, where we plot for packet loss rate lower than 15%

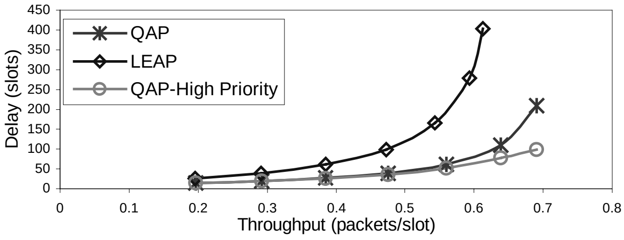

Fig. 7. Average packet delay versus throughput, where we plot for packet loss rate lower than 20%

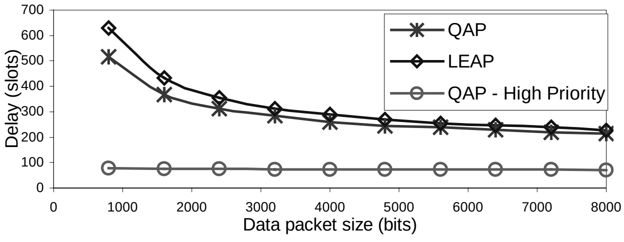

Fig. 8. Average packet delay versus data packet size, where we plot for any packet loss rate

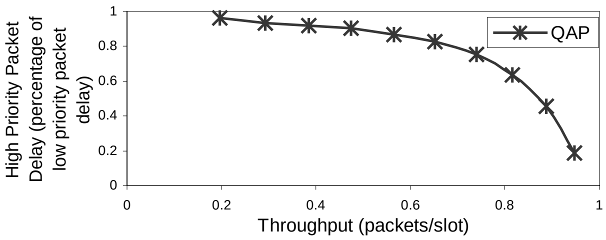

Fig. 9. High priority packet delay as a percentage of the low priority packet delay versus throughput

V. CONCLUSION

This work proposed the QoS supportive Adaptive Polling (QAP) protocol for wireless LANs. The protocol is capable of operating efficiently under bursty traffic conditions. The QAP protocol performs always superiorly than LEAP. The protocol is based on a self-adaptive polling algorithm [15], which decreases the number of wrong polls. The overhead is reduced and the polling scheme is generally optimized. QAP is able to support different kinds of traffic, by using packet priorities. QoS is always supported. This model is not difficult to implement, since the polling scheme based on the active nodes and the node priorities is rather simple. Furthermore, no simultaneous transmissions take place.

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] A. S. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[5] G. I. Papadimitriou, and A. S. Pomportsis, "Adaptive MAC protocols for broadcast networks with bursty traffic," *IEEE Trans. Commun.*, vol. 51, no. 4, April 2003.

[6] G. I. Papadimitriou, T.D. Lagkas, and A. S. Pomportsis, "HIPERSIM: A sense range distinctive simulation environment for HiperLAN systems", *J. Simulation*, to be published.

[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] P. Nicopolitidis and G. I. Papadimitriou, "Learning automata-based polling protocols for wireless LANs," *IEEE Trans. Commun.*, vol. 51, no. 3, March 2003.

[11] G. I. Papadimitriou, and A. S. Pomportsis, "Learning automata-based TDMA protocols for broadcast communication systems with bursty traffic," *IEEE Commun. Letters*, vol.4, no.3, March 2000.

[12] E. Gilbert, "Capacity of a burst noise channel," *Bell Syst. Tech. J.*, vol. 39, pp.1253-1265, Sept. 1960.

[13] 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.

[14] Krzysztof Pawlikowski, Hae-Duck Joshua Jeong, and Jong-Suk Ruth Lee, University of Canterbury, "On Credibility of Simulation Studies of Telecommunication Networks", *IEEE Communications Magazine*, January 2002.

[15] 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.