Priority Oriented Adaptive Polling for wireless LANs

T. D. Lagkas*, G. I. Papadimitriou, *Senior Member, IEEE*, P. Nicopolitidis, and A. S. Pomportsis
Department of Informatics, Aristotle University, Box 888, 54124 Thessaloniki, Greece.
E-mail: tlagkas@csd.auth.gr

Abstract

Today's wireless LANs require efficient integration of multimedia and traditional data traffic. Multimedia network applications are time-bounded and have stricter QoS demands. The IEEE 802.11e workgroup is standardizing a new QoS enhanced access scheme for wireless networks. It is based on a mechanism called Enhanced Distributed Channel Access (EDCA). EDCA seems capable of differentiating the traffic, however, it exhibits great overhead that limits the actually available bandwidth and degrades the overall performance. This work proposes an alternative protocol which could be used in place of EDCA. The Priority Oriented Adaptive Polling (POAP) is collision free, it prioritizes the different kinds of traffic, and it is able to provide QoS for all types of multimedia network applications, while efficiently supporting background data traffic. POAP compared to EDCA, provides higher channel utilization, distributes network resources to the mobile stations adapting to their real needs, and generally exhibits superior performance.

1. Introduction

Today, voice, audio and video have to be efficiently transmitted along with the traditional data traffic. Real-time applications require QoS support, because they are time-bounded, while slightly unreliable connections are allowed. On the other hand, data traffic does not demand particularly low delay, but reliability is essential. Thus, modern networks should be able to meet all types of traffic requirements.

The IEEE 802.11e [1] workgroup has enhanced the Distributed Coordination Function (DCF), with QoS support, proposing EDCA, which is the essential part of the 802.11e MAC protocol. However, it causes high network overhead, which degrades the system's performance, thus, efficiently serving multiple sources of different types of traffic becomes very difficult.

There are various MAC protocols proposed in the literature for different kinds of network conditions [2]–[9]. This work proposes the Priority Oriented Adaptive Polling which is able to be built into the access scheme defined by IEEE 802.11e. It belongs to the centralized access protocols, however, no bandwidth reservation is required. It efficiently supports simultaneous real-time and background traffic, by taking into account traffic priorities and the current status of the stations. It should be noticed that despite the fact that EDCA is a distributed scheme, most network scenarios that consider real-time traffic assume infrastructure topology with the use of an Access Point (AP) for packet relay and interconnection to the backbone network. POAP tries to exploit this common topology by using the AP for access control. This paper assumes that stations are able to communicate directly when in range, however the model where the AP acts as a packet forwarder could be also used. According to [1], the IEEE 802.11e access model also provides a Direct Link Protocol (DLP) as an extra feature.

This paper is organized as follows. Section 2 presents the EDCA access control. In Section 3, POAP is analyzed, focusing on the polling scheme, the priority model and the station choice algorithm. Section 4 presents our simulator, the network scenario, and the results, which prove the efficiency of POAP by comparing it with EDCA. Section 5 concludes the paper.

2. The IEEE 802.11e Enhanced Distributed Channel Access

The legacy IEEE 802.11 MAC does not support QoS. However, some modifications that enhance partial QoS support have been proposed [10]. The need for QoS has led IEEE to form the 802.11e workgroup [1]. The mandatory access scheme for the IEEE 802.11e access mechanism is the EDCA protocol. EDCA is the QoS enhanced version of the Distributed Coordination Function (DCF) employed by the legacy

Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06)
0-7695-2588-1/06 $20.00 © 2006 IEEE

IEEE 802.11 MAC. When a station needs to transmit a packet and the channel is busy, it waits until the medium becomes idle and then defers for an extra time interval, called Arbitrary Distributed Interframe Space (AIFS). If the channel stays idle for AIFS, the station starts backoff by selecting a random number of slots from a contention window (CW). The minimum size of CW is *CWmin*, while the maximum is *CWmax*. The traffic differentiation is based on the use of four buffers. Each one corresponds to an access category (AC) mapped to specific user priorities (UPs). The highest ACs are more probable to gain access, since they are assigned low *CWmin*, *CWmax* and AIFS values. An additional RTS/CTS (Request To Send/Clear To Send) mechanism is defined to solve the hidden terminal problem. However, it increases the overhead and does not provide a completely collision-free medium.

This model provides only minimal QoS. The backoff procedure leads to waste of bandwidth, however, it is necessary in order to avert collisions. Furthermore, the "hidden terminal" problem leads to collisions despite of the backoff mechanism. The use of the RTS/CTS handshake limits this problem, however, it increases the overhead. EDCA definitely enhances DCF with QoS support, however, it is shown that it can actually serve only limited traffic of low QoS demands. For these reasons, we propose alternatively the POAP protocol which greatly reduces the network overhead and optimizes the priority model, providing significantly stricter QoS and generally higher system performance, as it will be shown in the next sections.

3. The POAP protocol

3.1. The polling scheme

According to POAP, the AP polls the stations in order to give them permission to transmit. The employed polling scheme eliminates the collisions and causes low overhead. The protocol uses the POLL, NO_DATA, and STATUS control packets, with transmission duration t_{POLL}, t_{NO_DATA}, and t_{STATUS}, respectively. A STATUS packet is marked as ACK or NACK according to the specific case. The transmission duration of a DATA packet is t_{DATA} and the propagation delay is t_{PROP_DELAY}. The possible polling events are depicted schematically in Figure 1.

- *The AP polls an inactive station* (Figure 1a): The AP sends POLL to the mobile station at time *t* and waits for feedback. The station responds with a NO_DATA packet, which is received by the AP at $t + t_{POLL} + t_{NO\_DATA} + 2t_{PROP\_DELAY}$. Then, the latter initiates a new polling procedure.

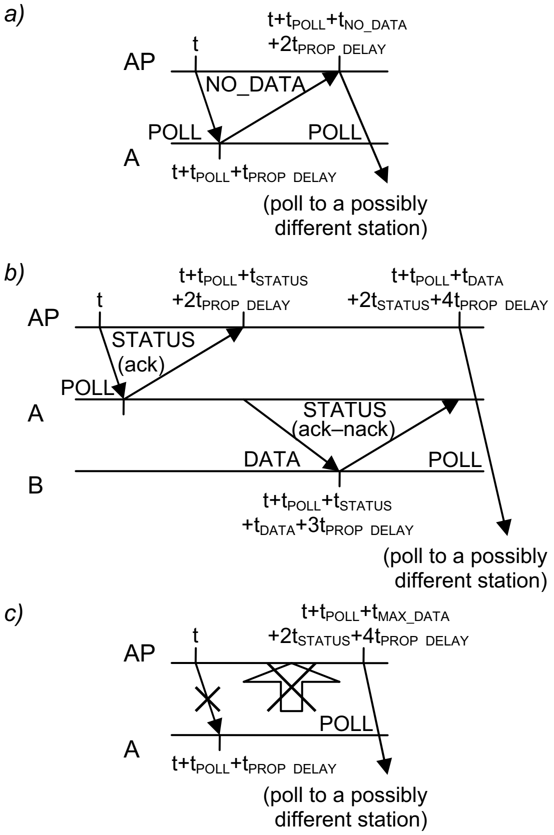

Figure 1. The polling scheme of POAP

- *The AP polls an active station* (Figure 1b): The AP sends POLL to the station at time *t* and waits for feedback. The station replies with a STATUS packet marked as ACK, which carries the destination address and the size of the following DATA packet. Then, the polled station starts transmitting the DATA packet directly to the destination. Upon successful reception, the destination broadcasts a STATUS packet marked as ACK. Otherwise, if the reception fails but the station has realized that the specific packet is destined to it, it responds with a STATUS packet marked as NACK. The transmission of a NACK is not wasted time, since either way the stations had to wait for a possible ACK. The AP can proceed to a new poll at time $t + t_{POLL} + t_{DATA} + 2t_{STATUS} + 4t_{PROP\_DELAY}$. It should be noticed that we consider variable DATA packet size, thus, t_{DATA} is not static.

- *The communication fails* (Figure 1c): In case the station does not successfully receive the POLL packet, the polling fails. The AP has to wait for the maximum polling cycle before proceeding to a new poll, since it has to be certain that it will not collide with a possible on going transmission. When the POLL packet is received successfully by the polled station, but then the

AP fails to receive any feedback, it waits for the maximum polling cycle similarly to the previous case. The duration of the maximum polling cycle is $t_{POLL} + t_{MAX\_DATA} + 2t_{STATUS} + 4t_{PROP\_DELAY}$, where t_{MAX_DATA} is the duration of the largest allowed DATA packet. At the end of this cycle, it is certain that the medium is idle in any event. When such a communication failure occurs, the AP lowers the probability to choose this station in the new polling procedure assuming a bad link between them. However, it is most likely that the AP will receive some feedback either from the polled or the destination station.

The proposed polling scheme provides efficient feedback and low overhead. The purpose of the control packets is to keep the concerned stations informed of the network's status and minimize the idle intervals. The AP needs to monitor the transmissions so that it can proceed to the next poll right after the completion of a communication. For this reason, it has to be aware of the actual duration of the specific polling cycle. In order to gain this knowledge, the AP just has to successfully detect the NO_DATA packet or the STATUS ACK packet, which contains the duration of the following data transmission, or the DATA packet from the polled station or the STATUS ACK-NACK packet from the destination station. In all these cases, the AP is aware of the polling cycle duration and can proceed to the next poll without wasting any time. Actually, when POLL is successfully received, then it is most likely that the AP will obtain the necessary feedback.

Furthermore, our purpose is to poll stations that are actually active in a way that QoS is provided. For this reason, the AP needs to be aware of the stations' status so as to avoid wasting time at polling inactive stations, favor the high priority traffic, and provide fairness.

3.2. The priority model

POAP adopts the packet priorities concept used by IEEE 802.11e to retain compatibility. However, it is differentiated in the way of choosing a packet to send.

Firstly, the priority (access category) of each packet buffer has to be considered, so that high priority traffic is favored. Also, in order to provide low packet delays, low packet drops and fairness among the ACs, the number of packets in each buffer is examined before choosing a packet. Specifically, heavy loaded buffers should have higher probabilities of transmitting. Lastly, the earlier generated packets must be favored.

The respective packet choice mechanism is depicted in Figure 2. Initially, we examine if there are any buffered packets, otherwise, the polled station replies with a NO_DATA message. Then, each one of the four buffers is examined in order to calculate its normalized priority (P_{PR}) and normalized number of buffered packets (P_B). Specifically, assuming that the priority of buffer *i* is equal to $p[i] = i + 1$, so that it is not null for *AC[0]*, then it holds: $P_{PR}[i] = p[i] \Big/ \sum_{k=0}^{3} p[k]$. Also, if *b[i]* is the number of packets carried by buffer *i*, then it holds: $P_B[i] = b[i] \Big/ \sum_{k=0}^{3} b[k]$.

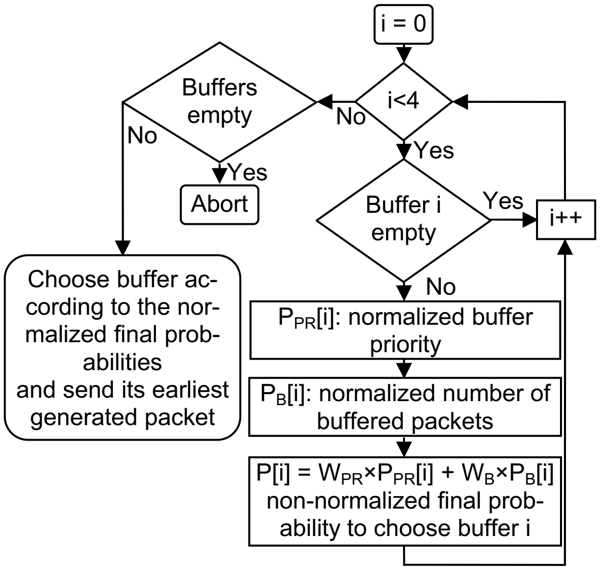

Figure 2. The packet choice mechanism

The buffer priority and the number of buffered packets should have different contribution to the final buffer choice probability (*P*). Thus, we use the weights W_{PR} (default *6*) and W_B (default *2*) for P_{PR} and P_B, respectively. The non-normalized choice probability for buffer *i* is: $P[i] = W_{PR} \times P_{PR}[i] + W_B \times P_B[i]$. When W_{PR} is high compared to W_B, it is most probable that a high priority buffer will be chosen for transmission regardless of the buffer load. On the other hand, if W_B is increased, it is more probable to choose a packet from a highly loaded buffer. The default values favor packet priorities over buffer load by a ratio of *3*. The normalized choice probability equals to: $P[i] \Big/ \sum_{k=0}^{3} P[k]$. After the buffer selection, the station chooses to send the earliest generated packet in it.

Before the AP decides which station to poll, it has to be well informed of their buffers' status. However, this feedback should not cause a great overhead. Thus, we exploit the ACK and NACK messages, which are already useful. Specifically, the STATUS packet apart from acknowledging receptions, it also carries its source's priority score, which is analyzed below.

When a station broadcasts a STATUS packet, it includes its priority score, which is an indication of the status of its buffered traffic. The priority score depends on the priority of each buffer and the number of packets it contains. For station *i*, the priority score is: $P_S[i] = \sum_{k=0}^{3} p[k] \times b[k]$. So, every time a STATUS packet is broadcasted, the AP examines it in order to update the stored priority score of the transmitting station. This way, the model provides efficient feedback for the AP with minimum network overhead. The gathered priority scores are then used by the AP to implement a high performance polling mechanism.

3.3. The station choice mechanism

The first factor taken into account by the algorithm that chooses the polled station is the priority score. Thus, it is most probable to poll a station with high priority traffic and large number of buffered packets, according to the definition of the priority score presented in the previous subsection. The second factor is the time that has elapsed since the last poll of each station (τ). Specifically, in order to provide fairness and avoid the total exclusion of stations that are inactive for quite long, the stations that have not been polled for a long time are favored to some degree. Furthermore, the AP, which also participates in the channel contention, is assigned a higher probability of getting access, since it usually plays a central role. Lastly, it should be noticed that the AP halves a mobile station's priority score, when it receives no feedback after polling it, assuming that the link between them is bad.

Figure 3 depicts the operation of the proposed algorithm that returns the station to be polled. Initially, we check if the AP has any buffered packets. If it has not, then it is not included in the station choice procedure. Then, the priority score of each considered station *i* is normalized: $P_P[i] = P_S[i] \big/ \sum_{k=0}^{M-1} P_S[k]$, where *M* is the number of stations considered by the algorithm. The time elapsed since its last polling is also normalized: $P_T[i] = \tau[i] \big/ \sum_{k=0}^{M-1} \tau[k]$. The non-normalized final probability of polling station *i* is: $P_{POLL}[i] = W_{PR} \times P_P[i] + W_T \times P_T[i]$, where W_T (default *1*) is the weight of the contribution of the P_T factor. A high W_T value provides extended fairness among stations, however, this way traffic differentiation via packet priorities fades. If the examined station *i* is the AP, then its non-normalized final probability of getting access is multiplied by the factor W_{AP} (default *10*): $P_{POLL}[i] = W_{AP}(W_{PR} \times P_P[i] + W_T \times P_T[i])$. Thus, the AP has W_{AP} times higher access chances. Lastly, the AP decides which station will be polled according to each one's normalized polling probability, which is for station *i*: $P_{POLL}[i] \big/ \sum_{k=0}^{M-1} P_{POLL}[k]$.

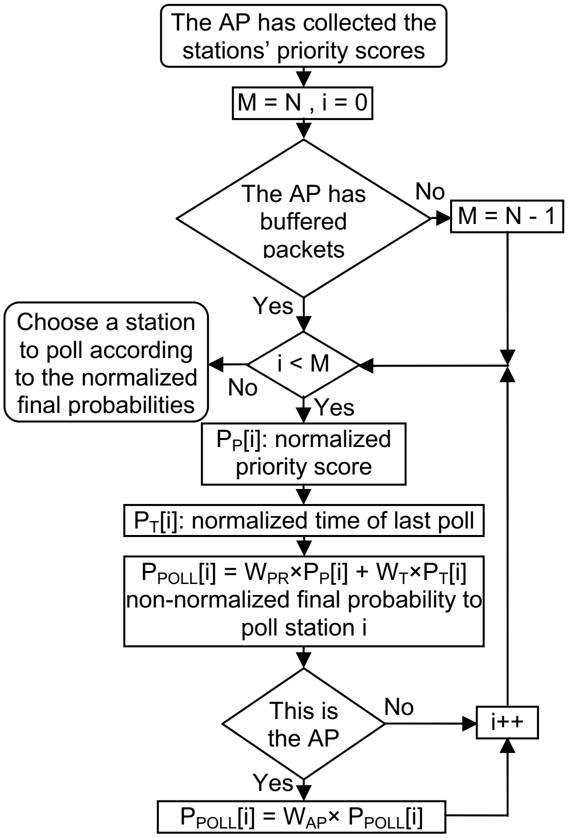

Figure 3. Station choice mechanism, where N is the number of stations including the AP

4. Performance evaluation

A C++ simulator was developed. The physical layer is the IEEE 802.11g, which provides up to *54* Mbps rates. In order to simulate realistic links, we assumed 3*6* Mbps data rate. The 802.11e super-frame was used.

The condition of any link was modeled using a finite-state machine with three states (Good, Bad, Hidden). We considered rather not "clean" conditions, in order to simulate more accurately the wireless environment. The AP-station links are considered to be more reliable than the inter-station links. Also, the hidden terminal issue is present in our network.

The results are produced by a statistical analysis based on the "sequential simulation" method [11]. We perform simulations, until the relative statistical error of the estimated mean value falls below an acceptable threshold. We used *95%* confidence intervals. The relative statistical error threshold varies depending on the meaning of the metric and the magnitude of its value. However, it was usually lower than *2%*.

We consider one bidirectional voice communication and one bidirectional video communication between the AP and each mobile station. Also, there is a bidirectional TCP flow between any two adjacent mobile stations. The simulation duration is *60* sec, every communication lasts for *30* sec, and a new flow is added every second. We simulated *14* topologies, starting with *2* mobile stations and finally reaching *28* stations, producing results for *10* to *140* one-way flows.

The traffic has realistic characteristics derived by the analysis of traces. We use TCP flows with features typical for file transfers. Voice communication is based on the G.726 codec. The new H.264 codec is employed for live video. Our purpose was to evaluate the capability of POAP and EDCA of handling voice, video, and background traffic simultaneously. Regarding packet losses, the majority of voice and video packet drops are caused by lifetime expiration, while the TCP drops are due to buffer overflow.

Regarding voice, in Figure 4, it can be seen that POAP exhibits lower packet delays and loss rates when the throughput is lower than *550* kbps, while EDCA performs better for higher values of the voice throughput. However, both protocols achieve similar maximum throughput. Despite the fact that POAP eventually causes higher packet delays, it keeps them below *18* ms which is very satisfactory for voice communications. The conclusion is that both schemes are able to provide QoS in voice transmissions. As it will be shown next, this behavior of the two protocols is due to the fact that POAP conserves recourses in order to serve video and TCP traffic as well, while EDCA favors the voice packets to such a degree that seems unable to simultaneously serve video and TCP flows.

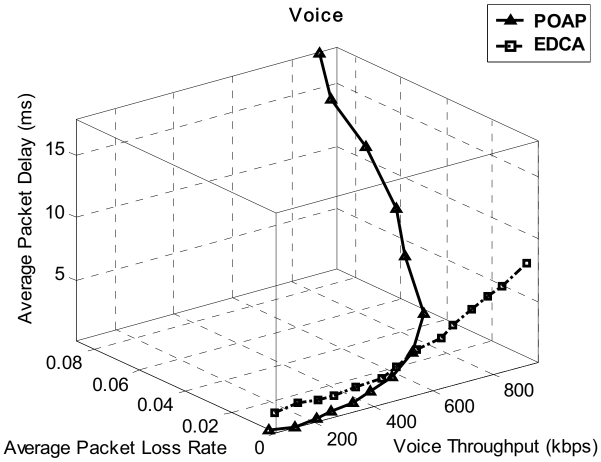

Figure 4. Voice: Average packet delay and packet loss rate versus voice throughput

Figure 5 shows that EDCA suffers from so many packet losses that it cannot really support live video when the video throughput is over *3* Mbps. POAP exhibits particularly low packet delays and loss rates when the video throughput is lower than *12* Mbps, while it achieves significantly high throughput.

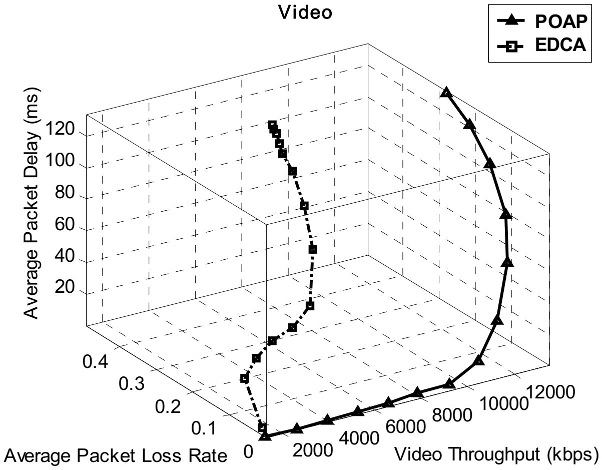

Figure 5. Video: Average packet delay and packet loss rate versus video throughput

According to Figure 6, POAP can efficiently support background traffic the same time it provides QoS for voice and video. It guarantees significantly lower packet delays and loss rates, while it achieves higher throughput than EDCA. The latter seems to perform generally worse than POAP when serving TCP traffic.

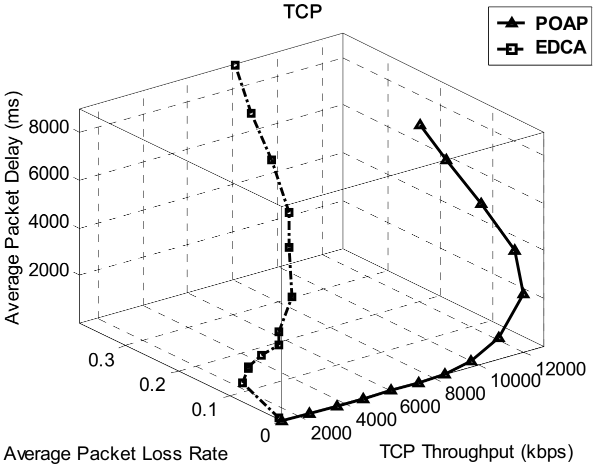

Figure 6. TCP: Average packet delay and packet loss rate versus TCP throughput

Figure 7, which concerns the whole traffic, shows that EDCA has steady performance till total throughput of *15* Mbps, while POAP is steady till the total throughput of *20* Mbps. Also, POAP exhibits lower bit delay, lower bit loss rate, and higher throughput.

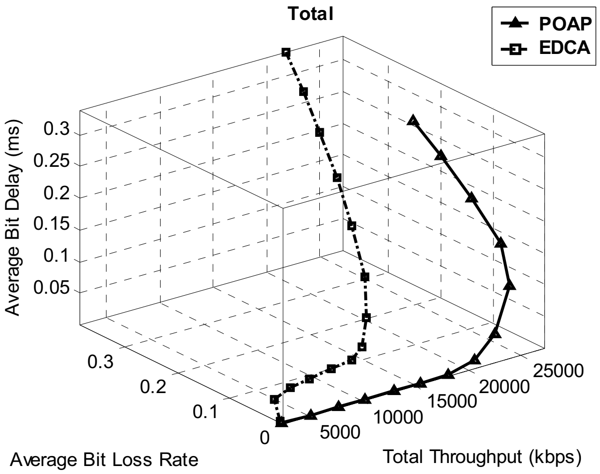

Figure 7. Total Traffic: Average bit delay and average bit loss rate versus total throughput

The fact is that POAP exhibits higher utilization than EDCA. The latter favors the voice traffic to such an extent that it is not actually capable of simultaneously supporting video and TCP traffic, which have to wait for long backoff times. The decentralized nature of EDCA leads to "dead" operation intervals, when no station is transmitting due to backoff. It also causes collisions. The RTS-CTS mechanism deals with this issue, however, it causes extra overhead. POAP is more deterministic. The AP continuously polls the stations, based on the feedback. This model eliminates the "dead" idle periods. Additionally, the polling scheme provides minimum overhead with optimized feedback and zero collisions. Since the AP is aware of the traffic status, it adapts the transmission opportunities granted to the stations accordingly. This mechanism leads to stable operation, where packet delay and loss rate are kept low. Notice that POAP ensures that no flow or station can dominate the medium. It provides fairness by considering the time since the last poll and the buffered bits. It is shown that POAP serves the flows for higher load values than EDCA.

5. Conclusion

The proposed Priority Oriented Adaptive Polling protocol can be adapted into the IEEE 802.11e MAC in place of EDCA. It assumes stations polling, resulting in a collision free medium. It should be noticed that despite the distributed nature of EDCA, its most common topology when serving integrated data is also an infrastructure one. POAP efficiently provides QoS to integrated time-bounded and background traffic. Traffic differentiation is based on the same priority assignment scheme employed by EDCA. In order to provide the AP with valuable feedback, POAP efficiently exploits the use of the control packets in the polling model. This information allows the AP to optimize its decisions about the access grants. It is shown that POAP exhibits higher utilization, since it eliminates the overhead caused by contention and collisions. Furthermore, it generally provides significantly lower packet delays, lower loss rate, and higher throughput. As future work, the network parameters could be further tuned. Also, POAP could be combined with a resource reservation scheme to provide guaranteed QoS.

References

[1] IEEE 802.11e WG, "P802.11e: Draft Amendment to Standard for Information Technology-- Telecommunications and Information Exchange Between Systems--LAN/MAN Specific Requirements-- Part 11 Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Medium Access Control (MAC) Quality of Service (QoS) Enhancements," Draft Version: D13 2005.

[2] T. D. Lagkas, G. I. Papadimitriou, and A. S. Pomportsis, "QAP: A QoS supportive Adaptive Polling Protocol for Wireless LANs," *Computer Communications, Elsevier*, Vol. 29, Issue 5, Mar. 2006, pp. 618-633.

[3] T. D. Lagkas, G. I. Papadimitriou, P. Nicopolitidis, and A. S. Pomportsis, "POAC-QG: Priority Oriented Adaptive Control with QoS Guarantee for Wireless Networks," *Proceedings of IEEE EUROCON 2005*, Belgrade, Serbia & Montenegro, Nov. 2005, pp. 1858-1861.

[4] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou and A. S. Pomportsis, *Wireless Networks*, Wiley, Jan. 2003.

[5] P. Nicopolitidis, G. I. Papadimitriou and A. S. Pomportsis, "Learning-Automata-Based Polling Protocols for Wireless LANs", *IEEE Transactions on Communications*, vol.51, no.3, March 2003, pp. 453-463.

[6] G. I. Papadimitriou, A. S. Pomportsis, "Adaptive MAC protocols for broadcast networks with bursty traffic", *IEEE Transactions on Communications*, vol.51, no.4, Apr. 2003, pp. 553-557.

[7] Imrich Chlamtac, Marco Conti, Jennifer J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *ELSEVIER Ad Hoc Networks 1*, 2003, pp. 13-64.

[8] Jaime Sanchez, Ralph Martinez, Michael W. Marcellin, "A survey of MAC protocols proposed for Wireless ATM," IEEE Network, vol. 11, no. 6, Nov/Dec 1997, pp. 52-62.

[9] EN 300 652 V1.2.1 (1998-07), ETSI, Broadband Radio Access Network (BRAN); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Funct. Specification.

[10] Qiang Ni, Lamia Romdhani, and Thierry Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN," *Wiley Journal of Wireless Communication and Mobile Computing (JWCMC)*, John Wiley and Sons Ltd., Vol. 4, Issue 5, 2004, pp. 547-566.

[11] Krzysztof Pawlikowski, Hae-Duck Joshua Jeong, and Jong-Suk Ruth Lee, University of Canterbury, "On Credibility of Simulation Studies of Telecommunication Networks", *IEEE Communic. Magazine*, vol.40, no.1, Jan.2002, 132-139.