APSM: Bounding the Downlink Delay for 802.11 Power Save Mode

Xavier Pérez-Costa and Daniel Camps-Mur
NEC Network Laboratories, Kurfürsten-Anlage 36, Heidelberg, Germany.
E-mail: {xavier.perez-costa, daniel.camps-mur}@netlab.nec.de

Abstract— **The popularity of wireless LANs, due to its provision of high speed wireless Internet access at a low cost and the cheap chipset price, has resulted in a strong trend toward the integration of this technology in the upcoming all-in-one mobile devices that could include for instance cellular, wireless LAN and personal digital assistant (PDA) capabilities. Such devices though require of power saving mechanisms in order to guarantee a reasonable battery duration. The 802.11 standard provides a power save mode that reduces the wireless LAN technology power consumption, however, this mode can result in downlink delays (AP to station) unacceptable for the QoS of some applications, e.g., VoIP. To overcome this problem, we propose an adaptive power save mode algorithm (APSM) that adapts the data frames MAC downlink delay of a certain station according to the downlink frame interarrival time observed at the AP MAC layer. We conducted an evaluation of our proposal with respect to downlink delay, power efficiency and signaling load using the OPNET simulator and compared its performance with 802.11 standard power save mode and two different static alternatives. The results show the effectiveness of our algorithm in providing a soft upper bound of the MAC downlink delay while significantly decreasing the power consumption and requiring a signaling load similar to the one of standard power save mode.**

I. INTRODUCTION

Over the last years there have been two main drivers for the evolution of communications systems. On the one hand, the Internet has achieved a great popularity and is the de facto standard for data transmission. On the other hand, the number of cellular technologies users has experienced an enormous growth.

In parallel with the evolution of cellular systems, emerging license-free wireless technologies as Wireless LAN (WLAN), which require minimum initial investment and operational cost, have captured a fast growing market. The success of the Wireless LAN technology in providing a low cost high speed wireless Internet access and its low chipset price is driving a strong trend toward the inclusion of this technology in mobile devices, e.g., personal digital assistants (PDAs) and cellular phones. Most of the mobile devices, however, are limited with respect to the battery capacity due to size restrictions. Considering that the capabilities of laptops are converging with the ones of PDAs and mobile phones, which are of a considerable smaller size, it is foreseen that power managament issues will become even more relevant for such devices.

Because of the intrinsic nature of 802.11, which is based in a shared channel access (CSMA/CA), wireless stations continuosly listen to the channel to determine its current status. As a result, a handheld device, e.g., PDA, connected through an 802.11b wireless LAN, will drain its battery after a few hours as opposed to cellular phones that have a standby battery lifetime up to several days. Ideally, the wireless LAN technology should achieve a battery consumption similar to cellular phones one in order to guarantee its success when being incorporated to the upcoming mobile devices.

The IEEE 802.11 standard provides a power save mode that reduces the time required for a station to listen to the channel. Once every beacon interval, usually 100ms, the access point (AP) sends a beacon indicating whether or not a certain station has any data buffered at the AP. Wireless stations wake up to listen to beacons at a fixed frequency and poll the AP to receive any buffered data by sending power save polls (PS-Polls). Whenever the AP sends data to a station, it indicates whether or not there are more data frames outstanding, using the More Data bit in the data frames, and the station goes to sleep only when it has retrieved all pending data. Although this mechanism significantly alleviates the power consumption problem, a new dependency between the data frames MAC downlink delay (AP to station) and the listen interval is introduced. Consequently, some listen interval values can result in downlink delays unacceptable for certain QoS-sensitive applications, e.g., voice over IP. Further details about the power save mode operation can be found in [1].

Several solutions can be designed to reduce the downlink delay as we will explain in Sections II and III. In this paper we propose a dynamic approach that provides a *soft* upper bound (statistical guarantee) of the MAC downlink data frame delay by adapting to the downlink rate.

To the best of the authors' knowledge previous work on the 802.11 power save mode has focused on problems different than the one studied here. Regarding the infrastucture mode, [2] described an algorithm for reducing the power consumption for Web-like transfers based on the request messages while [3] introduced a secondary low-power radio element in the stations to improve again the energy usage efficiency. On the other hand, [4], [5] analysed the issues that arise when using the 802.11 independent power save mode for ad-hoc networks and proposed solutions to improve its performance. In our current and previous work [6], we analyse the 802.11 infrastructure power save mode from a perspective different than the power efficiency, i.e., from the perspective of the downlink delay introduced for the delivery of data frames, and propose a generic solution in order to adapt the resulting downlink delay to the data frame interarrival time received at the MAC layer of the AP.

II. MECHANISMS TO REDUCE THE DOWNLINK DELAY

The standard power save mode functionality defined by IEEE 802.11 introduces a dependency between the Beacon Interval value (BI) and the data frames downlink delay (AP to station) since the process to retrieve frames from the AP only starts after receiving a beacon indicating that frames are buffered at the AP. This dependency implies that some BI configurations might result in downlink delays above the acceptable for some applications, e.g., VoIP, as we will show in Section IV-A. In order to remove the dependency between the BI and the downlink delay several modifications of the standard Power Save mode can be designed which can be divided in two main different groups: proactive and reactive.

Proactive approaches remove the dependency between the BI and the downlink delay by forcing a station to send PS-Polls at certain time intervals independently of whether a beacon has been received indicating that there is traffic buffered at the AP for that station. By setting a certain PS-Poll send interval below the BI, a *soft* upper bound (statistical guarantee) of the downlink delay can be provided.

Reactive approaches take advantage of the instant when a station awakes to send a data frame for sending a PS-Poll. In this case, the dependency with the BI is also eliminated but a new one with the uplink data frame transmission interval is introduced.

Both approaches present a common problem when an AP receives a PS-Poll and no frame is buffered for the station that sent it. The problem comes from the fact of trying to reduce the downlink delay by generating PS-Polls without being totally sure that a data frame is buffered at the AP. According to the 802.11 standard Power Save mode description, PS-Polls sent by an 802.11 station to an AP can either be acknowledged by receiving the corresponding data frame or by an acknowledgment frame. However, in both cases, if no frame is buffered at the AP for the station, the station will remain awake expecting the reception of the data frame after sending the PS-Poll or after receiving the acknowledgment. As a result, the power saving efficiency will suffer a significant degradation. Note that this could not occur with standard power save mode since a PS-Poll is only sent after the indication that a frame is buffered at the AP.

We designed a simple solution for this problem that consists in using the More Data field included in the acknowledgment frames, which is currently not being used, to indicate to the station whether it should expect a data frame, i.e., remain awake, or not. By doing this, a station that sent a PS-Poll but has no frame buffered at the AP can go to doze mode immediately after receiving the acknowledgment, i.e., reducing in this case the time that a station remains awake to the minimum. In the rest of the document we will refer to an acknowledgment indicating that no frame is buffered for a certain station as a *No Data Acknowledgment (NDAck)*.

The aforementioned proactive and reactive solutions are the first solution that one could think of but, as we will show in Section IV, they present similar problems with respect to the assumption of a certain application traffic pattern. Because of their *static* nature, when the traffic pattern assumed does not match with the one currently generated by the application, either unnecessary PS-Polls are generated or a large downlink delay is experienced.

In order to overcome the above mentioned issues, we propose an adaptive power save mode algorithm (APSM) that, using the information available at the MAC layer, provides a soft upper bound of the MAC downlink delay by adapting to the downlink data frame interarrival time.

III. ADAPTIVE POWER SAVE MODE ALGORITHM (APSM)

The adaptive power save algorithm proposed has been designed to fulfill the following objectives:

1) Provide a soft upper bound of the MAC downlink delay according to common application requirements independently of the beacon interval value.
2) Keep the bound guarantee even in the case of more than one application per station (traffic mixed).
3) Guarantee a power saving efficiency similar or better than the one of standard power save mode.
4) Minimize the signaling load introduced in the channel.
5) Minimize the impact on the standard power save mode.

To achieve these objectives we have opted for an adaptive proactive approach instead of for a reactive one to have a solution independent of any uplink application characteristic or WLAN specific parameter, e.g., BI. The algorithm designed is based on the estimation of the current downlink data frame interarrival time to adapt the PS-Poll interval accordingly and in this way upper bound the MAC downlink delay to the frame arrival interval. We have chosen this approach because in general applications can cope with an end-to-end delay well above their frame generation interval, e.g., a delay sensitive application as VoIP, which codecs usually generate frames every 10-30ms, can deal with an overall delay of 150-300ms. Considering that in most of the networks including WLAN access the main contributor to the overall delay is usually the MAC layer, upper bounding the MAC downlink delay to the downlink frame interarrival time should satisfy the most stringent application requirements.

The APSM algorithm is started by a station after the reception of a beacon indicating that frames are buffered for that station and is stopped when the number of NDAck received in a row reach the n_NDAck_max value. Note that ASPM becomes equivalent to standard power save mode (except from the transmission of n_NDAck_max PS-Polls) when the downlink data frame interarrival time is so low that due to NDAcks we always disable the APSM mechanism before a new data frame arrives. The n_NDAck_max value represents a trade-off between certainty that the communication with a station has stopped and signaling load and power consumption savings.

Once the APSM algorithm is started, a station continuosly sends PS-Polls to the AP at a certain PS-Poll interval ($pspoll\_interval$) starting from the configurable predefined value $pspoll\_interval\_init$ and converging to the downlink data frame interarrival time following the procedure described in Algorithm 1. The $pspoll\_interval\_init$ value should be chosen considering that the algorithm might require several iterations until is finally adapted and that the speed to increase the PS-Poll interval value is faster than the speed to decrease it. During the phase where the APSM algorithm is active, stations still follow the standard procedure of the More Data mechanism when receiving a data frame but do not awake to check the information of the beacon regarding buffered data frames or react to them. The More Data mechanism is still necessary to correct deviations when the PS-Poll interval estimated by the APSM algorithm is not yet adapted.

The information collected at the MAC layer to infer the actual downlink data frame interarrival time is: number of data frames received during a certain period (n_fr_rcvd), data frames received with the More Data bit set to true (MD) and acknowledgments indicating that no frames are buffered at the AP for a certain station ($NDAck$). Based on this information, we have designed the APSM algorithm described in Algorithm 1 which adapts the PS-Poll sending interval to the downlink data frame interarrival time observed at the MAC layer

The reasoning used to design the algorithm is as follows. Ideally, if the PS-Poll interval configured would perfectly match the downlink data frame interarrival time, when a PS-Poll is sent to the AP a *single* data frame would be buffered at the AP waiting for a PS-Poll arrival. When the PS-Poll interval is not yet *adapted* though, what happens is that if the PS-Poll interval is above the downlink data frame interarrival time the station will recover some packets using the More Data mechanism while if the PS-Poll interval is below the downlink data frame interarrival time, then the station will receive some NDAcks. Therefore, our algorithm is based on the usage of the data frames received with the More Data bit set to true (MD) and the reception of no data acknowledgments (NDAcks) as indicators of whether our PS-Poll interval estimation is above the desired value or below, respectively. Additionally, when the decision of modifying the current PS-Poll interval is taken, the required change is calculated taking into account the number of data frames received during the period comprised from the last moment this decision was taken until the current one (n_fr_rcvd).

Since one of our objectives is to minimize the signaling load introduced by the algorithm, the way of deciding when a change in the estimation is required and how fast this change should be is different depending on whether the estimation is above the actual downlink frame interarrival time or below. To achieve our target the PS-Poll interval used should always be equal or above the downlink frame interarrival time to avoid sending unnecessary PS-Polls. Therefore, we have introduced a factor $k \in [1, \infty]$ used to control the difference in the estimation value change and we modify the PS-Poll interval value for each NDAck received while in the MD case we change the estimation if we previously received another one. By introducing this asymmetry we achieve i) a faster adaptation of the algorithm when the estimation is below the desired value and ii) the reduction of the danger of bouncing around the actual value once the estimation is above but close to the desired one. To avoid the problem that could occur due to our slower decreasing speed when a significant decrease of the PS-Poll interval would be necessary to adapt to the current downlink data frame interval, we included a mechanism (n_MD_burst) that detects whether a considerable change has occurred based on the configurable threshold j and reacts accordingly.

Algorithm 1 Adaptive algorithm to bound the downlink delay introduced by the power save mode mechanism to the downlink data frame interarrival time

```
On receiving an ACK frame:
if $NDAck$ then
  if $n\_NDAck == n\_NDAck\_max$ then
    Stop the APSM algorithm
  else
    $pspoll\_interval = pspoll\_interval(1 + \frac{1}{n\_fr\_rcvd+1})$
    $n\_NDAck \leftarrow n\_NDAck + 1$
    $n\_fr\_rcvd \leftarrow 0$
    $update\_next\_MD \leftarrow false$
    $previous\_MD \leftarrow false$
  end if
else
  $n\_NDAck \leftarrow 0$
end if
On receiving a DATA frame:
if $MD$ then
  if $update\_next\_MD$ then
    $pspoll\_interval = pspoll\_interval(1 - \frac{1}{k(n\_fr\_rcvd+1)})$
    cancel pending PS-Poll and schedule a new one
    $update\_next\_MD \leftarrow false$
    $n\_fr\_rcvd \leftarrow 0$
  end if
  if $previous\_MD == false$ then
    $n\_fr\_rcvd \leftarrow 0$
  end if
  $previous\_MD \leftarrow true$
else
  if $previous\_MD$ then
    if $n\_fr\_rcvd > 1$ then
      if $n\_MD\_burst > j$ then
        $pspoll\_interval = \frac{pspoll\_interval}{n\_fr\_rcvd+1}$
        cancel pending PS-Poll and schedule a new one
      else
        $n\_MD\_burst \leftarrow n\_MD\_burst + 1$
        $update\_next\_MD \leftarrow true$
      end if
    else
      $update\_next\_MD \leftarrow true$
      $previous\_MD \leftarrow false$
      $n\_MD\_burst \leftarrow 0$
    end if
  end if
  $n\_MD\_burst \leftarrow 0$
end if
$n\_fr\_rcvd \leftarrow n\_fr\_rcvd + 1$
```

We deliberately restricted the algorithm design to use only MAC layer information in order to keep it transparent to the upper and lower layers easing thus its implementation. The complexity of the mechanism has been kept low by minimizing the parameters to be configured ($pspoll\_interval\_init$, n_NDAck_max, k and j) and by distributing the computation load between the stations instead of centralizing it at the AP. Our aim is to design a generic enhancement of the 802.11 power save mode that covers the case not only of constant bit rate traffic sources but also of variable bit rates and interarrival times and specially that is suited for the case of a mix of traffic for a station. We consider the case of mixed traffic specially relevant since we assume that in the future mobile devices could be used for a voice/video communication while at the same time run an e-mail or web browsing application. Therefore, our algorithm has been designed to absorbe this additional bursty traffic by rapidly reducing the PS-Poll interval for keeping the soft upper bound downlink delay guarantee.

IV. PERFORMANCE EVALUATION & DISCUSSION

In this section we evaluate via simulation the performance of the different enhancement approaches described in Sections II and III as compared to standard 802.11 power save mode. We extended the 802.11b libraries provided by OPNET 10.0 [7] to include the 802.11 standard power save mode and the potential enhancements described in Sections II and III.

For illustration purposes we have configured the static *proactive* mechanism used in our simulations to send a PS-Poll every 30ms and the *reactive* mechanism to generate a PS-Poll after the uplink transmission of each data frame (ratio 1). The configuration chosen for our proposed APSM algorithm is $pspoll\_interval\_init$ of 10 ms, n_NDAck_max equal to 3, k to 2 and j to 1. The implementation of all enhancement approaches include the NDAck mechanism. With respect to the standard power save mode, the listen interval used is 1.

In the following sections, unless otherwise indicated, the number of wireless stations used for the simulations is 5. The value has been chosen large enough to guarantee that in our experiments the effect of collisions in the channel is taken into account but small enough to avoid saturation. Communication occurs always between wireless-wired pairs through the AP. The length of the simulations performed is 120 seconds and the values shown in the graphs have been computed using at least 10 different seeds.

A. Impact of the Beacon Interval

The purpose of the enhancements described in Sections II and III is to remove the dependency of the average MAC downlink data frame delay with respect to the beacon interval value. In this experiment we evaluate the effectiveness of the proposals computing the average data frame downlink delay for a beacon interval value varying from 20 to 100ms.

Figure 1 shows the results for a *symmetric application* sending and receiving every 40ms (dashed line), an *uplink asymmetric application* sending in the uplink (station to AP) every 20ms and receiving every 60ms (dotted line) and a *downlink asymmetric application* sending every 60ms and receiving every 20ms (solid line), all of them generate traffic at a 64kbps rate. The results are as expected, while the downlink delay of data frames increases according to the beacon interval value for standard power save mode, the one of the enhancement approaches remains constant around a value corresponding to half of the PS-Poll generation rate since the arrival time of the data frames to the AP within a PS-Poll interval is uniformly distributed. Note that the standard power save mode downlink delay does not increase linearly according to the beacon interval increase in the symmetric and asymmetric uplink case. The reason for that is the More Data mechanism effect that when it comes into play reduces the medium access delay since the probability of collision is lower than directly after receiving a beacon.

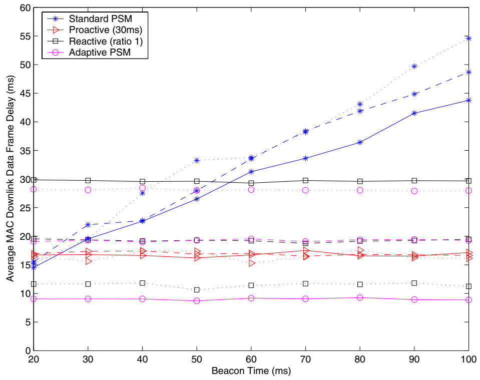


Fig. 1. Impact of the beacon interval on the MAC downlink delay of data frames

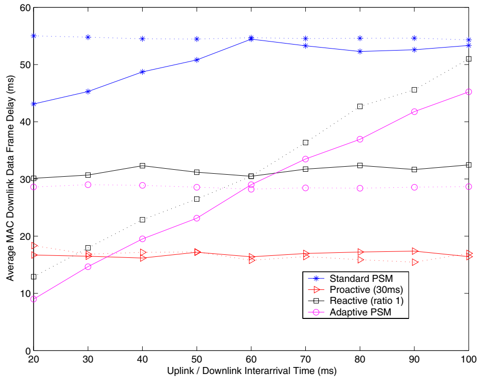


Fig. 2. Impact of the traffic characteristics on the MAC downlink delay

In the rest of the experiments the beacon interval used is 100ms.

B. Dynamic Adaptation of the APSM algorithm

A design goal of our APSM algorithm is to adapt the downlink delay of the data frames according to the characteristics of the downlink traffic generated by the application. In order to accomplish this, each station predicts the downlink data frame interarrival time at the MAC layer to generate PS-Polls at a rate that provides a soft upper bound guarantee for the downlink data frame delay.

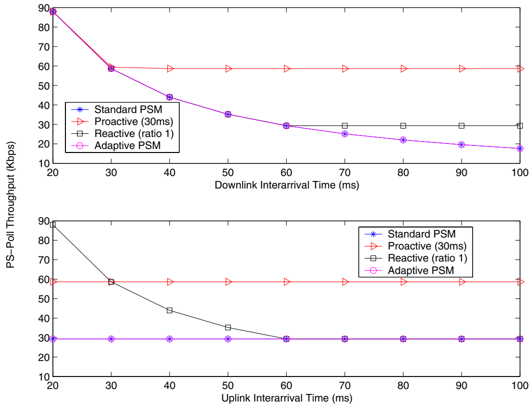


Fig. 3. Impact of the traffic characteristics on the PS-Polls signaling load

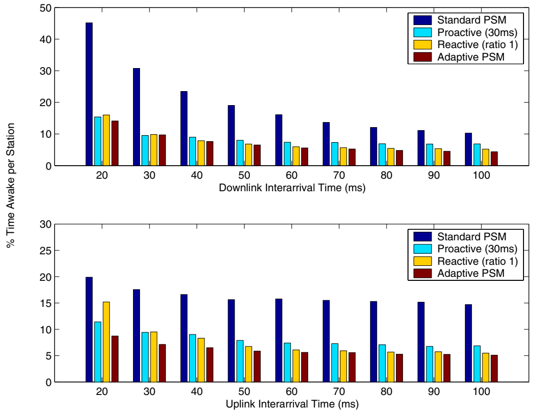
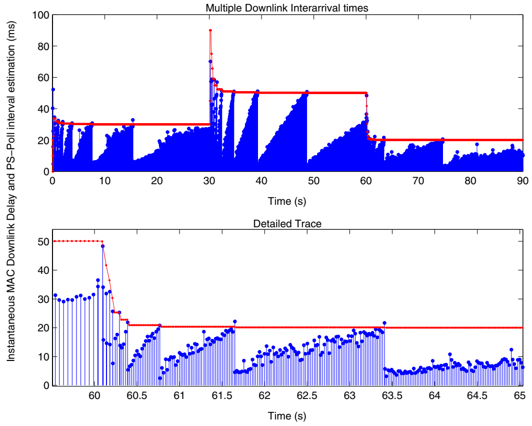


Fig. 4. Impact of traffic characteristics on the power saving efficiency

Figure 2 presents the results regarding the average MAC downlink delay for an experiment where, in the case of the uplink asymmetric application (dotted line), we fix the downlink frame interarrival time to 60ms and increase the uplink from 20 to 100ms, and in the case of the downlink asymmetric application (solid line), we fix the uplink frame interarrival time to 60ms and increase the downlink from 20 to 100ms. The results show that the APSM algorithm achieves the desired objective of upper bounding the downlink data frame delay according to the downlink frame interarrival time. For the uplink asymmetric application case the downlink delay keeps constant around the 30ms value since the downlink frame interarrival time is 60ms and in the downlink asymmetric case the downlink delay increases according to an increasing downlink frame interarrival time.

With respect to the static proactive scheme, the downlink delay is constant for all cases due to the fact that, independently of the frame interarrival time in the downlink, the probability of a frame to arrive within a certain point between two consecutive PS-Polls is uniformly distributed. The same applies to the downlink asymmetric case of the reactive scheme with the only difference that now the PS-Poll generation rate is 60ms instead of 30ms. On the other hand, the downlink delay increases according to the uplink frame interarrival time for the reactive mechanism because the PS-Poll interval increases in the same way. Regarding the standard power save mode performance, in the asymmetric uplink case the average delay is constant since a change in the uplink rate does not influence the mechanism. However, in the asymmetric downlink case, the average delay decreases for a downlink rate of 60 to 20 ms due to the fact that occasionally the More Data phase lasts longer than the arrival of new frames.


Multiple Downlink Interarrival times

Detailed Trace

Fig. 5. Dynamic adaptation of the APSM algorithm to the traffic characteristics

In Section III we noted that another objective of our algorithm was to minimize the signaling load introduced in the wireless channel. Using the same experiment already described we analyze the PS-Poll signaling load introduced by our algorithm and present the results in Figure 3. Obviously, the ideal would be to achieve with our APSM algorithm, which reduces the downlink delay, the same signaling load results than standard power save mode which does not send any unnecessary PS-Polls. In the figure we see that for both cases, asymmetric uplink and downlink applications, APSM introduces a PS-Poll signaling load almost perfectly matching the one of standard power save mode. We can conclude then that the additional signaling load introduced by the APSM algorithm during the adaptation phase as compared to standard PSM is negligible while it results in a significant performance improvement in the downlink delay.

In the case of the proactive approach, the signaling load introduced is always a constant value except for the case of the asymmetric downlink application when it generates frames at 20 ms since some additional PS-Polls are sent due to the More Data mechanism. The reactive mechanism shows a very similar behaviour to the standard power save mode or APSM until an uplink/downlink interarrival time of 60ms because for both kind of applications the same PS-Poll rate is generated either due to the More Data mechanism that sends the additional PS-Polls required in the asymmetric downlink application case or due to the unnecessary PS-Polls sent in the asymmetric uplink application case. From the 60ms interarrival time point on the signaling load keeps constant in the asymmetric uplink case due to the More Data mechanism.

Even though the main aim of the APSM algorithm designed is to reduce the downlink delay introduced by the standard power save mode to a value acceptable for the QoS required by the applications, it is equally important to guarantee that our APSM algorithm does not degradate the power saving efficiency as compared to standard power save mode. Figure 4 represents the percentage of time that a station is awake in average for the same experiment previously described. The results show that APSM is always the best performing power saving mechanism. Standard power save mode is always the worst one since the transmissions of the PS-Polls of the stations are always concentrated around the time that follows the reception of the beacon so the probability of collision is higher than for the enhancement approaches that do not introduce this synchronization. The proactive and reactive mechanisms present a power saving efficiency closer or further to the one achieved by APSM depending on whether for a particular case the assumption of the downlink frame rate is better or worse.

Two important characteristics of an adaptive algorithm which are related with each other are the speed at which the algorithm adapts to changes and the stability once it is adapted. This trade-off has been studied by modifying during a simulation the downlink frame interarrival time from 30ms to 50ms and then to 20ms. In Figure 5 we can observe the dynamic adaptation during the simulation time of the PS-Poll interval estimation (solid line) and the instantaneous MAC downlink delay of the data frames (deltas) for one of the stations. In the detailed part of the graph, each point in the solid line represents the transmission of a PS-Poll and each delta the arrival of a data frame in the downlink direction with an amplitude corresponding to the MAC downlink delay experienced. As it can be observed, the APSM algorithm properly estimates the corresponding downlink frame interarrival time and adapts significantly fast to the rate modifications. The 'ramp up' behaviour occurs due to the algorithm design that prioritizes to be above the actual downlink frame rate to minimize the unnecessary PS-Polls sent by the station resulting in ramps that decrease their slope based on the data frames received with the More Data bit set.

C. Impact of a Realistic Scenario

In the previous section we have analyzed the performance of the APSM algorithm considering applications generating traffic at fixed time intervals and at a fixed bit rate in order to facilitate the interpretation of the results obtained. Once the expected behaviour of the algorithm has been validated, we focus in a more general scenario where realistic applications do not generate traffic constantly, e.g., G.711 audio codec with silence suppression (data rate 64kbps, frame rate 30ms), or do not generate traffic at a constant bit rate, e.g., MPEG-4 streaming of the movie Jurassic Park (average rate 150kbps, frames generated every 40ms [8]).

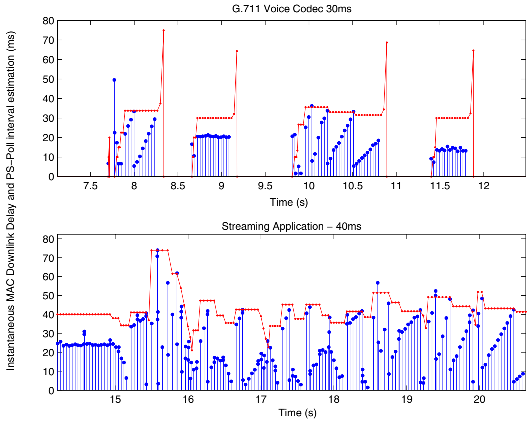


Fig. 6. Dynamic adaptation of the APSM algorithm to the downlink frame interarrival time of real applications

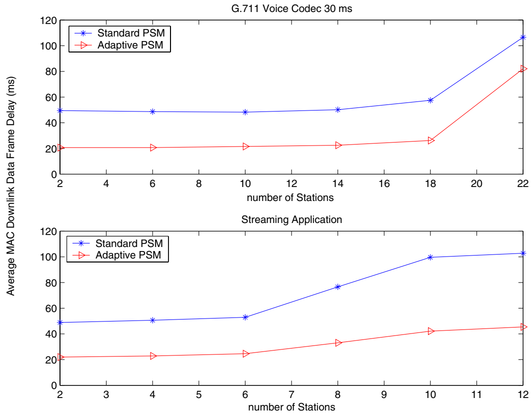


Fig. 7. Impact of congestion in the channel on the MAC downlink delay

Figure 6 shows the proper adaptation of the PS-Poll rate estimated by the APSM algorithm during the active periods of the VoIP source (solid line where each point represents a PS-Poll transmission) and the detection of the silence periods going back to doze mode. As a result, the soft upper bound downlink delay guarantee is provided (amplitude of deltas).

The adaptation to the variable data rate generated by the streaming application is depicted also in Figure 6. In this case, the variable bit rate of the application results in a variable downlink frame interarrival time, e.g., due to segmentation of large packets. The PS-Poll rate estimation successfully adapts to these changes while keeping around the expected average value of 40ms.

Regarding the effect of channel congestion over our algorithm, we study the mechanism robustness by increasing the number of wireless and corresponding wired stations in our scenario. In Figure 7 and 8 it can be appreciated that the APSM algorithm manages to improve the downlink delay and decrease the power consumption even in congestion conditions.

D. Impact of Traffic Mix

One of the main design objectives for our algorithm, as described in Section III, is to keep the delay bound guarantee even in the case of more than one application per station

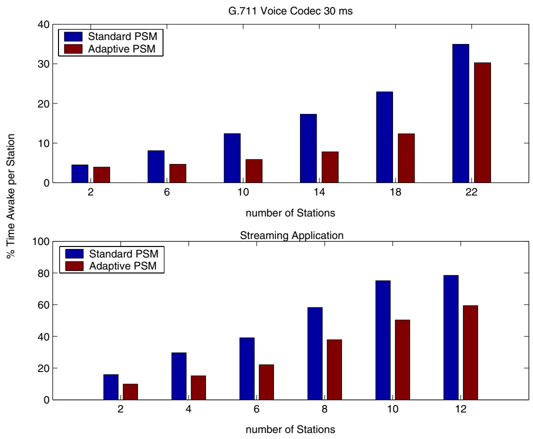


Fig. 8. Impact of congestion in the channel on the power saving efficiency

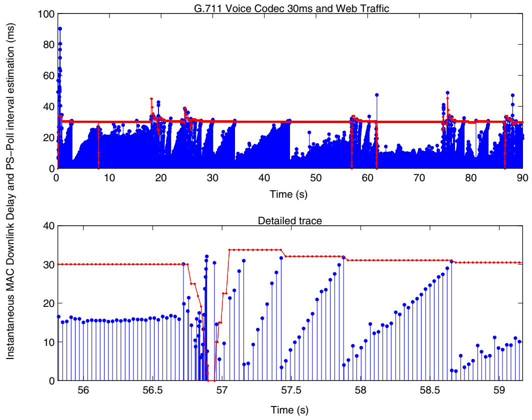


Fig. 9. Dynamic adaptation of the APSM algorithm to the resulting downlink frame interarrival time produced by a voice and web application

(traffic mixed). In this experiment we analyze the effect over the APSM algorithm of adding Web traffic to the VoIP application used in the previous experiment for the case of 5 wireless stations. In this case we do not use silence suppression to facilitate the recognition of the web traffic burst over the voice traffic. The configuration used to emulate web traffic is page interarrival time exponentially distributed with mean 10s and page size 10 KB plus 5 images of a size uniformly distributed between 0.5 and 2 KB.

As shown in Figure 9, the PS-Poll interval estimation of the algorithm (solid line) reacts as desired to the additional load decreasing the PS-Poll interval and therefore keeping the soft upper bound downlink delay guarantee of data frames (amplitude of deltas).

V. SUMMARY & CONCLUSIONS

In this paper, we propose an adaptive power save mode algorithm that provides a soft upper bound of the data frames downlink delay (statistical guarantee) according to the downlink frame interarrival time. The decision for taking this approach is based in the observation that in general applications can cope with an end-to-end delay well above their frame generation rate. For instance, a delay sensitive application as VoIP, which codecs usually generate frames every 10-30ms, can deal with an overall delay of 150-300ms. Since in most of the networks including WLAN access the main contributor to the overall delay is usually the MAC layer, upper bounding the MAC downlink delay to the downlink frame interarrival time should satisfy the most stringent application requirements.

The power saving mechanism designed minimizes the impact over the standard power save mode to facilitate its implementation by i) re-using some of its mechanisms, e.g., beacon indication for starting the APSM algorithm or the More Data mechanism to correct disadjustments, ii) introducing a prediction algorithm of the PS-Poll rate which requires very simple operations and iii) using only MAC layer information.

The performance of our proposal has been analyzed and compared to 802.11 standard power save mode, a static proactive mechanism and a reactive one with respect to the resulting data frame MAC downlink delay, power saving efficiency and required signaling load. The study included the validation of the proper behaviour of APSM under congestion conditions, when used with realistic traffic sources as VoIP with silence suppression or streaming and also when used with traffic mixed from different applications for the same station. The results show that i) our algorithm fulfills its objective of providing an upper bound of the downlink delay according to the downlink frame interarrival time ii) the additional signaling load required in comparison to standard power save mode is negligible and iii) the power saving efficiency is increased because of the desynchronization of the PS-Polls transmissions.

VI. ACKNOWLEDGMENTS

The authors would like to thank their colleagues from the development team of the 3G/WLAN NEC N900iL mobile terminal for inspiring the direction of this research work.

REFERENCES

[1] IEEE, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Standard 802.11, June 1999.
[2] R.Krashinsky and H.Balakrishnan, “Minimizing energy for wireless web access with bounded slowdown,” in *Proceedings of the eighth Annual International Conference on Mobile Computing and Networking (MOBICOM)*, september 2002.
[3] E.Shih, P.Bahl, and M.J.Sinclair, “Wake on wireless: An event driven energy saving strategy for battery operated devices,” in *Proceedings of the eighth Annual International Conference on Mobile Computing and Networking (MOBICOM)*, september 2002.
[4] C.-S. H. Y.-C. Tseng and T.-Y. Hsieh, “Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks,” in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002.
[5] R.Zheng and R.Kravets, “On-demand power management for ad hoc networks,” in *Proceedings of IEEE INFOCOM*, March 2003.
[6] X.Pérez-Costa, D.Camps-Mur, and T.Sashihara, “Analysis of the Integration of IEEE 802.11e Capabilities in Battery Limited Mobile Devices,” IEEE Wireless Communications Magazine, to be published.
[7] “OPNET Simulator,” http://www.opnet.com.
[8] F. Fitzek and M.Reisslein, “MPEG-4 and H.263 Video Traces for Network Performance Evaluation,” IEEE Network,Vol. 15,No. 6,pages 40-54, November/December 2001.