Power Save Adaptation Algorithm for Multimedia Streaming to Mobile Devices

Janet Adams
Performance Engineering Laboratory
School of Electronic Engineering
RINCE, Dublin City University
Ireland
Email: adamsj@eeng.dcu.ie

Gabriel-Miro Muntean
Performance Engineering Laboratory
School of Electronic Engineering
RINCE, Dublin City University
Ireland
Email: munteang@eeng.dcu.ie

***Abstract*— Batteries have not followed the exponential technological improvements of other mobile device related hardware, such as CPU, memory and wireless networking. Battery power often introduces significant limitations to the use of mobile devices and their applications, including those involved in multimedia streaming that have significant high power requirements. This paper proposes a power save adaptation algorithm for mobile multimedia streaming that aims to increase streaming time given limited battery power resources. The multimedia streaming process is divided into three stages: data reception, decoding and playing and power saving solutions for each of these stages are proposed. These power save mechanisms are then combined to give the power save algorithm. Preliminary tests results show that significant increases in battery lifetime have been achieved when the power save mechanisms proposed in this paper are used.**

I. Introduction

There has been significant development in areas of both portable devices and wireless networks in recent years. It is now considered reasonable to support multimedia streaming applications on mobile devices via wireless networks [1]. However, development has concentrated on various pieces of hardware and software and has, to a large extent, neglected power. For example, although memory, CPU and network bandwidth resources have increased exponentially in recent years, batteries have fallen behind in terms of development, improving by only about 2% per year over the last fifty years [2]. Therefore, there is a need to improve battery life span in order to keep up with the rising curve of application-based processing, device complexity and wireless networking capabilities.

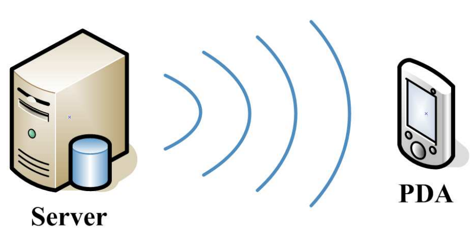

Fig. 1. Typical Multimedia Streaming Architecture

Our research proposes an algorithm that adapts the multimedia streaming process based on the existing battery life/power level. The algorithm's goal is to enable the streaming process to last longer by employing modifications in streaming-related parameters that have significant implications in terms of reduction in battery power consumption. The typical multimedia streaming architecture considered in this paper is shown in Figure 1. The proposed algorithm relies on the idea that the multimedia streaming process can be divided into three stages: reception, decoding and playing. The proposed algorithm makes changes in each of these stages in order to give the highest overall power saving effect.

As the reception stage has the biggest impact on the battery, it will be here that the biggest effort to save power will be made. For this stage, a novel approach called the Adaptive-Buffer Power Save Mechanism was proposed. Results show that significant savings can be achieved when this mechanism is used.

This paper describes the stages of multimedia streaming process when streaming to a portable device and presents comparison-based results which show how battery power consumption is dependent on some stage-specific multimedia streaming-related parameters. Power save mechanisms for each stage are then proposed, including the Adaptive-Buffer Power Save Mechanism. The total power save algorithm is then described in detail. The paper finishes with conclusions and future work directions.

II. Related Works

Many researchers have studied power conservation in mobile devices as a response to the slow improvement in batteries. This development is also needed in order to support the increasing power requirements of the latest devices, as well as the increasing complexity of the tasks to be carried out. Without any improvement to batteries needed to support them and their utilization, the complexity of the streaming tasks will bypass the ability of the batteries to such an extent that the tasks will not be able to be performed.

A. Reception Stage

In relation to the reception stage, Chandra and Vahdat [3] propose an application-specific server side traffic shaping mechanism that can offer energy savings by allowing the client to sleep for longer periods of time. Although this scheme looks promising, it is not compatible with the 802.11 standard as it involves a revised power-save mechanism and seems to ignore the beacon interval which is the basis of the standard power-saver mechanism. Another scheme which is proposed for power saving in the reception stage is presented by Bae et al [4]. The authors describe a Buffer-based Energy efficient CPU scheduler for mobile devices, particularly those that run real time multimedia applications. Although good results are shown to be achieved in simulations, this scheme relies on a low power hardware technique, such as Dynamic Voltage Scaling and on the pre-buffering method, which makes it difficult to be implemented on a real system or device.

B. Decoding Stage

One of the solutions for power saving in the decoding stage of the multimedia streaming process is presented by Pakdeepaiboonpol and Kittitornkun in [5]. The authors describe two high level power-saving techniques which are based on reducing the number of memory/bus accesses by high level language optimization. These solutions are aimed specifically at ARM-based devices which limits their applicability. The high level nature of this solution makes it problematic to carry out adaptively and application specifically. Lu et al [6] propose to reduce the decode power of multimedia by using feedback control. The advantage of this scheme is that no pre-playback or server-side profiling is required and that slack can be reclaimed across frame boundaries.

C. Playing Stage

There has been little research performed for energy saving in the playing stage of the multimedia streaming process. What has been carried out is primarily related to the display, particularly the back light. Pascrich et al [7] propose an adaptive middle ware based approach to optimize back light power consumption for mobile handheld devices when streaming MPEG-1 video. Another back light power management scheme is proposed by Shim, Chang and Pedram [8]. In this case, a back light power management framework for color TFT LCD panels is proposed. The authors propose a scheme called Extended DLS or EDLS, compensates for loss of brightness when there is a rich or moderated power budget and compensates for loss of contrast when the power budget is low. Both of the above solutions are viable.

D. General Power Saving in Multimedia Streaming

There has also been some research performed in the area of general power saving in multimedia streaming. Acquaviva, Benini and Ricco [9] propose a software-controlled approach for adaptively minimising energy in embedded systems for real time multimedia processing. This is a very low level solution involving hardware related solutions and it is not independent of the platform. Korhonen and Wang [10] study the impact of the burst length and peak transmission rate for observed packet loss and delay characteristics. Anastasi et al [11] address energy saving by including periodic transmission interruptions in the schedule of audio frames at the server. Mohapatra et al [12] propose an integrated power management approach that unifies low level architectural optimizations, OS power saving mechanisms and adaptive middle ware techniques. Zhu and Cao [1] present a power conserving service model for streaming applications over wireless networks. Although both of these schemes achieve power savings, neither are relevant to the application based adaptive scheme proposed in this paper.

III. Stages of the Multimedia Streaming Process

As can be seen from Figure 1, the typical architecture over which multimedia streaming is performed includes a server, a mobile client and a wireless network over which the multimedia streaming is performed. This process of streaming multimedia content to a mobile device can be divided into three stages: reception, decoding and playing.

A. Reception Stage

The reception stage involves all network related tasks in the multimedia streaming process. This includes the media being sent across the network from the server as well as the receiving of the streamed multimedia data by the client. Due to the fact that the Network Interface Card (NIC) is responsible for a large amount of the power consumed in any mobile device [3] [10], this is the first aspect this paper will discuss. Tests have been performed which show that when the media content is streamed instead of played, hence when the reception stage is introduced, there is a significant decrease in the battery lifetime of the device. For MPEG 1 Layer 3 encoded audio, there is a 50% increase in battery life when the media is played instead of streamed, as shown in Figure 2. For MPEG 4 encoded video, there is a 66% increase in battery life when the media is played instead of streamed. The results of this test are shown in Figure 3.

B. Decoding Stage

The second stage of the multimedia streaming process involves decoding received data. The decoding process will depend on the encoding scheme that was used by the server. To investigate the effect this stage has on the battery, tests were carried out comparing a media clip encoded at different bit rates. In order not to account for the influence of the reception-related factors, the media was played in these tests. The results, shown in Figure 4, indicate that the battery life increased by 100% when the average content bit rate was reduced from 192 kbps to 128 kbps.

C. Playing Stage

The final stage involves the decoded multimedia content being (dis)played to the user. This involves speakers and a screen, or sometimes both, depending on the type of media clip used. To investigate the effect that this stage has on the battery,

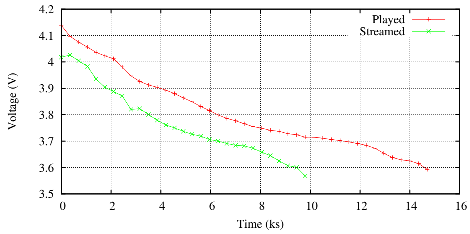

Fig. 2. Reception Stage: MPEG 1 Layer 3 - Play versus Stream

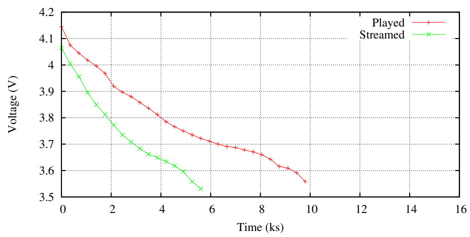

Fig. 3. Reception Stage: MPEG 4 - Play versus Stream

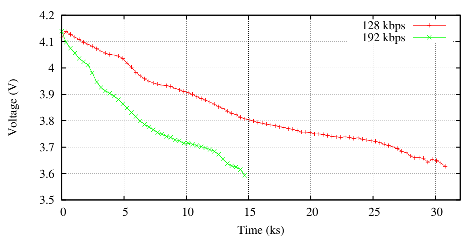

Fig. 4. Decoding Stage: MPEG 1 Layer 3 - Compare Different Bit Rates

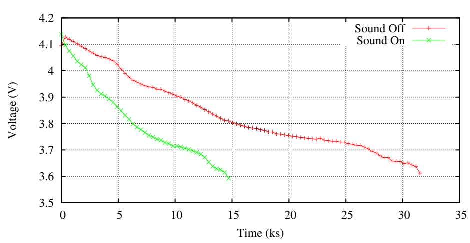

Fig. 5. Playing Stage: MPEG 4 - Compare Different Volume Levels

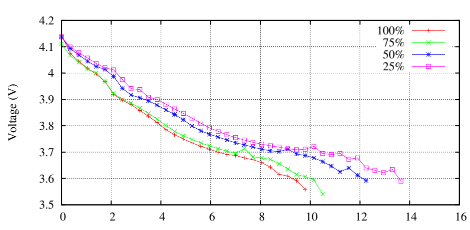

Fig. 6. Playing Stage: MPEG 4 - Compare Different Brightness Levels

it was necessary to testing the effect of both the speakers and the screen. In looking at the effect of the speakers, two tests were carried out using an MPEG 1 Layer 3 audio clip. In one of the tests, the volume was at an average level, in the second the volume was muted. The results, presented in Figure 5, show that when the volume was muted, the battery life increased by over 100%. To investigate the effect that the screen has on the battery, tests using an MPEG 4 encoded video clip, were carried out with varying levels of screen brightness. The brightness was set to 100%, 75%, 50% and 25%, each of the reductions affected the battery by increasing its lifetime. An increase of approximately 40% was achieved when the screen brightness was set to 25% in comparison with the situation when maximum brightness was used for the display. The results of these tests are shown in Figure 6.

The MPEG 1 Layer 3 audio clips used were encoded with an average bit rate 192 kbps and a sampling frequency of 44 kHz, except were otherwise indicated. The MPEG 4 encoded clips had a resolution of 352x288 pixels and frame rate of 25 fps. The client used was a Dell Axim PDA with an IEEE 802.11b wireless network card. The server was a 1.86 GHz PC running Windows XP. Other details related to these tests can be found in [13].

IV. Stage-Related Power Saving Solutions

In order to achieve the highest overall power savings, power save mechanisms for each of the three stages already described are proposed. As the reception stage is the biggest drainer of the battery power [3] [10], we will focus more on this stage than the other two.

A. Power Saving in the Reception Stage

To save power in the reception stage, the Adaptive-Buffer Power Save Mechanism (AB-PSM) is proposed. This is a mechanism that allows traffic to be hidden temporarily from a mobile station, allowing it to sleep for longer, achieving power savings. The idea behind this scheme is that the multimedia data is sent in larger amounts at less frequent intervals, resulting in the same average bit rate as if AB-PSM is not employed. One of the main advantages of AB-PSM is that it requires no changes to be made to the existing IEEE 802.11 standard. IEEE 802.11 defines a buffer at the access point which is used to store traffic for sleeping stations. At regular intervals, known as the beacon interval and usually set to 100 ms, every station wakes up and checks if there is traffic for it and if so, receives it. If there is no traffic for the station, it can return to the low power sleep mode. In AB-PSM an additional buffer, known as the Application Buffer, is introduced. This buffer effectively hides the packets so that when the station wakes up at the beacon interval, it sees no waiting traffic and returns to sleep. Once the beacon interval has passed and the station has returned to sleep, the Application Buffer sends the packets to the original Access Point buffer so that at the next beacon interval, the station will receive them. The Application Buffer can hold the packets for more than one beacon interval if required, however there is a threshold above which the delay caused will be significant. This threshold will never be crossed in order to avoid additional delay to the streamed multimedia data. Preliminary tests that have been performed show the effectiveness of the proposed AB-PSM. Comparisons were made between AB-PSM and the legacy PSM in IEEE 802.11, as well as with IEEE 802.11 with no power save mechanism employed. Two variations of AB-PSM were tested, however it should be noted that in the four tests, the amount of data sent remained constant. The results are shown in Table I. From these results, it can be seen that between the legacy PSM and AB-PSM, significant savings of up to four hundred minutes were achieved.

B. Power Saving in the Decoding Stage

Using the results in section III-B, power saving in the decoding stage is achieved by sending multimedia content which has been encoded at lower bit rates. In spite of the bit rate being reduced, it is always maintained at a level that gives adequate user perceived quality while saving as much battery as possible.

C. Power Saving in the Playing Stage

As the results presented in section III-C show, there are two possible power saving solutions in the playing stage. The first involves adjusting the screen brightness and the second requires volume adaptations. Both of these are only adjusted to such a level that a fair user quality is maintained.

V. Power Save Adaptive Algorithm for Mobile Multimedia Streaming

The power save methods described above are incorporated into an adaptive power save algorithm for mobile multimedia streaming. The algorithm considers a client-server architecture, as presented in Figure 1. When the client requests data from the server, the server streams the requested content across the wireless network. Initially, the content is streamed with no power save mechanisms applied, other than the legacy IEEE 802.11 PSM. The client receives the content and compares an estimation of the remaining battery life to the time duration it would require to complete the streaming task. If the battery power level does not enable the mobile device to complete the task, then the client will notify the server that power save mechanisms need to be implemented.

The power save mechanisms will be performed incrementally. The first power save mechanisms employed will be the ones related to the reception stage, as it drains the most power from battery and significant savings can be obtained with very little influence on end-user perceived quality. The next will be decoding stage related power save mechanisms, then the playing stage ones. The different proposed power save mechanisms are shown in Table II. They are divided by stage

TABLE I
Effectiveness of AB-PSM

	Packet Size (Bytes)	Interval (ms)	Battery Life (minutes)	
802.11	**No PSM**	25	512	168
	Legacy PSM	25	512	250
802.11 + Proposed Idea	**AB-PSM**	100	2048	606
	AB-PSM	200	4096	643

TABLE II
Scaling of Power Save Mechanisms

Stage	Power Save Mechanism	State
Reception	Legacy Power Save	0
	AB-PSM Level 1	1
	AB-PSM Level 2	2
Decoding	Max Bit Rate	0
	Average Bitrate	1
	Low Bitrate	2
Playing	Max Brightness/Volume	0
	Average Brightness/Volume	1
	Low Brightness/Volume	2

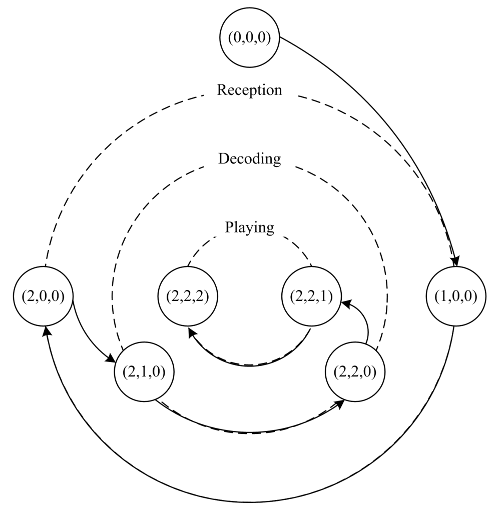

Fig. 7. State Diagram

with each stage having three different states. Each time a power save mechanism is implemented, the client will wait a small amount of time and performs again the comparison between the remaining battery life and the battery life required to complete the task. If the task is not completed, the next power save mechanism is implemented, otherwise no further power save mechanisms will be employed.

According to our proposed algorithm, throughout the battery lifetime, the system will always be in some state. Each of the stages has three different power save states associated with it. Different indexes are used for each stage as follows: the reception stage uses i, the decoding stage uses j and the playing stage uses k. Therefore, the current power save state of the device can always be represented by the triplet $\{i, j, k\}$, where i, j and k will have a value of 0, 1 or 2 depending on the current state. The different states associated with each stage are shown in Table II. The state diagram for the system that deploys the total power save adaptive algorithm is shown in Figure 7.

The system architecture for the deployment of the proposed total power save adaptive algorithm is shown in Figure 8. On the server side there is an application buffer for use with AB-PSM. There is also scalable media storage, which allows for adjusting the encoding bit rate if necessary. On the client side there is the reception, decoding and (dis)playing units and also

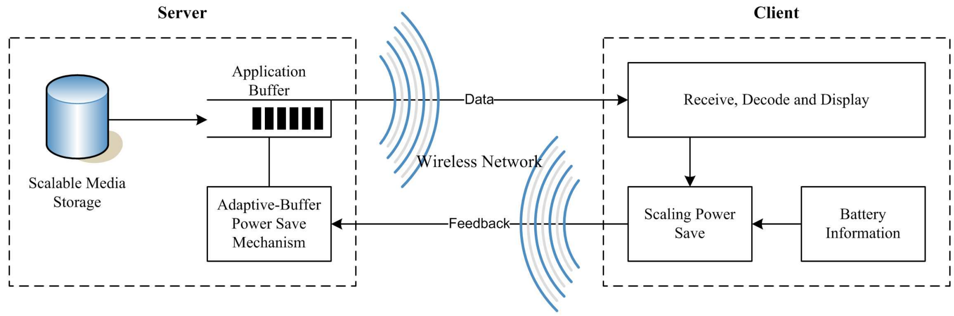

Fig. 8. Total Power Save Algorithm

the battery information module which estimates the remaining battery life. The scaling power save module suggests which power save mechanisms is implemented next. If the server is required to perform power save-based adaptations, either in terms of multimedia content encoding-related tasks or in terms of implementation of a form of AB-PSM, the scaling power save module uses feedback to inform the server.

User preference can be accommodated by allowing the user to choose which power save mechanisms they would or would not like to implement. The user can also choose a level at which they want the power save to stop. For example, if a user specifies that they do not want the power save mechanisms to affect the playing stage, the algorithm will never pass state $\{2, 2, 0\}$.

Conclusion

This paper proposed a power save adaptive algorithm for mobile multimedia streaming. The proposed algorithm divides the multimedia streaming process into three stages and has power save mechanisms for each of the three stages. These stages are reception, decoding and playing, of which the reception stage drains the most power from the battery.

To save power in the reception stage, the Adaptive-Buffer Power Save Mechanism (AB-PSM) is proposed that takes advantage of the IEEE 802.11 standard rather than changing it. Significant increases in the battery lifetime were achieved when AB-PSM was used. To save power in the decoding stage, the multimedia content was encoded at a lower bit rate, and in order to save power in the playing stage, adjustments were made to the screen brightness and to the volume.

Future Work

Future work will involve testing the proposed power save adaptive algorithm on different device types and different battery types. The tests will also involve more than one client and incorporate background traffic. Subjective testing will be performed with a large number of users in order to examine the effect on end user quality when each of the power save mechanisms are used. The results will be used to determine the optimal power save mechanisms application order to maximize end user perceived quality during the multimedia streaming process.

Acknowledgment

The support of the Irish Research Council for Science Engineering and Technology (IRCSET) Embark Initiative and the Enterprise Ireland Commercialization Fund is gratefully acknowledged.

References

[1] H. Zhu and G. Cao, "On supporting power-efficient streaming applications in wireless environments," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 391 – 403, July 2005.

[2] T. Lan and A. Tewfik, "A resource management strategy in wireless multimedia communications-total power saving in mobile terminals with a guaranteed QoS," *IEEE Transactions on Multimedia*, vol. 5, pp. 267 – 81, June 2003.

[3] S. Chandra and A. Vahdat, "Application-specific network management for energy-aware streaming of popular multimedia formats," *Proceedings of the General Track. 2002 USENIX Annual Technical Conference*, pp. 329 – 42, 2002.

[4] G. Bae, J. Kim, D. Kim, and D. Park, "Low-power multimedia scheduling using output pre-buffering," *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 389 – 96, 2005.

[5] P. Pakdeepaiboonpol and S. Kittitornkun, "Energy optimization for mobile MPEG-4 video decoder," pp. 1–6, November 2005.

[6] Z. Lu, J. Lach, M. Stan, and K. Skadron, "Reducing multimedia decode power using feedback control," *Proceedings 21st International Conference on Computer Design*, pp. 489 – 96, 2003.

[7] S. Pasricha, S. Mohapatra, M. Luthra, N. Dutt, and N. Venkatasubramanian, "Reducing backlight power consumption for streaming video applications on mobile handheld devices," 2003.

[8] H. Shim, N. Chang, and M. Pedram, "A backlight power management framework for battery-operated multimedia systems," *IEEE Design and Test of Computers*, vol. 21, no. 5, pp. 388 – 396, 2004.

[9] A. Acquaviva, L. Benini, and B. Ricco, "Software-controlled processor speed setting for low-power streaming multimedia," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, pp. 1283 – 92, November 2001.

[10] J. Korhonen and Y. Wang, "Power-efficient streaming for mobile terminals," *Proceedings of the 15th International Workshop on Network and Operating Systems Support for Digital Audio and Video. NOSSDAV 2005*, pp. 39 – 44, 2005.

[11] G. Anastasi, A. Passarella, M. Conti, E. Gregori, and L. Pelusi, "A power-aware multimedia streaming protocol for mobile users," *Proceedings. International Conference on Pervasive Services 2005*, pp. 371 – 80, 2005.

[12] S. Mohapatra, R. Cornea, N. Dutt, A. Nicolau, and N. Venkatasubramanian, "Integrated power management for video streaming to mobile handheld devices," pp. 582–591, 2003.

[13] J. Adams and G. M. Muntean, "Power-dependent adaptation algorithm for mobile multimedia networking," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, April 2006.