Energy Management for Interactive Applications in Mobile Handheld Systems

Jung-hi Min and Hojung Cha
Dept. of Computer Science,
Yonsei University
Seoul, 120-749, Korea
{jhmin,hjcha}@cs.yonsei.ac.kr

Jongho Nang
Dept. of Computer Science,
Sogang University
Seoul, 121-742, Korea
jhnang@sogang.ac.kr

ABSTRACT
The usage of interactive applications increases in handheld systems. In this paper, we describe a system-level dynamic power management scheme that considers interaction between the CPU and the WNIC, and interactive applications to reduce the energy consumption of handheld systems. Previous research efforts considered the CPU and the WNIC separately to reduce energy consumption. The proposed scheme reduces the energy consumption of handheld systems by using the information gathered from the WNIC to control the CPU voltage and frequency when interactive applications are executed. Experimental results show that on average the proposed scheme reduces energy consumption by 46% when compared to DVFS (Dynamic Voltage and Frequency Scaling) for the CPU and DPM (Dynamic Power Management) for the WNIC.

Categories and Subject Descriptors
C.0 [**Special-Purpose and Application-based Systems**]: Real-time and embedded system*s*

General Terms
Management

Keywords
Energy management, WNIC, DVS, DPM

1. INTRODUCTION
Modern handheld systems run complex applications such as multimedia applications and web applications using wireless networks. In particular, the usage of web applications on mobile devices has increased in the energy consumption. User activity influences interactive applications, which can be represented by web applications, because they work according to user request. It is more complicated to formulate power management policy for interactive applications than for BE (Best Effort) applications or CBR (Constant Bit Rate) applications [1, 2, 3]. Reducing the energy consumption of the overall system is an important goal. The CPU, display, and WNIC (Wireless Network Interface Card) are main consumers of power in handheld systems. In this paper, we consider mainly the CPU and the WNIC for efficient system power management.

Previous research efforts on interactive applications have considered the CPU and the WNIC separately to reduce energy consumption. They have used the DVFS (Dynamic Voltage and Frequency Scaling) technique for reducing the energy consumption of the CPU [4-8] and the DPM (Dynamic Power Management) technique for reducing the energy consumption of the WNIC, based on the assumption that these two techniques are orthogonal [9–13]. They also combined these techniques to make power management policy for overall system [3, 14, 15].

In this paper, we propose a system-level power management method that reduces the energy consumption of handheld systems by using characteristics of interactive applications with control of voltage for the CPU and of the power mode for the WNIC. We collected information regarding bandwidth, listening interval, elapsed time, request interval, packet size, and burst level from the WNIC, and network traffic during the execution of interactive applications. The proposed scheme has the ability to find appropriate voltage for the CPU and the power mode, and the listening interval for the WNIC according to the gathered information.

The rest of this paper is organized into three sections. Section 2 describes the proposed scheme. Section 3 discusses the experimental results. Section 4 concludes the paper.

2. POWER MANAGEMENT FOR INTERACTIVE APPLICATIONS
Interactive applications such as web applications represent one type of application used in modern handheld systems. Conventional approaches to reduce the energy consumption of handheld systems executing interactive applications use separate energy-saving techniques for the CPU and the WNIC [16, 17]. Not much research on interaction between the CPU and the WNIC, and system-level power management for interactive applications has been done. To make an efficient system-level power management scheme, we have observed the energy consumption of the CPU and the WNIC while running interactive applications with various voltages for the CPU during CAM (Constantly Awake Mode) mode and PSM (Power Saving Mode) mode, and with various listening intervals for the WNIC. We used

Intel XScale (PXA270) for the CPU, CISCO AIRONET 350 for the WNIC, and Agilent 34970A for the measuring instrument. The clock frequencies used for the CPU are: 104MHz (1.11V), 208MHz (1.24V), 312MHz (1.37V), and 416MHz (1.52V). The listening intervals used for the WNIC are: 100ms, 200ms, 400ms, and 800ms. We measured the energy consumption of the CPU and the WNIC on Mainstone II (Bulverde PXA 270). The collected information—bandwidth, listening interval, elapsed time, request interval, packet size, burst level from the WNIC, and analysis of network traffic— was used to classify network traffic into interactive applications [1, 18, 19]. The representative interactive applications are telnet and web applications.

The proposed scheme consists of two steps: classification of interactive application and mode and listening interval setting. For the first step, the information of the WNIC and network traffic at kernel level is analyzed. The data is collected to determine bandwidth usage ratio, burst level, and efficient listening interval. The following equations are used for analysis of network traffic and classification as interactive applications. Equation (1) is used to calculate bandwidth usage ratio. Each type of application shows a different bandwidth usage ratio. For example, FTP (BE application) shows a high utilization of bandwidth constantly; streaming application (CBR application) shows a medium and periodic utilization of bandwidth; and web application (interactive application) shows irregular bandwidth usage ratio and has a big standard deviation of bandwidth utilization. Telnet application (another interactive application) shows irregular and low bandwidth utilization [1, 9].

$$BandwidthUsageRatio = \frac{UsedBandwidth * 100}{MaximumBandwidth} \quad (1)$$

The size of the received packet is different according to different types of applications. FTP application has a bigger packet size than streaming application; streaming application has a bigger packet size than web application; and web application has a bigger packet size than telnet application. The size of the received packet relates to burst level. Burst level shows how much data is currently being transmitted. Equation (2) shows the method that calculates burst level.

$$BurstLevel = \frac{ReceivedPacketSize}{MaximumPacketSize} \quad (2)$$

The burst level of a web application is relatively high compared to telnet application due to the amount of transmission of data. The network traffic can be classified as interactive applications with calculated bandwidth usage ratio, size of received packet, and burst level as described above.

If network traffic is determined to be interactive applications, the system needs to collect information such as listening interval, elapsed time, and request interval from the WNIC and TCP layer for the 2nd step of the proposed mechanism. The power mode and several parameters of the CPU and the WNIC are determined in the second step. Listening interval should be controlled in the range that can guarantee QoS. The degradation rate of QoS is conditional on the type of applications. Table 1 shows the success ratio of transmission of data according to listening intervals. We control listening interval from 100ms to 400ms for the proposed method. [1] shows a similar experimental result to Table 1.

Table 1. QoS of transmission of web application data according to listening interval

Type of web page \ Listening interval	100ms	400ms	800ms
Text-centric	100%	100%	89%
Normal	100%	100%	80%
Image-centric	100%	98%	76%

Equation (3) shows the method that determines the best listening interval. Best listening interval can be calculated with request interval, packet size, available data rate, elapsed time, and number of available buffered packets. Let *BLI*, *RI*, *ET*, and *ABP* represent best listening interval, request interval, elapsed time, and available buffered packet, respectively. *c* is a proportional constant.

$$BLI = c * \frac{\#ofABP * PacketSize * RI * ET}{AvailableDataRate} \quad (3)$$

ET can be calculated by the difference between time stamp for sending and time stamp for receiving. *ET* is a kernel level information. *RI* is calculation by Equation (4). *RI* is related to user interactivity. Let *TSS* represent time stamp for sending. TSS_{pre} means previous time stamp for sending.

$$RI = TSS - TSS_{pre} \quad (4)$$

Figure 1 shows the trace of the listening interval dynamically adjusted to find *BLI* for the image-centric web page. Lower voltage selects a longer listening interval because there are many idle times. Figure 2 shows the flowchart of proposed mechanism that can control voltage of the CPU, power mode and listening interval for the WNIC by using gathered information from WNIC and network traffic. The proposed scheme set the WNIC into PSM mode when the network traffic was classified as interactive application because interactive application works according to intermittent user request. The CPU is set into lowVF (low Voltage and Frequency) during periods of no traffic. To set the CPU into lowVF, the total amount of energy consumption for voltage transition and consumed energy in lowVF during a fixed listening interval is smaller than the energy consumption for highVF during

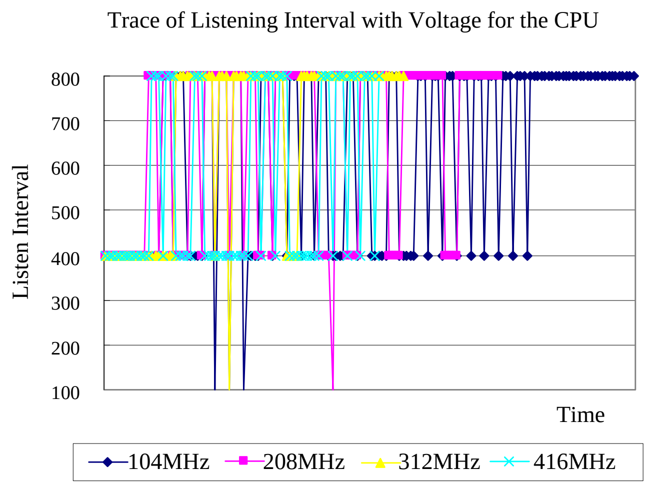

Figure 1 Trace of listening interval with various voltages for the CPU

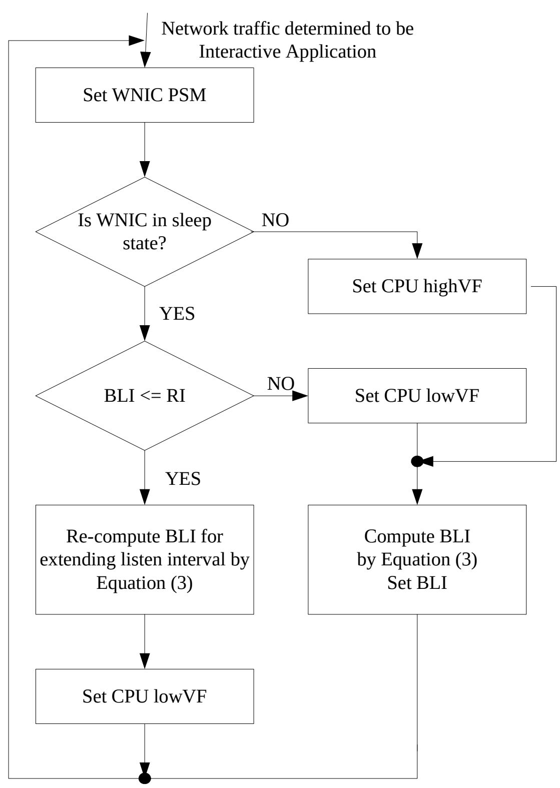

Figure 2 The proposed scheme to control power modes of the CPU and the WNIC for interactive applications

the same length of listening interval. If there is network traffic, it means that the WNIC is in an active state. In this case, the CPU is set into highVF to receive data from the AP (Access Point) and execute it as soon as possible. RI is another factor that influences the setting modes of the CPU and the WNIC as well as listening interval. RI is not easy to predict because every one has different RI in various situations. However, the proposed mechanism just requests a comparison with fixed *BLI* and *RI* to control the listening interval. If *RI* is equal or greater than *BLI*, the WNIC needs to be in a sleep state longer and the *BLI* needs to be recalculated applying *RI*. Burst level also can be used for setting the mode of the CPU. High burst level means high data rate. The CPU is set into highVF with high burst level.

3. EXPERIMENTS

We now compare the proposed scheme to the conventional method using the parameters of power consumption, time consumption, and energy consumption. The conventional methods to manage power mode of the CPU and the WNIC control the CPU and the WNIC separately. The CPU is set to DVFS and the WNIC is set to PSM. We measured power consumption of the CPU, WNIC, and the overall system on Mainstone II (Bulverde PXA270). We used CISCO AIRONET 350 for the WNIC and Agilient 34970A as the measuring instrument. The applications used for this experiment were telnet application and web application for interactive workload. We use three types of web page to evaluate web application: text-centric web page, normal web page, and image-centric web page. The voltages and frequencies used for evaluation of the CPU are: 104MHz (1.11V), 208MHz (1.24V), 312MHz (1.37V), and 416MHz (1.52V). The listening intervals used to evaluate the WNIC are: 100ms, 200ms, 400ms, and 800ms. We describe the evaluation results for 100ms, 400ms, and 800ms, because there was no difference of evaluation results between 100ms and 200ms.

In the case of telnet application, the listening interval influenced the energy consumption of handheld system more than voltage and frequency of the CPU. We compared the sum of energy consumption of the CPU and the WNIC with 416MHz and DVFS as the voltage and frequency for the CPU, and 100ms, 400ms, 800ms, and BLI as the listening interval for the WNIC. Many DVFS techniques have been proposed for the CPU. In this paper, we used the DVFS technique [20] that controls voltage of the CPU to extend the lifetime of handheld devices. Figure 3 shows the comparison result of energy consumption for highVF (416MHz) and DVFS for the CPU when the FTP application transmission is a 1Mbyte file. We used FTP application to evaluate the policy of voltage and frequency for the CPU because of constant bandwidth and consecutive data transmission. Figure 4 shows the energy consumption according to variation in the listening interval for telnet applications. We can see the efficiency of BLI from Figure 4. The energy consumption of the CPU and WNIC for telnet applications can be reduced by 28% with BLI.

Figure 5 shows the power consumption of the CPU and the WNIC according to the proposed mechanism for web applications. As we can see from Figure 5, it is efficient to control the CPU in highVF and the WNIC in PSM mode with BLI. To operate the CPU in high-speed helps to reduce the time period that WNIC is in an active state. We observed that on average 35% of energy is saved when the CPU operates with 416MHz compared to 104MHz. Figure 6 shows the comparison of time consumption with the variation of voltage and frequency of the CPU for a text-centric web page, a normal web page, and an image-centric web page. In common with the three types of web page, higher voltage and frequency shows shorter time consumption. Time consumption influences the level of power and energy consumption. Figure 7 shows the sum of energy consumption of the CPU and the WNIC for web applications according to voltage and frequency of the CPU with the proposed mechanism. The experimental results show that on average, 57% of the sum of energy consumption of the CPU and the WNIC is saved when the CPU is set to highVF. The reason why high voltage and frequency of the CPU can reduce the energy consumption of the overall system is that the duration of active state of the WNIC is shortened by the fast processing of incoming network traffic. This means that lowVF of the CPU can increase energy consumption of the WNIC while it can reduce energy consumption of the CPU. Therefore, the interaction between the CPU and the WNIC should be considered for the efficient reduction of energy consumption.

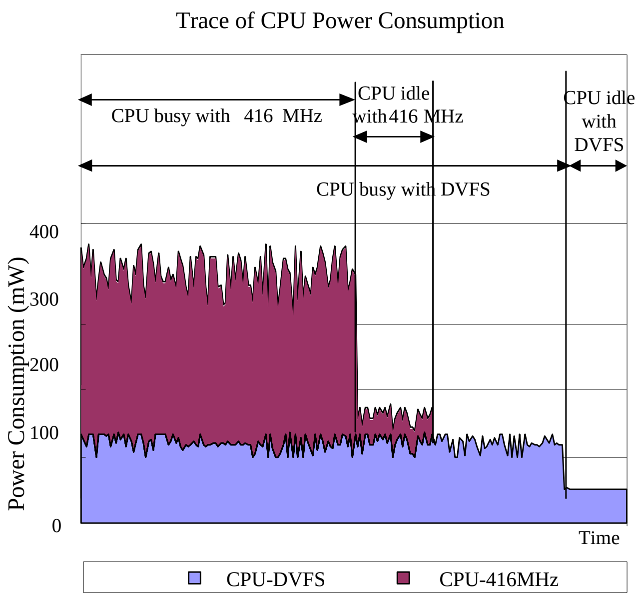

Figure 3 The comparison of power consumption between 416MHz and DVFS for the CPU

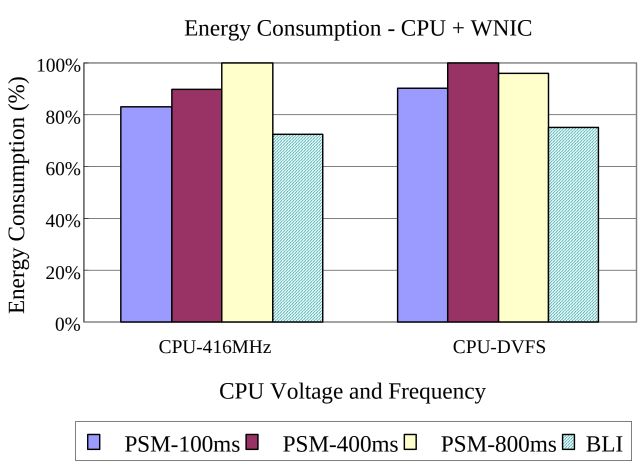

Figure 4 The comparison of energy consumption according to listening interval for telnet application

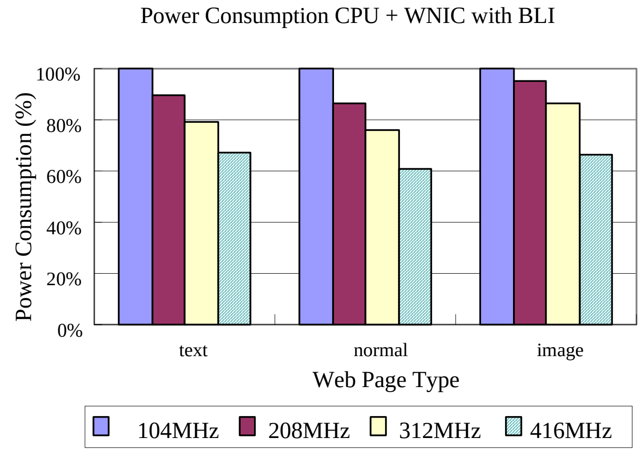

Figure 5 Sum of power consumption of the CPU and the WNIC for web application according to the proposed method.

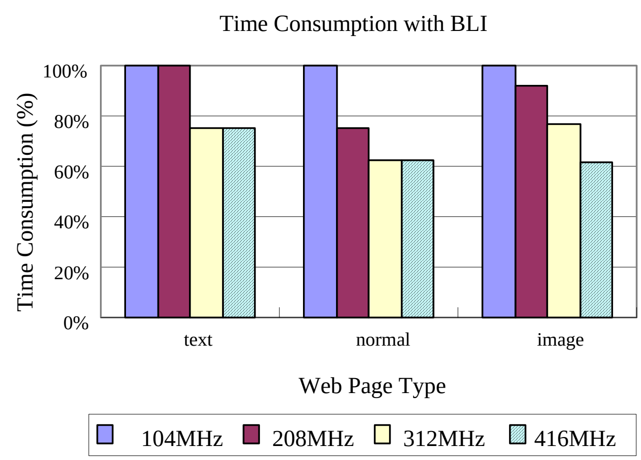

Figure 6 Time consumption for web application

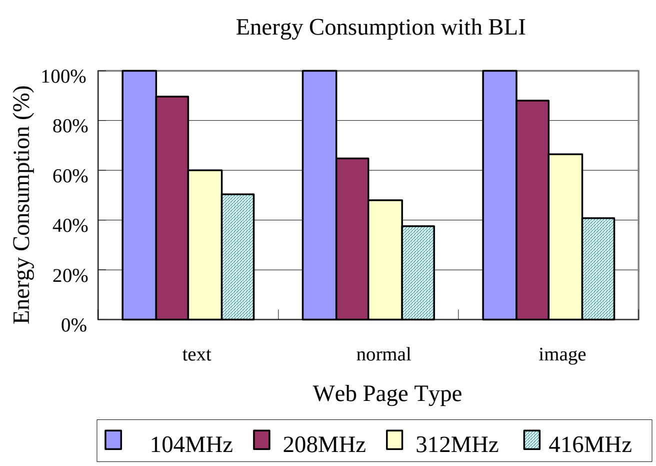

Figure 7 Sum of energy consumption of the CPU and the WNIC for web application with the proposed mechanism.

4. CONCLUSION

In this paper, we described an efficient integrated power management mechanism to reduce the energy consumption of handheld systems for interactive applications, represented by web applications. The proposed scheme has two contributions; First, the proposed method analyzes the interaction between the CPU and WNIC that were considered orthogonal in conventional research. The second is to control the voltage and frequency of the CPU with gathered information such as bandwidth utilization, listening interval, elapsed time, request interval, packet size, and burst level from the WNIC and network traffic. We dynamically control listening interval to enhance the reduction of energy consumption and calculate the best listening interval called BLI. To find the best listening interval, we also consider request interval which is influenced by user activity. We control BLI with the result of the comparison between BLI and request interval. The calculated BLI enables the system to find the most efficient voltage and frequency of the CPU and to extend the period in sleep state for the WNIC. We have shown that on average the proposed scheme can reduce energy consumption for interactive applications by 46%.

5. ACKNOWLEDGMENTS

This work was supported in part by the National Research Laboratory (NRL) program of the Korea Science and Engineering Foundation (2005-01352), the Seoul R&BD Program of City of Seoul, Korea, and the MIC(Ministry of Information and Communication), Korea, under the ITRC support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2006-C1090-0603-0015).

6. REFERENCES

[1] Weissel, A., Faerber, M., and Bellosa, F. Application Characterization for Wireless Network Power Management. Proceedings of the International Conference on Architecture of Computing Systems (ARCS'04), 2004.

[2] Chandra, S. and Vahdat, A. Application-specific Network Management for Energy-aware Streaming of Popular Multimedia Formats. Proceedings of USENIX Annual Technical Conference, 2002.

[3] Zhong, L. and Jha, N. K. Dynamic Power Optimization of Interactive Systems. VLSI Design, 1041–1047, 2004.

[4] Grunwald, D., Levis, P., C. B. M. III, Neufeld, M., and Farkas, K. I. Policies for Dynamic Clock Scheduling. Proceeding of the Fourth USENIX Symposium on Operating Systems Design and Implementation, 72-86, 2000.

[5] Lorch, J. Operating systems techniques for reducing processor energy consumption. Ph.D. thesis, University of California at Berkeley, 2001.

[6] Govil, K., Chan, E., and Wasserman, H. Comparing algorithms for dynamic speed-setting of a low-power CPU. International Conference on Mobile Computing and Networking, Berkeley, California, United States, 13–25, 1995.

[7] Benini, L. and Michele, G. System-level power optimization: techniques and tools. ACM Transactions on Design Automation of Electronic Systems, 115–192, 2000.

[8] Varma, A., Ganesh, B., Sen, M., Choudhury, S. R., Srinivasan, L., and Bruce, J. A Control-Theoretic Approach to Dynamic Voltage Scheduling. Proceeding of International Conference on Compilers, Architectures and Synthesis for Embedded Systems, 255–266, 2001.

[9] AbouGhazaleh, N., Mayo, R. N., and Ranganathan, P. Idle time Power management for personal Wireless Devices. Technical Reports, HP laboratories, 2003.

[10] Jones, C. E., Sivalingam, K. M., and Agrawal, P., Chen, J. C. A survey of energy efficient network protocols for wireless networks. Wireless Networks, Vol.7, No.4, 343–358, 2001.

[11] Martin, T.L. Balancing Batteries, Power and Performance: System Issues in CPU Speed—Setting for Mobile Computing. Ph.D. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1999.

[12] Flautner, K. and Mudge, T. Vertigo: automatic performance-setting for Linux. Proceedings of the Fifth USENIX Symposium on Operating Systems Design and Implementation, 105–116, 2002.

[13] Flautner, K., Reinhardt, S., and Mudge, T. Automatic performance-setting for dynamic voltage scaling. Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM-7), Rome, 260-271, 2001.

[14] Sorber, J., Banerjee, N., and Corner, M. D. Turducken: Hierarchical Power Management for Mobile Devices. Proceedings of The Third International Conference on Mobile Systems, Applications, and Services (MobiSys '05), Seattle, WA, June 2005.

[15] Rakhmatov, D. S. and Vrudhula. Energy management for battery-powered embedded systems. ACM Transactions on Embedded Computing System 2 (3), 277–324, 2003.

[16] Luo, J., Niraj, J. K. Battery-aware static scheduling for distributed realtime embedded systems design. Proceedings of ACM/IEEE Conference on Design Automation, Las Vegas, Nevada, United States, 444–449, 2001.

[17] Krashinsky, R., Balakrishnan, H. Minimizing Energy for Wireless Web Access with Bounded Slowdown. Proceedings of the Eighth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2002), 2002.

[18] Hohlt, B., Doherty, L., and Brewer. E. Flexible power scheduling for sensor networks. Proceedings of the Third International Symposium on Information Processing in Sensor Networks, Berkeley, CA, Apr. 2004.

[19] Qadeer, W., Rosing, T. S., Ankcorn, J., Krishnan, V., and Micheli, G. Heterogeneous wireless network management. Mobile and Media Systems Laboratory. HP Laboratories Palo Alto. HPL-2003-252. December 3, 2003.

[20] Choi, S., Cha, H., and Ha, R. A selective DVS technique based on battery residuals. Microprocessors and Microsystems, 1–10, 2005.