Heterogeneous wireless network management

W. Qadeer*, T. Simunic*^, J.Ankcorn^, V. Krishnan^ and G. De Micheli*
*Stanford University, ^HP Labs

Abstract—**Today's wireless networks are highly heterogeneous with diverse range and QoS. The maintenance of a wireless link by a mobile device requires support of multiple network interfaces. Since the battery lifetime is limited, power management of their interfaces without any significant degradation in performance has become essential. In our research we developed an integrated approach for the management of power and performance of mobile devices in heterogeneous wireless environments. Our policy decides what wireless network interface (WNIC) to employ for a given application and how to optimize the WNIC usage. This decision is governed by the current power and performance needs of the system. The policy dynamically switches between interfaces during program execution if data communication requirements and/or network conditions change. For the verification of our power and performance management algorithm, we have experimentally characterized Bluetooth and 802.11b wireless interfaces. We implemented our policy on HP's IPAQ portable device that is communicating with HP's HotSpot server [14]. The applications we tested range from MPEG video to email. The results show that our policy offers a large improvement in power savings as compared to singly using 802.11b or Bluetooth while enhancing performance.**

I. Introduction

Mobile communications today has heterogeneous wireless networks providing varying coverage and QoS. Various communication services are available. The infrastructure enables mobile devices to run applications with diverse bandwidth and network connectivity requirements, such as distributed speech recognition, video streaming, gaming etc. To satisfy the bandwidth and QoS constraints of the applications, the mobile devices need to allow seamless switching among various wireless network interfaces.. Additionally, the high communication and computation cost of applications is a burden on the battery life of portable devices. Capacity of a battery has not increased tremendously. Improvements of only a factor of 2-4 have been observed during the past 30 years. The ever-increasing need for battery lifetime in mobile devices demands a tighter control over its energy consumption.

Although low-power circuit design forms the basis of power management in a mobile device, higher-level management of power dissipation offers many more advantages. These techniques allow seamless integration between user applications and power management policy design thus allowing energy consumption to be reduced while maintaining a desired QoS. During program execution communication interfaces are placed in low-power states depending upon their acces patterns and application performance needs. Various components of the mobile system, such as the user, the wireless channel and the individual interfaces can be modeled using a state based abstraction.

The techniques developed to date for the enhancement of heterogeneous networks concentrate on improving their accessibility and QoS. These methods enable mobile devices to communicate with each other by introducing changes in the network protocol stack. They also allow establishment and maintenance of connections between mobile hosts using any available links to improve robustness and performance. However, none of the techniques adequately addresses power management. Power reduction methodologies presented in the past largely focus on improving energy consumption of one single device e.g. WaveLAN, CPU etc. Policies for the reduction of power dissipation range from simple time-out methods to complex techniques based upon stochastic models.

This work presents a new methodology for managing power and performance of mobile devices consisting of heterogeneous WNICs. The policy formulated decides what network interface to employ on a portable device for a given pattern of usage. The decision is governed by the current power dissipation and QoS requirements of the system. The maximum likelihood estimator is employed for tracking system changes. It detects variations in the average throughput of available wireless interfaces and the data usage patterns. The policy for power and performance management (PPM) decides:

a. What wireless network interface card to use
b. What low-power state to employ
c. Transition times between active and low-power states
d. Buffer size to use for good application QoS

We implemented the policy on HP's IPAQ portable device that is communicating with HP's HotSpot server [14] via Bluetooth and 802.11b. The applications we tested range from MPEG video to email. Our results show both large savings in power when using a single WNIC, as well as seamless switching with concurrent power savings among WNICs.

II. RELATED WORK

Mobile devices require wireless communication interfaces to facilitate connectivity with Internet and with the other devices. A mechanism is required for forwarding packets between different wireless networks due to increasing device mobility. Mobile IP [1] provides one example of such mechanism. Changes are introduced in the network and link layers of the network protocol stack that assist the host's home network in forwarding packets to its network of residence. However, with mobile IP even if communicating devices are in the same wireless network, data needs to traverse a multi-hop path. In order to perform localized communication between devices, which are one hop distance away, *Contact Networking* [2] has been proposed. By allowing seamless switching between multiple diverse interfaces, this technique enhances robustness and QoS of the network.

Mobile hosts experience varying data rates during communication in part due to lossy nature of the wireless link. In order to avoid disruptions, a distributed file system has been developed [3], [4]. It allows application aware and application independent adaptation to a temporary loss or degradation of the wireless link thus enhancing robustness. A method for improving hand-offs proposes buffering data on multiple base stations in close proximity to the mobile host [5] thus achieving seamless switching between base stations. Telephony and data services spanning diverse access networks have been integrated in [6]. However, the focus of these techniques has been on the enhancement of performance and QoS of heterogeneous networks. Power management of communicating hosts has been mainly overlooked.

Several techniques have been proposed to efficiently manage power dissipation in portable devices. These methods employ diverse mechanisms to predict periods of inactivity during communication. Based upon these predictions the mobile device is put into a low-power state. The most basic power management policy is a time-out. If the device remains idle for a certain period, it is put into a low power state. Similarly, a device can enter low-power mode when idleness is being anticipated in a connection [7]. However, incorrect estimates cause performance and power penalties. In contrast, stochastic models derive provably optimal power management policies. *Pure Markov decision processes* [8], [9] employ either discrete or continuous time memory-less distributions. However, discrepancies have been observed in predicted and actual power savings owing to history dependent nature of real world processes. *Time-indexed semi Markov decision processes* [10] are based upon history based distributions. This technique has demonstrated energy savings in real-world applications. The power management techniques presented to date mostly focus on the reduction of power dissipation in one WNIC. This leads to inefficient power management for portables with multiple diverse communication interfaces.

Methods being employed for the performance enhancement of homogeneous networks put a lot of emphasis on power management. IEEE 802.11 [11] standard implements power management by sending a traffic indication map (TIM) with the beacon to the client. It enables the client to enter doze mode if no more data is available. Since the device still has to wake up after every beacon interval for TIM, a new technique proposes decoupling of control and data channels [12]. The control channel uses low-power radio and wakes up the device whenever data is present. Application level information is used for power management in [13]. In our work we developed an integrated policy for power and performance management. Our power and performance management (PPM) algorithm dynamically selects the appropriate wireless network interface with the goal of minimizing the overall energy consumption while meeting application's QoS requirements. We present measurement results that show large energy saving with good QoS while using Bluetooth and 802.11b on HP's IPAQ for a typical set of applications.

The rest of the paper is organized as follows. Section 3 discusses the characterization of Bluetooth and 802.11b interfaces. Details of the heuristic policy for choosing among network interfaces are presented in Section 4 whereas the results and conclusion are discussed in Sections 5 and 6 respectively.

III. CHARACTERIZATION OF DEVICES

A. Bluetooth

Bluetooth has been developed as a radio link with a short range to provide wireless connectivity to portable and fixed devices. It operates in 2.4GHz ISM band. Bluetooth supports point-to-point and point-to-multi-point connections called *piconets*. A piconet can consist of two to eight active Bluetooth devices. One device is the master and the rest are its slaves. In addition, a master can support several other inactive slaves, which have been *parked*. These slaves remain synchronized to the master but do not become a part of the piconet. A s*catternet* is composed of multiple piconets with an overlapping coverage area. Bluetooth provides both synchronous and asynchronous connections. A synchronous connection can support 64kb/s in each direction whereas the asynchronous one can support a maximum of 723.2 kb/s asymmetric and 433.9kb/s symmetric data rates [16].

Bluetooth supports multiple low-power states: hold, sniff and park. As shown in the state space figure below, these states can only be activated once a connection exists between Bluetooth devices.

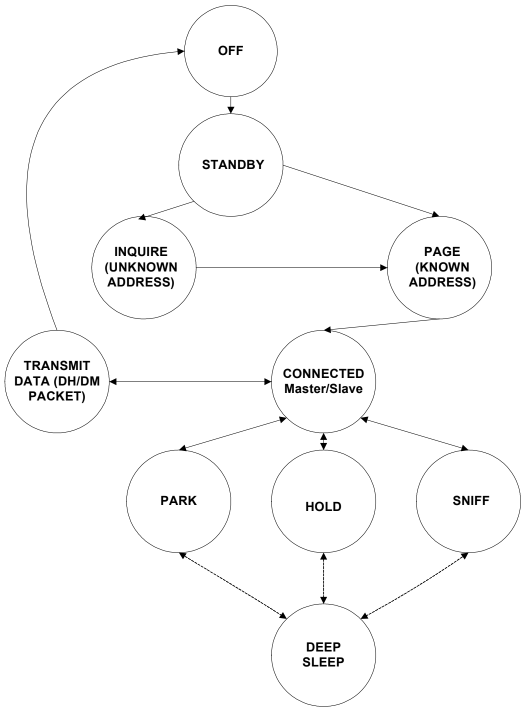

FIGURE 1. BLUETOOTH STATE SPACE

a. Hold mode is employed to stop data transfer by the requested device for a negotiated interval. It is especially useful if the requesting device wants to perform inquiry, page and scan or burst mode transfer operation in scatternets.

b. Sniff mode is useful on low data rate links such as email where a quick response is required whenever data is present. During an attempt window the device looks for any incoming data. If no data is present, it goes into low-power mode; however, if data is present, the device listens to the master for the specified time-out period. Sniff mode can also be useful in scatternets for devices that are a part of two piconets.

c. Park mode is used to enhance the number of simultaneous connected slaves. As link set up takes about 10s in Bluetooth, it is best to retain an established connection. In this mode no data transfer takes place as the parked slave gives up its connection ID but it remains synchronized to the master.

Transition times and average power dissipation for switching between active and hold, sniff or park mode are shown in Table 1. The transition times are very short. Consequently, Bluetooth devices can seamlessly shift between active and low-power states with very little energy dissipation. In addition, the CSR Bluetooth chips supports *deep sleep* state with only 270uW power consumption [15]. This state can only be entered when a BCSP based device is in one of the low-power states. Deep sleep state can be entered only if there is no activity on UART for at least 250ms.

TABLE 1. BLUETOOTH LOW-POWER MODE MEASUREMENTS

	TRANSITION TIME (M SEC)	AVG. POWER (W)
HOLD MODE		0.061
Hold mode entry	1.68	0.068
Hold mode exit	11.62	0.216
PARK MODE		0.061
Park mode entry	2.16	0.077
Park mode exit	4.12	0.126
SNIFF MODE		0.061
Sniff mode entry	0.94	0.078
Sniff mode exit	7.36	0.194

Bluetooth supports multiple packet types for both asynchronous and synchronous connections. These packet types differ in data payload size and error correction algorithms. Maximum achievable throughput for various packet types is tabulated in Table 2. Our measurement results come close to throughput values reported in Table 2. For instance, the maximum and the average throughput numbers were measured on CSR Bluetooth for DH5 packets at 87kB/s and 79kB/s respectively. The throughput increases with an increase in the payload capacity of the base-band packet. However, throughput can significantly decreased in the presence of noisy channels if less error-correction bits are present. The range of Bluetooth devices is enhanced by an increase in transmission power; but that causes a further energy drain from the battery.

TABLE 2. MAXIMUM THROUGHPUT FOR BLUETOOTH ACL CONNECTION

	Symmetric max rate (kb/s)	Asymmetric max rate (kb/s)	
		Forward	Reverse
DM1	108.8	108.8	108.8
DM3	258.1	387.2	54.4
DM5	477.8	477.8	36.3
DH1	172.8	172.8	172.8
DH3	390.4	585.6	86.4
DH5	433.9	723.2	57.6

B. WaveLan-802.11b

802.11b has been developed to provide fast wireless connectivity to mobile devices. Theoretically, 802.11b can support a maximum data rate of 11Mbps in 2.4GHz band. It is designed to work in adhoc as well as infrastructure network topologies. Today a large majority of all WLAN communication happens in infrastructure mode, thus this is the mode we will be focusing on in this work. An access point acts as a bridge between wired and wireless networks. An association is developed between the access point and the 802.11b card before commencing data communication. In order to facilitate mobility, access-points also support roaming.

WLAN has two active states, transmit and receive, in addition to two low-power modes, doze and off. Table 3 shows average power dissipation measurements for the above mentioned power states. According to the 802.11b standard, a synchronization beacon is transmitted to the awake card by a central access point (AP) every 100ms. The beacon is followed by a traffic indication map (TIM) indicating any required data transfers. Doze mode is activated until the next beacon if no data transfer is required. This power management (PM) policy does not always give optimal power savings due to the following factors:

a. An increasing number of clients causes radios to stay on longer since there is more contention due to multiple simultaneous synchronization attempts by the mobiles.
b. 802.11b's response time to the AP suffers due to the delays imposed by the doze mode.
c. Even without any running applications, 802.11b spends a considerable amount of time listening with an increase in broadcast traffic and is thus unable to enter doze mode.

Doze mode can only be activated by the hardware. However, transitions to the off state from either the active or the doze state can be controlled at the OS level. Transition times and average power dissipation for switching between active and low-power states have been tabulated in Table 3 for Cisco 350 WNIC.

TABLE 3. 802.11B LOW-POWER MODE MEASUREMENTS

	TRANSITION TIME (MS)	AVG. POWER (W)
OFF TO DOZE STATE		
Doze state entry	0.1	1.4
Doze state exit	1	1.6
ACTIVE TO OFF STATE		
Off state entry	1	1.7
Off state exit	300	2.3

IV. POWER AND PERFORMANCE MANAGEMENT

The goal of PPM is to enhance QoS while minimizing power dissipation in a portable device. PPM's primary task is to determine what network interface is most suitable for the application needs and how to manage its power and performance states. When an application starts on a portable device, PPM pre-selects those WNICs for data communication whose average throughput is greater than the data consumption rate of the application. This ensures that the QoS requirements of the application are satisfied. In streaming applications a special emphasis is placed upon the data buffer size. It not only determines the average sleep time of the communication device but also the energy dissipated in the RAM. Since the size of the buffer is determined by the difference between the throughput and the data consumption rate, all the pre-selected WNIC are further examined to determine not only their communication power dissipation but also the resulting RAM power consumption. The one that offers maximum power savings is selected. Additionally, during the examination of the communication energy only those low-power states, which are most suitable for the current scenario, are considered. PPM also defines the switching time between active and sleep states for the selected WNIC.

PPM dynamically keeps track of the variations in the application data consumption rate and the throughput of wireless interfaces using the log of the maximum likelihood estimator as shown in equation (1). A change in rate is defined to occur at point *c* when computed likelihood over the last *w* data points is greater than a preset threshold. In our work we use 99.5% as a threshold. The change is observed between the old, λ_{old}, and the new rate, λ_{new}. Details of this algorithm are further discussed in [17].

$$\ln(P_{\max}) = (w-c+1)\ln\frac{\lambda_{new}}{\lambda_{old}} - \left(\lambda_{new} - \lambda_{old}\right)\sum_{j=k}^{m}\Delta t_j \qquad (1)$$

Whenever a change occurs, the PPM evaluates which subset of WNICs could handle the applications currently running by insuring that the available WNIC's throughput rate, λ_t, is greater than the application's data consumption rate, λ_u.

$$\lambda_t \geq \lambda_u \qquad (2)$$

Network interfaces satisfying equation (2) are further analyzed to identify the interface that offers maximum power savings for the given application while keeping the required quality of service. The total energy, E_{total}, consumed during a given session along with the average power dissipation, P_{avg}, is given by the following equations:

$$E_{total} = E_{comm} + E_{RAM} + P_{switch}T_{switch} \qquad (3)$$

$$P_{avg} = E_{total} / \left(\left(B_{active} / \lambda_u\right) + T_{switch}\right) \qquad (4)$$

Where E_{comm} and E_{DRAM} denote the energy consumed by the WNIC and the DRAM respectively during the communication period, B_{active} specifies the size of the buffer for streaming applications that is actively read and written to in steady state. $\mathrm{P_{switch}}$ and $\mathrm{T_{switch}}$ indicate the average power dissipated and the time taken when switching from one WNIC to another. Only WNICs with throughput high enough to meet applications demands are considered. Figure 2 depicts the process of switching interfaces. The details of this procedure are further elaborated in [2] and [18].

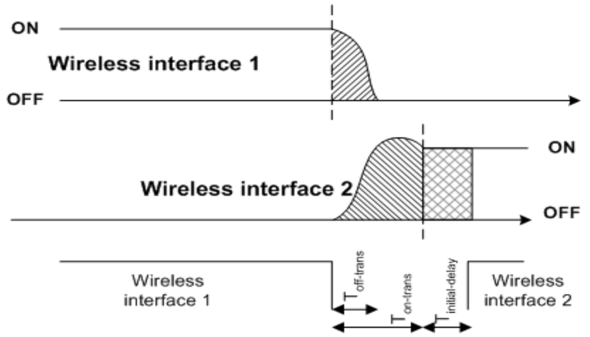

FIGURE 2. PROCEDURE FOR SWITCHING WIRELESS NETWORK INTERFACES

In streaming applications the size of the buffer directly affects E_{comm} and E_{DRAM}. If the size of the buffer undergoes an increase, the average power dissipation of the communication device diminishes due to longer sleep periods and thus less overhead in transition between power states. On the other hand, the RAM energy increases with increasing buffer sizes as the number of active banks increases. Thus the determination of the buffer size is of principal importance for enhancing power savings.

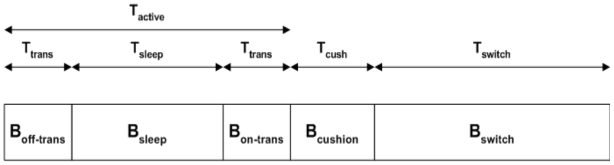

FIGURE 3. BUFFER LAYOUT AND THE ASSOCIATED TIME INTERVALS

The size of the buffer is chosen in a way such that the transmissions could be scheduled in bursts. In between bursts the WNIC can transition into low power mode, thus saving energy and freeing bandwidth from contention. The size of the buffer is determined according to the equations given below.

$$B = B_{active} + B_{switch} + B_{cush} \tag{5}$$

$$B_{active} = B_{sleep} + B_{on-trans} + B_{off-trans} \tag{6}$$

$$B_{sleep} = \left(T_{sleep}\lambda_t\lambda_u\right)/\left(\lambda_t - \lambda_u\right) \tag{7}$$

$$B_{off-trans} = \left(T_{off-trans}\lambda_t\lambda_u\right)/\left(\lambda_t - \lambda_u\right) \tag{8}$$

$$B_{on-trans} = \left(T_{on-trans}\lambda_t\lambda_u\right)/\left(\lambda_t - \lambda_u\right) \tag{9}$$

$$B_{switch} = T_{switch}\lambda_u \tag{10}$$

$$B_{cush} = T_{cush}\left(\lambda_t\left(1-\chi\right) - \lambda_u\left(1+\delta\right)\right) \tag{11}$$

$$\left(T_{be}\lambda_t\lambda_u\right)/\left(\lambda_t - \lambda_u\right) \le B \le B_{\max} \tag{12}$$

Where:

a. T_{sleep} is the average sleep interval of the communication device.
b. T_{switch} accounts for the worst case delay encountered in dynamically switching between two network interfaces.
c. T_{cush} provides a cushion for any small variations present in the system.
d. χ and δ denote small variations in throughput and data consumption rate respectively.
e. $T_{off-trans}$ and $T_{on-trans}$ are the transition times between active and low-power and low-power and active states respectively.
f. T_{be} is the break-even time and is defined in terms of power consumed during the transition, $P_{trans} = P_{on-trans} + P_{off-trans}$, the power consumed in the active and sleep states, P_{active} and P_{sleep}.

$$T_{be} = T_{trans} + T_{trans}\frac{P_{trans} - P_{active}}{P_{active} - P_{sleep}} \tag{13}$$

g. B_{max} is the maximum amount of memory available.

When an application starts, it waits for $T_{initial-delay}$ before beginning to read from the buffer. This time interval is influenced by the maximum delay a user can tolerate at start-up. Thus the time to enter steady state, $T_{steadystate}$, is given by equation 14. During this interval the communication device stays in the active mode.

$$T_{steadystate} = T_{initial-delay} + \left(B - T_{initial-delay}\lambda_t\right)/\left(\lambda_t - \lambda_u\right) \tag{14}$$

The total energy consumed by the communication device, E_{comm}, during the buffer refill period is given by equation (15). Note that for simplicity the transition power and time have been combined into one variable, P_{trans} and T_{trans}. The communication energy needs to be balanced by the energy consumed by memory, as larger buffer sizes cause higher energy consumption.

$$E_{comm} = P_{active}\left(T_{active} + T_{cushion}\right) + P_{trans}T_{trans} + P_{sleep}T_{sleep} \tag{15}$$

The amount of energy consumed by memory, E_{RAM}, is determined from the energy consumed by the banks that are actively participating in reading and writing of data, E_{active}, and the energy of non-active banks, $E_{non-active}$:

$$E_{RAM} = E_{active} + E_{non-active} \tag{16}$$

$$E_{active} = P_{write}T_{active} + \left(P_{read} + N_{abanks}P_{refresh}\right)\left(B_{active}/\lambda_u\right) \tag{17}$$

$$E_{non-active} = \left(N_{banks} - N_{abanks}\right)P_{non-active} \tag{18}$$

$$N_{abanks} = \left\lceil B/Size_{bank} \right\rceil \tag{19}$$

Where:

a. P_{write} and P_{read} specify the average power dissipated when the RAM is written and read respectively.
b. N_{banks} specifies the total number of available memory banks.
c. $P_{non\text{-}active}$ is the average power consumed by the memory banks that are non-active.
d. $P_{refresh}$ is the average power spent in refreshing the active banks not participating in read and write operations.
e. N_{abanks} is the number of memory banks where each has size $Size_{bank}$

Figure 4 depicts the decrease in communication energy as the buffer size increases whereas the increase in RAM energy with an increase in the buffer size is shown in Figure 5.

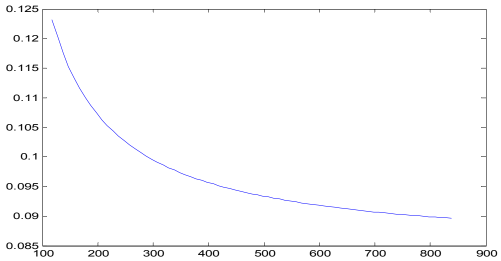

FIGURE 4. COMMUNICATION POWER VS BUFFER SIZE

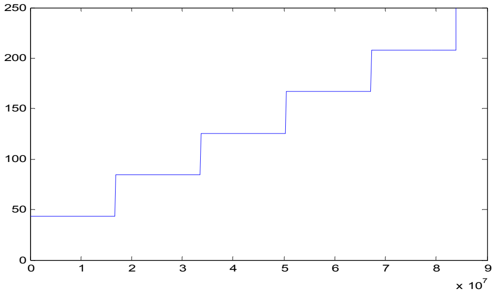

FIGURE 5. DRAM POWER VS BUFFER SIZE

Thus PPM is employed to enhance performance while minimizing power dissipation. PPM pre-selects WNICs for a particular application based upon their average throughputs and the data consumption rate of the application. The WNIC that offers minimum power dissipation with regards to communication and RAM energies is selected. The appropriate low-power state of the WNIC along with the switching points is also defined by PPM. Additionally, it can dynamically switch the selected WNIC if a change in its throughput and/or the average data consumption rate of the application is detected. In the next section we present the results obtained by using our PPM with a typical set of applications having diverse data usage patterns.

V. RESULTS

The server used is a research prototype of HP's HotSpot [14] server whereas the policy has been implemented on IPAQ 3970 that supports both 802.11b (CISCO Aironet 350 PCMCIA WLAN) and Bluetooth (CSR) interfaces. The operating system running on the IPAQ is Linux. The power measurements have been performed with a DAQ card at 10ksamples/sec. In our experiments, we have used transmission control protocol (TCP) for all data communication. For Bluetooth this has been done using *bnep.*

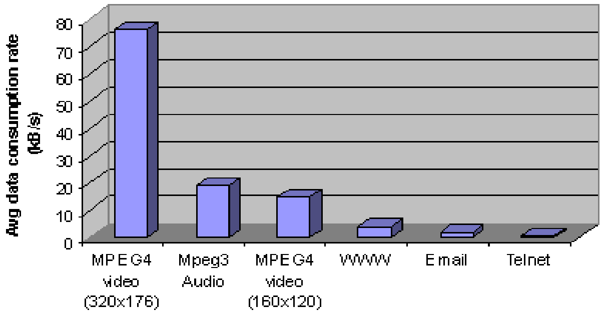

FIGURE 6. APPLICATION DATA CONSUMPTION RATE (kB/s)

At first we consider individual applications and employ our PPM to determine the appropriate WNIC for each one based upon its data consumption rate and the average throughput supported by the WNIC. For this experiment we have assumed that the throughput and the data usage pattern do not change significantly during program execution.

For MP3 audio streaming Bluetooth and 802.11b offer similar performance, but 802.11b gives more power savings when used in the off mode. Best power savings in Bluetooth are obtained in park with deep sleep enabled. In the off state 802.11b does not consume any power. In contrast Bluetooth cannot be turned off as the connection drops whose reestablishment requires 1-10 sec. The results are shown in Figure 7.

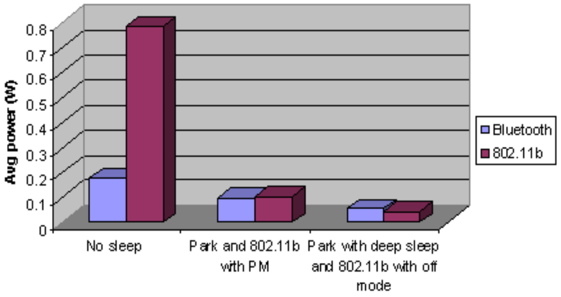

FIGURE 7. POWER CONSUMPTION OF WIRELESS INTERFACES FOR MP3 AUDIO

Bluetooth is more suitable than 802.11b for email traffic as it offers larger power savings. The appropriate low-power state for Bluetooth is again Park with deep sleep enabled. Similarly for 802.11b, the appropriate low-power mode is off mode. However, 802.11b incurs a large power dissipation penalty when switching from off to active state. Thus Bluetooth with park and deep sleep behaves better than 802.11b with off mode enabled. The power measurements are shown in Figure 8.

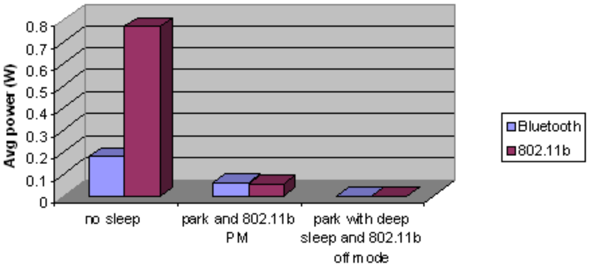

FIGURE 8. POWER CONSUMPTION OF WIRELESS INTERFACES FOR EMAIL

Bluetooth seems to be the connection of choice for telnet and WWW based applications owing to its faster response time and low-power dissipation as compared to 802.11b. Again the suitable low-power state for Bluetooth is Park with deep sleep enabled. However, due to a significant decrease in the response time for 802.11b with off mode, the low-power state of choice for 802.11b is PM. The switching overhead associated with the results is shown in figures 9 and 10.

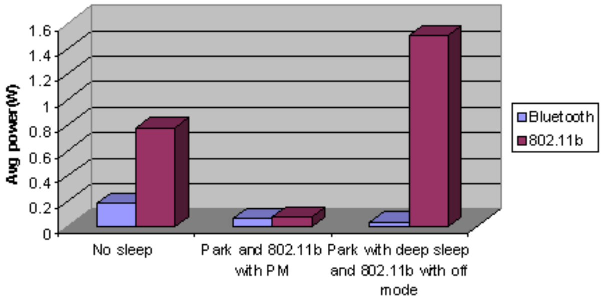

FIGURE 9. POWER CONSUMPTION OF WIRELESS INTERFACES FOR TELNET

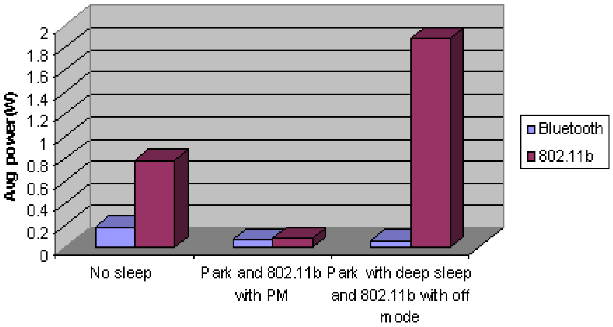

FIGURE 10. POWER CONSUMPTION OF WIRELESS INTERFACES FOR WWW

For MPEG4 video streaming, Bluetooth can only be employed for small images due to its lower average throughput. The low-power mode for Bluetooth is again park with deep sleep enabled. However, 802.11b with off mode offers more power savings due to higher throughput and non-existent power dissipation in the off state. The results are shown in Figure 11.

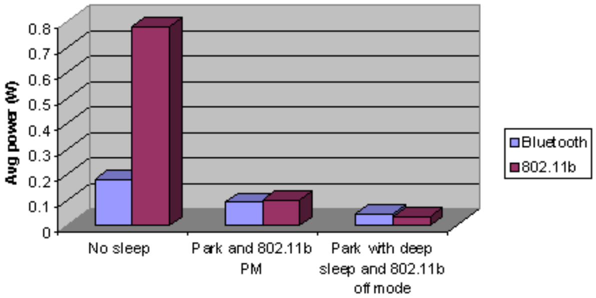

FIGURE 11. POWER CONSUMPTION FOR MPEG4 VIDEO (160X120)

For streaming applications, power dissipated by the communication device is an increasing function of the buffer size. However, by using PPM in our particular setup, we found that the buffer size is limited by the power dissipated in the DRAM. If the size of the buffer is increased by one, the cumulative average power dissipation of the wireless card and the DRAM doubles.

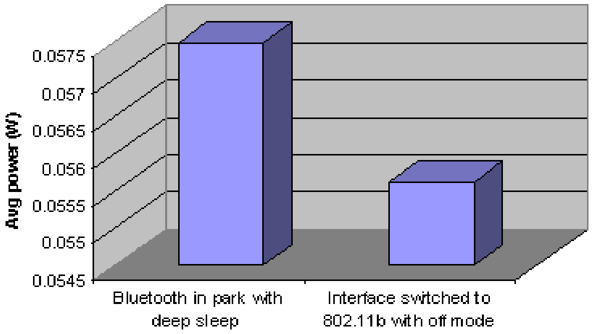

FIGURE 12. THROUGHPUT CHANGE FORCES A WNIC SWITCH

Next we present results of the experiment that include a dynamic switch between wireless interfaces during program execution. In this experiment the data consumption rate of the application stays constant, but the throughput of the selected WNIC undergoes a change. Let's suppose that a person with a portable device is streaming MP3 audio using Bluetooth. After a certain period of time he moves away from the server and the throughput of Bluetooth experiences a sharp decrease. The estimator detects the change in throughput and forces PPM to reevaluate the suitability of the selected network interface. The reevaluation suggests a change in the network interface and 802.11b is chosen over Bluetooth. The comparison of power dissipation is shown in Figure 12.

Lastly, we analyze the performance of PPM when application data consumption rate changes. We created an application trace consisting of MP3 audio, Email, telnet, WWW and MPEG4 video. The trace is executed first by using 802.11b only with PM enabled, secondly by employing Bluetooth only with park as the low-power mode and in the end by PPM with Bluetooth and 802.11b as the available wireless interfaces. We found that PPM offers a factor of 2.9 times improvement in power savings over just employing Bluetooth with park mode whereas in comparison to 802.11b with PM the power savings are a factor of 3.2 times higher. Moreover, PPM enhances the QoS as wireless interfaces are switched to match the data usage pattern of the application. The results are shown in Figure 13.

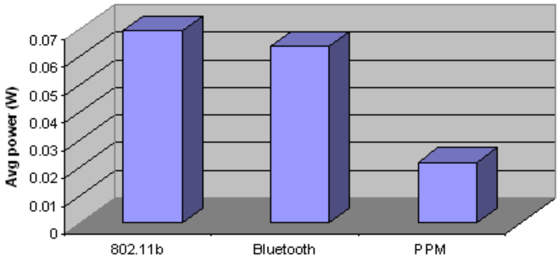

FIGURE 13. CHANGE IN DATA CONSUMPTION RATE FORCES A WNIC SWITCH

When the application trace is executed using 802.11b only, the wireless interface is placed in the doze mode whenever TIM indicates periods of inactivity. Similarly for Bluetooth, the interface is placed in the park mode whenever no data communication is needed. However, due to the varying data usage patterns, throughputs and low-power modes of the wireless cards, power savings vary considerably from 802.11b to Bluetooth. PPM selects the appropriate wireless interface card at the start of each application. For this experiment the interfaces are not dynamically switched during program execution.

VI. CONCLUSION

This work presents a new methodology for enhancing QoS while maximizing power savings in heterogeneous wireless systems. A policy for selecting the most appropriate network interface for a particular application has been developed. We have tested our policy on IPAQ 3970 supporting 802.11b and Bluetooth wireless interfaces using various typical applications. We have shown that our PPM offers 2.9 and 3.2 power savings over solely using Bluetooth and 802.11b respectively when running a string of applications including MPEG4 video and MP3 audio.

VII. REFERENCES

[1] D. B. Johnson, and D. A. Maltz, “Protocols for adaptive wireless and mobile networking,” IEEE Personal Communications, 1996.

[2] C. Carter, R. Kravets, and J. Tourrilhes, “Contact Networking: A Localized Mobility System,” in Proc. of the First International Conference on Mobile Systems, Applications, and Services, 2003.

[3] J. Flinn, M. Satyanarayanan, “Energy-aware adaptation for mobile applications,” Proceedings of the 17th ACM Symposium on Operating Systems Principles, 1999.

[4] B. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, K. Walker, “Agile Application-Aware Adaptation for Mobility,” in Proc. of 16th ACM Symposium on Operating System Principles, 1997.

[5] Eric A. Brewer, Randy H. Katz, Elan Amir, Hari Balakrishnan, Yatin Chawathe, Armando Fox, Steven D. Gribble, Todd Hodes, Giao Nguyen, Venkata N. Padmanabhan, Mark Stemm, Srinivasan Seshan, and Tom Henderson, “A network architecture for heterogeneous mobile computing,” IEEE Personal Communications Magazine, 5(5):8 -- 24, 1998..

[6] Helen J. Wang, Bhaskaran Raman, Chen-nee Chuah, Rahul Biswas, Ramakrishna Gummadi, Barbara Hohlt, Xia Hong, Emre Kiciman, Zhuoqing Mao, Jimmy S. Shih, Lakshminarayanan Subramanian, Ben Y. Zhao, Anthony D. Joseph, and Randy H. Katz, “ ICEBERG: An Internet-core Network Architecture for Integrated Communications,” IEEE Personal Communications: Special Issue on IP-based Mobile Telecommunication Networks, 2000.

[7] C. H. Hwang and A. Wu, “A Predictive System Shutdown Method for Energy Saving of Event-Driven Computation”, in International Conference on Computer Aided Design, pp. 28–32, 1997.

[8] E. Chung, L. Benini and G. De Micheli, “Dynamic Power Management for non-stationary service requests”, Design, Automation and Test in Europe, pp. 77–81, 1999.

[9] Q. Qiu and M. Pedram, “Dynamic power management of Complex Systems Using Generalized Stochastic Petri Nets”, Design Automation Conference, pp. 352–356, 2000.

[10] T. Simunic, L. Benini, P. Glynn, G. De Micheli, “Event-Driven Power Management,” IEEE Transactions on Computer-Aided Design, July 2001.

[11] The Editors of IEEE 802.11, IEEE P802.11D5.0 Draft Standard for Wireless LAN, July, 1996.

[12] E. Shih, P. Bahl, M. J. Sinclair, “Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices,” in Proc. of ACM MobiCom, 2002.

[13] A. Acquaviva1, T. Simunic, V. Deolalikar, and S. Roy, “Remote power control of wireless network interfaces,” in Proc PATMOS, 2003.

[14] D. Das, G. Manjunath, V. Krishnan, P. Reddy, “HotSpot! – a service delivery environment for Nomadic Users System Architecture,” Mobile Systems and Storage Lab., HP Laboratories Palo Alto, CA Rep. HPL-2002-134, 2002.

[15] CSR, “BlueCore Power Saving Modes,” Jan, 2003.

[16] Bluetooth Special Interest Group, "Specification of the Bluetooth System 1.1, Volume 1: Core," http://www.bluetooth.com, Feb, 2001.

[17] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli, “Dynamic voltage scaling for portable systems,” in Proceedings of the 38th Design Automation Conference, June 2001.

[18] J. Tourrilhes and C. Carter, “P-Handoff: A protocol for fine-grained peer-to-peer vertical handoff,” Proc. of PIMRC, 2002.