Energy-Efficient Upload Engine for Participatory Sensing

Takahiro Yamamoto*†, Shunsuke Saruwatari*, Hiroyuki Morikawa*
*Research Center for Advanced Science and Technology, University of Tokyo, Japan
†CORE Research and Development Institute, Japan

***Abstract*—Participatory Sensing enables us to build a large platform for wide-area sensing by utilizing cellphones as sensor nodes. One of the main problems in the participatory sensing is the high power consumption on a cellphone because of the limitation of its battery. In this paper, we propose a low power data transport protocol, which is called as Piggyback Transport Protocol (PBTP). The PBTP utilizes the *user think time* to upload sensor data. In the *user think time*, a user watches the display and thinks about what should he or she do next. The PBTP uploads sensor data in the *inactivity timer period* on the *user think time*. The *inactivity timer period* represents the cellphone has connection to a base station, but does not communicate to the base station. The paper evaluates the PBTP with simulation, and the result shows PBTP reduces power consumption compared to the previous work.**

I. INTRODUCTION

Sensor network enables us to retrieve real-space information and has renovated a lot of research until now. However, when collecting the wide-area information, such as urban sensing and habitat monitoring, sensor network faces problems of large installation cost and large operation load[1]. Participatory sensing[2] and people-centric sensing[3] can collect the wide-area information at low cost and with small operation load by utilizing the user's mobile phone with the various kinds of sensors such as GPS, camera, microphone, accelerometer, gyroscope and so on. The mobile phone senses the environment around the user and uploads sensor data via wireless network periodically. Participatory sensing is also currently under study in fields, such as air monitoring[4], [5], noise monitoring[6], [7], road monitoring[8], personal monitoring[9], social network tool[10], [11] and so on.

Since mobile phone runs on battery, participatory sensing will require a new power-saving technique. The existing research in sensor network and participatory sensing reduces power consumption by minimizing the active time of sensing device and communication device under a single sensing application environment[12], [13], [14]. However, multiple applications run on a mobile phone at the same time. Therefore, participatory sensing needs an energy-saving technique which is applicable to various sensor environments and realized at communication protocol level.

In mobile network, a cellphone decreases connection latency and overheads by maintaining a connection for a certain period after data transfer ends. If no data transfer occurs for this period, then the inactivity timer[15], [16] expires and the mobile phone terminates its wireless connection. In this paper, we define the period from the completion time of the last transfer to the expiration time of inactivity timer as *inactivity timer period*. If its wireless connection is established every time a request occurs, all the connections have *inactivity timer period* and the energy consumption increases. To tackle this problem, TailEnder[17] shortens the total time of the *inactivity timer periods* by buffering requests until their deadline and sending them in groups at the earliest deadline. However, when being applied to participatory sensing, TailEnder degrades the user operability because the user's request and the sensor data upload occur at the same time and the communication throughput decreases.

We focus on the periods when user thinks about the next action during his or her operation and propose PBTP (Piggyback Transport Protocol) which uploads sensor data during the *user think time*. By improving the utilization efficiency of communication device, PBTP reduces the energy consumption. PBTP buffers sensor data and uploads them during the unused time of the communication device which happens during the user's web browsing operations. When detecting the generation of the user request or the termination of the user operation, PBTP stops uploading the sensor data.

The main contributions of this paper are as follows. First, we show the power consumption can be reduced by uploading sensor data during the *user think time*. Second, we model the power consumption of the data transfer from the connected state and the disconnected state based on the measurement results with the smartphone and the state transition mechanism in the wireless communication. Third, we show PBTP can reduce more energy than TailEnder by simulation.

The rest of the paper is structured as follows. Section II reviews the related research about energy conservation technique in the wireless module and outlines our contribution. Section III presents PBTP protocol which achieves low power transfer of sensor data in participatory sensing. Section IV describes an energy model of data transfer and shows the energy-reduction effect of PBTP by simulation. Finally, Section V concludes this paper and presents the future work of this research.

II. PARTICIPATORY SENSING

Participatory sensing realizes a large-scale data collection by building sensor node functions on the mobile phone which senses and uploads the environmental information periodically. For example, Maisonneuve, et al.[6] gather noise data using the microphone of the mobile phone. Since mobile phone runs on a battery, the repetitive sensing and uploading operations require a new power-saving technique.

A new energy-reduction technique for participatory sensing needs to meet the following three requirements. First, it can

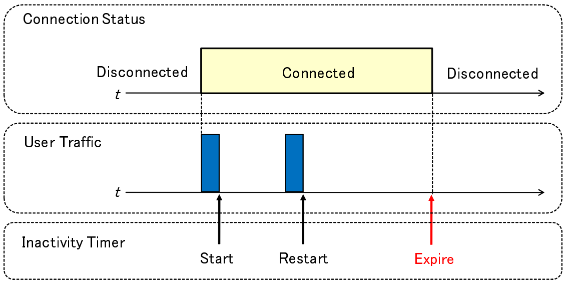

Fig. 1. The behavior in wireless communication of mobile phone

reduce energy consumption in any monitoring environment. Second, it can work on the existing mobile network because participatory sensing utilizes the user's mobile phone. Third, it does not interfere with the user's communication.

Wang, et al.[13] and Musolesi, et al.[14] propose energy-saving methods for participatory sensing. These proposals optimize the sensing and uploading operations depending on the application and achieve low power consumption. Wang, et al.[13] reduce power consumption on the sensing devices by limiting running sensors based on the context estimation. Musolesi, et al.[14] achieve low energy consumption by performing the context estimation on both the server and the mobile phone to reduce the number of communications. However, since these techniques strongly depend on the application, they cannot be applied to the participatory sensing applications which use other sensors. In order to achieve low power consumption regardless of the application, a new power-reduction technique is necessary at the communication protocol level.

A lot of research explores low energy-communication protocols for the wireless devices, especially in the field of sensor network[12] and wireless LAN[18], [19]. These proposals achieve low power consumption by controlling sleep and wakeup time of the communication device on MAC layer. For example, Krashinsky, et al.[18] realize low latency and low energy consumption by changing the polling interval of power-saving mode in accordance with the user's traffic characteristics. However, these techniques are not applicable to the mobile phone because mobile communication operators have strictly regulated the MAC layer protocol.

Balasubramanian, et al.[17] and Nurminen, et al.[20] reduce communication energy with the existing mobile phones. The communication device on the mobile phone consumes energy not only in transmitting data but also in keeping its wireless connection with the base station. Fig.1 shows the behavior of wireless communication in mobile phone. By maintaining its wireless connection with the network for some seconds after data transfer, a mobile phone reduces communication latency in successive requests. When no data transfer occurs during this period, inactivity timer expires and a mobile phone terminates its wireless connection. We define the period from the end of the data transmission to the expiration of the inactivity timer[15], [16] as the *inactivity timer period*. When intermittent communications happen, this mechanism leads a problem that the total time of *inactivity timer periods* increases. Balasubramanian, et al.[17] propose TailEnder protocol and shorten total time of *inactivity timer periods* by buffering communication requests and sending them together. However, when we apply TailEnder to participatory sensing, the throughput during user's operations decreases. Therefore, user operability becomes degraded. Nurminen, et al.[20] decrease the communication energy by sending data during the user's voice call. However, the average number of voice calls is 1.4 times per day in 2008 in Japan and tends to decrease year by year. Therefore, energy-reduction effect using their method is very limited

III. PIGGYBACK TRANSPORT PROTOCOL

To tackle the energy problems mentioned in Section.II, we leverage the wireless connection which the user keeps for using a communication application like web-browser. We focus on the *inactivity timer period* during the user's operations when the wireless connection is maintained but the data transfer does not occur. We achieve low energy consumption by sending sensor data during this period and call this energy-efficient protocol Piggyback Transport Protocol (PBTP).

A. *Inactivity timer period*

A cellular network allocates an uplink channel when a mobile phone requests to transfer data. When no data transfer occurs, mobile phone performs discontinuous reception (DRX) operation. DRX shortens the active time of the communication device and reduces energy consumption. The communication device changes the behavior by performing the state transition according to RRC (Radio Resource Control) protocol as shown in fig.2. In CELL_DCH (Dedicated Channel) state, mobile phone occupies a dedicated uplink channel. In CELL_FACH (Forward Access Channel) state, mobile phone uses a shared uplink channel. In this paper, we define CELL_DCH and CELL_FACH as the *connected* states. In CELL_PCH (Paging Channel) state, mobile phone has no uplink channel and performs DRX operation. In Idle state, mobile phone does not have RRC connection. In this paper, we define CELL_PCH and Idle as the *disconnected* states. RRC protocol controls the state transitions based on the report of traffic volume from the mobile phone and the inactivity timer. The number of the supported states and the timer values T_1, T_2, T_3 depend on the

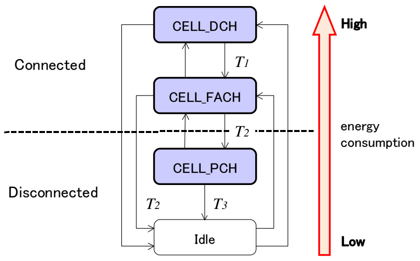

Fig. 2. RRC state transition

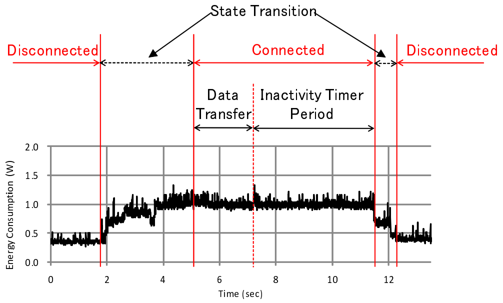

Fig. 3. Power consumption in sending 10 KB data

implementation and configuration of the wireless network. For example, Haverinen, et al.[21] set the inactivity timer values as followings. The default values of T_1 were 5 seconds for 8-32 kbits/s, 3 seconds for 64 kbits/s and 2 seconds for 128 kbits/s and faster. T_2 was 2 seconds and that for T_3 was several minutes or even ten of minutes.

In this paper, we define the period when the communication device keeps its wireless connection as the *inactivity timer period*. Fig.3 shows the power consumption in sending 10 KB data. In the measurement of this paper, we used Android Dev Phone 1[22], whose hardware components are the same with the commercial handset T-Mobile G1 and measured the power of the shunt resistance between the mobile phone and the battery with DL750[23]. The energy which is consumed through RRC state transition on the mobile phone contains not only the energy for data transfer but also the communication overheads caused by uplink channel assignment/release and the *inactivity timer period*. We measured the power for sending 10 KB data from the disconnected and the connected states with Android Dev Phone 1. We found that the communication overhead accounts for 4.40 J of the total power consumption of 6.22 J. This result shows that when a mobile phone sends sensor data every time getting environmental information, the communication overheads give a large impact on the energy resources in participatory sensing.

B. User think time

Crovella, et al.[24] clarify that the user's traffic has the *user think time* in which a user watches the screen and thinks about the next action on PC. Fig.4 shows the characteristic of the user's communication traffic on PC. Even in the user's consecutive operations, the communication device has the time when it processes communication and the time when it doesn't. Also on the mobile phone, the *user think time* is expected to happen during user's browsing operations.

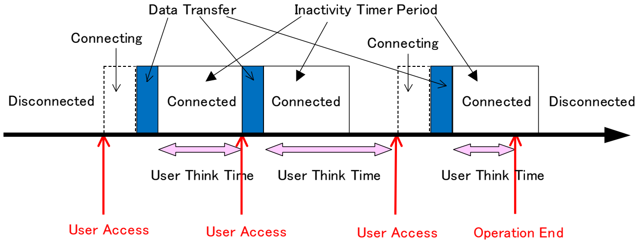

Fig. 4. The characteristic of user's traffic

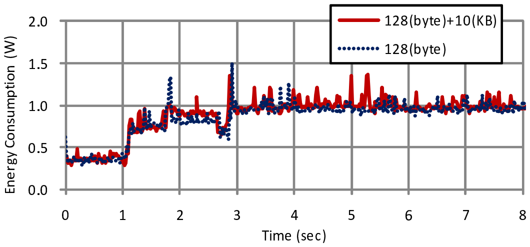

Fig. 5. The difference between single and continuous transmission

While a user browses websites, the communication device has the periods in which the wireless connection is maintained but data are not transferred. Fig.5 shows the power consumption when mobile phone uploads 128 bytes and when it uploads 128 bytes and 10 KB successively. The difference of these two waveforms is so little and the power consumption without data transfer is almost the same with that with data transfer. That is, if a mobile phone sends data during the *inactivity timer period* of the *user think time*, it can reduce energy consumption.

C. PBTP algorithm

Piggyback Transport Protocol (PBTP) achieves the energy-efficient upload by detecting *inactivity timer period* and the *user think time*. Fig.6 shows PBTP algorithm. PBTP has three states, namely "ON", "OFF" and "SLEEP", which the variable s represents. "ON" is the state when mobile phone is processing the user's communication request. "OFF" is the state when mobile phone is not processing the user's request. "SLEEP" is the state when the user does not operate a communication application. By changing upload operations according to the state s, PBTP reduces the power consumption. The variable q represents the size of sensor data in queue to be transmitted.

When sensor data occurs, PBTP waits for the generation of the user's communication request. When the user's request occurs, the state s transits to "ON". Afterward, when user's

```
$q$: the number of sensor data to be transmitted
$s$: the state of user communication
$D$: the deadline of transmission
while (true)
  wait ($q > 0$)
  repeat
    switch ($s$)
      case "ON": stop upload
      case "OFF": upload data
      case "SLEEP":
        if ($t < D$) stop upload
        else upload data
  until $q == 0$
```

Fig. 6. PBTP algorithm

request is completed, the state s transits to "OFF" and mobile phone starts uploading sensor data. When the user's request occurs again, the state s transits to "ON" and mobile phone stops uploading sensor data. With this operation PBTP does not degrade the user's operability. PBTP continues the above operations until $q = 0$. When detecting the end of the user operation, the state s transits to "SLEEP" and the mobile phone stops uploading sensor data. PBTP reduces the increase of the energy consumption through this operation.

If a mobile phone uploads sensor data only during the *user think time* and the *user think time* does not happen for a long time, the transmission delay increases. To resolve this problem, each sensor data has their deadline D. Each sensing application can set the deadline D to upload sensor data depending on the requirement. If the user's request does not happen for a long time and the current time t exceeds D, PBTP establishes a new wireless connection and upload all the data in queue.

We can achieve this PBTP algorithm by the following implementations. A mobile phone can transfer the user's request prior to the sensor data by the priority control method. For example, in the case of Android Dev Phone 1, we can set the traffic control conditions by adding the QoS function to the linux kernel and using *tc* command. A mobile phone can detect the occurrence of the user's request by monitoring the packet in the communication protocol stack. In the case of Android Dev Phone 1, the network device layer of the socket module can detect the occurrence of the user's request because it requests the network device driver to send the packets. A mobile phone can detect the end of the user's operation by monitoring the running applications or the state of LCD backlight. In the case of Android Dev Phone 1, we can detect the termination of the current application by using getRunningTasks() of ActivityManager and the LCD backlight off by receiving `ACTION_SCREEN_OFF` event from PowerManager.

IV. EVALUATION

A. Energy model

To evaluate the energy efficiency of PBTP, we model the energy consumption of the wireless data transfer in this section. We model the energy consumption with the linear equation which is often used for energy modeling.

The energy consumption in each upload can be represented with the energy consumption e_{init} in establishing wireless connection, $e_{send}(x)$ in sending data, $e_{inactive}$ caused by the inactivity timer period and e_{final} in terminating the wireless connection.

$$e(x) = e_{init} + e_{send}(x) + e_{inactive} + e_{final}$$

x is the data size to be sent in each upload. $T_{inactive}$ is the time of the *inactivity timer period*. Since $T_{inactive}$ is independent of the data size, we can consider it as a constant. That is, we can express $e_{inactive}$ as follow. W_{on} is the energy consumed when the wireless communication device is "ON".

$$e_{inactive} = W_{on}T_{inactive}$$

TABLE I

THE PARAMETERS OF ENERGY MODEL IN ANDROID DEV PHONE 1

parameter	value
e_{init}	1.295 J
c_0	0.025 J
c_1	0.134 sec
W_{send}	0.808 W
W_{on}	0.618 W
$T_{inactive}$	4.686 sec
R	5.123 KB/sec
e_{final}	0.302 J

We can represent $e_{send}(x)$ with the power W_{send} consumed when the wireless communication module is ON and with the size x to be transmitted.

$$e_{send}(x) = W_{send}(\frac{x}{R} + c_1) + c_0$$

R is the communication throughput for data transfer c_0 is the overhead of the upload time which occurs during the data transmission process. c_1 is the overhead of data size like a preamble in the data frame. We express the energy consumption $e_{disconnect}(x)$ in transmitting data from the disconnected state as follow.

$$\begin{aligned} e_{disconnect}(x) = e_{init} + W_{send}(\frac{x}{R} + c_1) + c_0 \\ +W_{on}T_{inactive} + e_{final} \end{aligned} \tag{1}$$

The energy consumption $e_{connect}(x)$ in sending data from the connected state does not contain the overheads by uplink channel assignment, *inactivitiy timer period* and uplink channel release. Therefore, we can express $e_{connect}(x)$ as below.

$$e_{connect}(x) = (W_{send} - W_{on})(\frac{x}{R} + c_1) + c_0 + W_{on}t \tag{2}$$

t varies depending on the relation among the *user think time* length T_{think}, inactivity timer length $T_{inactive}$ and the data transmission time $\frac{x}{R} + c_1$.

$$t = \begin{cases} 0 & T_{think} < T_{inactive} \quad (3) \\ T_{think} - T_{inactive} & T_{inactive} \le T_{think} < T_{inactive} \\ & +\frac{x}{R} + c_1 \quad (4) \\ \dfrac{x}{R} + c_1 & T_{inactive} + \frac{x}{R} + c_1 \\ & \le T_{think} \quad (5) \end{cases}$$

TailEnder sends all the requests in queue when a wireless connection is established. According to the equations (3), (4) and (5), the energy consumption varies depending on how much data x is transmitted and when the data is transmitted. For the large x, the energy consumption increases in proportion to x from the equations (4) and (5). Therefore, PBTP controls the transfer data size and the timing in order to make the most of the *user think time*. In this way PBTP controls the upload to increase the time represented in (3).

Table I shows the parameters which were obtained by the measurement in Android Dev Phone 1. e_{init}, $e_{inactive}$ and e_{final}, which are the overheads in establishing the wireless connection, are 1.295 J, 2.896 J, 0.302 J respectively. c_0, c_1,

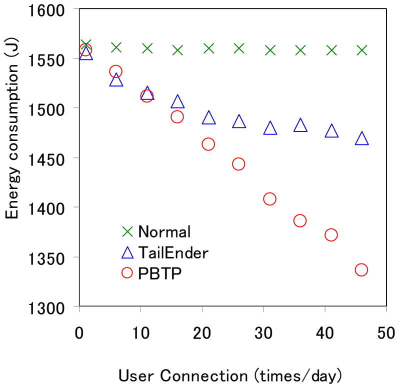

Fig. 7. Energy consumption versus user's communication number ($x_{sensor} = 100$)

W_{send}, W_{on} and R, which relate to the energy consumption for data transfer, are 0.025 J, 0.134 sec, 0.808 W, 0.618 W, 5.123 KB/sec respectively.

B. Simulation

In participatory sensing, the upload data size varies with the sensor data type like text or image, and the frequency of sensor data occurrence varies depending on the application. The frequency of the user's wireless communication, the number of user access during each connection and the *user think time* are different depending on the user's traffic pattern. In this section, we evaluate the energy-reduction effect of PBTP in the various sensor environments and the user traffic patterns by using the model obtained in Section.IV-A.

To simulate the energy consumption, we model the user traffic pattern and the monitoring environment as follows. For the parameters representing a user's traffic pattern, we define the user connection frequency r_{access} times/day, the number of user access during one wireless connection n_{access} times and the *user think time* t_{think} sec. r_{access} obeys Poisson process. t_{think} and n_{access} obey Pareto distibutuion[25]. For the parameters representing the monitoring environment, we define the data generation frequency r_{sensor} times/day and the sensor data size x_{sensor} KB. r_{sensor} obeys Poisson process and x_{sensor} is constant. We use the variable t_{detect} which expresses the time to detect the end of user's operation and evaluate the influence of t_{detect} on the energy-reduction effect. In order to compare with the existing techniques, we simulate the case which data transfer from the disconnected state happens 1 time per day and the case of TailEnder. If the user's request occurs at the time which is less than $\rho T_{inactive}$, TailEnder does not buffer the request and sends it to the server at once. In this simulation, we set $\rho = 0$ which is the lowest energy consumption case and the deadline D as 1 day in PBTP and TailEnder. About the parameters of the energy consumption model obtained in Section.IV-A, we use the variables of Table I.

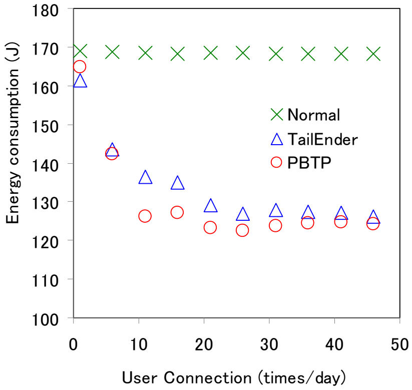

Fig. 8. Energy consumption versus user's communication number ($x_{sensor} = 10$)

Fig.7 shows the relationship between the energy consumption and the user connection frequency r_{access} where the sensor data size x_{sensor} is 100 KB. This result indicates that as r_{access} increases, the energy consumption for upload decreases because PBTP can send sensor data during the *user think time*. TailEnder also can reduce the energy consumption as r_{access} increases, but its energy-reduction effect is smaller than that of PBTP.

Fig.8 shows the relationship between the energy consumption and the user connection frequency r_{access} where the sensor data size x_{sensor} is 10 KB. Although the energy-reduction effect is smaller than the case of $x_{sensor} = 100$, PBTP provides the energy-reduction effect. In the large r_{access} area, the energy-reduction effect is small. This is because the number of the user access is large enough and both PBTP and TailEnder transfer almost the entire data during the *user think time*. When the number of the user's communication is extremely small ($r_{access} = 1$), the energy-reduction effect of PBTP is smaller than that of TailEnder. PBTP tries to

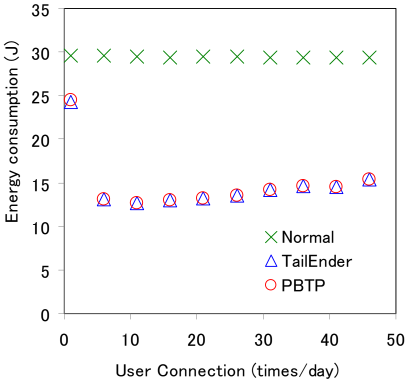

Fig. 9. Energy consumption versus user's communication number ($x_{sensor} = 1$)

upload sensor data in some user's connection. Therefore, when the number of the user's communication is extremely small, PBTP establishes the extra wireless connections caused by the deadline D and sends all the data in queue.

As is shown fig.9, when the sensor data size x_{sensor} is 1 KB, the difference between PBTP and TailEnder is small. Since the transfer data is small compared to the capacity of the *user think time*, both PBTP and TailEnder can transfer almost the entire data during the *user think time*. The energy consumption increases as the user connection frequency r_{access} increases. This is because the influence of the overhead c_0 represented as the equations (1) and (2) grows.

From the results above, PBTP provides a large energy-reduction effect when the data size and the number of the user's connection is large. The data size to be uploaded tends to grow in participatory sensing because the mobile phone collects the data periodically. This is especially the case for users with high browses and email applications as they require lower energy upload protocol. Therefore, PBTP is effective for energy-reduction in participatory sensing.

V. CONCLUSION

In this paper we propose a low power data transport protocol PBTP in participatory sensing. PBTP can upload sensor data more energy-efficiently than the existing TailEnder protocol. As such, we are now implementing PBTP protocol on the actual mobile phone.

REFERENCES

[1] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-Centric Urban Sensing," in *WICON, ACM*, vol. 220, no. 18, Aug. 2006.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Sensys, ACM*, Nov. 2006.

[3] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn, "The Rise of People-Centric Sensing," *Internet Computing, IEEE*, vol. 12, no. 4, Jul/Aug. 2008.

[4] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR: the personal environmental impact report as a platform for participatory sensing systems research," in *MobiSys, ACM*, Jun. 2009, pp. 55–68.

[5] E. Paulos, R. Honicky, and E. Goodman, "Sensing atmosphere," in *SenSys, ACM*, Nov. 2007.

[6] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "Noisetube: Measuring and mapping noise pollution with mobile phones," in *ITEE*, May. 2009, pp. 215–228.

[7] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," in *MobiSys, ACM*, Jun. 2009, pp. 165–178.

[8] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones," in *SenSys, ACM*, Nov. 2008, pp. 165–178.

[9] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The Bikenet Mobile Sensing System for Cyclist Experience Mapping," in *SenSys, ACM*, Nov. 2007, pp. 87–101.

[10] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application," in *SenSys, ACM*, Nov. 2008, pp. 337–350.

[11] M. Musolesi, E. Miluzzo, N. D. Lane, S. B. Eisenman, T. Choudhury, and A. T. Campbell, "The Second Life of a Sensor Integrating Real-World Experience in Virtual Worlds using Mobile Phones," in *HotEmNets*, Jun. 2008.

[12] G. Anastasi, M. Conti, M. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May. 2009.

[13] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition," in *MobiSys, ACM*, Jun. 2009, pp. 179–192.

[14] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A. T. Campbell, "Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones," in *Pervasive*, May. 2010.

[15] J. H. Yeh, J. C. Chen, and C. C. Lee, "Comparative analysis of energy-saving techniques in 3gpp and 3gp2 systems," *Vehicular Technology, IEEE*, vol. 58, no. 1, pp. 432–448, Jan. 2009.

[16] S. R. Yang and Y. B. Lin, "Modeling UMTS discontinuous reception mechanism," *Wireless Communications, IEEE*, vol. 4, no. 1, pp. 312–319, Jan. 2005.

[17] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications," in *SIGCOMM, ACM*, Nov. 2009, pp. 280–293.

[18] R. Krashinsky and H. Balakrishnan, "Minimizing Energy for Wireless Web Access with Bounded Slowdown," *Wireless Networks*, vol. 11, no. 1-2, pp. 135–148, Jan. 2005.

[19] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "802.11 Powersaving Mode for Mobile Computing in Wi-Fi Hotspots: Limitations, Enhancements and Open Issues," in *Wireless Networks*, vol. 14, no. 6, Dec. 2008, pp. 745–768.

[20] J. K. Nurminen and J. Noyranen, "Parallel Data Transfer with Voice Calls for Energy-Efficient Mobile Services," in *Mobiware*, Apr. 2009.

[21] H. Haverinen, J. Siren, and P. Eronen, "Energy Consumption of Always-On Applications in WCDMA Networks," in *VTC, IEEE*, Apr. 2007, pp. 964 – 968.

[22] *http://developer.htc.com/adp.html*.

[23] *http://tmi.yokogawa.com/products/oscilloscopes/scopecorders-oscillographic-recorders/dl750dl750p-scopecorder/*.

[24] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *Networking, IEEE*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[25] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *SIGMETRICS, ACM*, Jun. 1998, pp. 151–160.