MetroSense Project: People-Centric Sensing at Scale

Shane B. Eisenman,[†] Nicholas D. Lane,[*] Emiliano Miluzzo,[*]
Ronald A. Peterson,[*] Gahng-Seop Ahn,[†] Andrew T. Campbell[*]

[*]Computer Science, Dartmouth College, {campbell,niclane,miluzzo,rapjr}@cs.dartmouth.edu
[†]Electrical Engineering, Columbia University, {ahngang,shane}@ee.columbia.edu

Abstract

Looking forward 10-20 years we envision Internet scale sensing where the majority of the traffic on the network is sensor data and the majority of applications used every day by the general populace integrates sensing and actuation in some form. *Sensing will be people-centric*. On the other hand, nearly all published sensor network research over the last five years has focused on isolated, small scale testbeds designed for specialized applications (e.g., environmental sensing, industrial sensing, etc.) of interest to engineers and scientists. We believe the gap between the state of the art and our future vision can be bridged through the development of a new wireless sensor edge for the Internet. To this end, in the MetroSense Project we are developing a general purpose sensing infrastructure capable of realizing a wealth of sensing applications with mass appeal for producers and consumers of sensed data. In this paper we motivate the need for a new architecture to support people-centric sensing at Internet scale, outline our MetroSense architecture [1], and highlight our progress to date in designing and deploying prototype implementations of the MetroSense architecture via the deployment of our campus area sensor network.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless Communications.

General Terms: Design.

Keywords: Applications, System design.

1 Motivation

Today's wireless sensor networks do not play a role in the every day lives of the general populace, despite the fact that many key enablers are in place. The era of the small cheap embedded devices capable of integrated computation, sensing, and wireless communications is here. Over the last five years there have been significant advances in applying self-organizing multihop wireless sensor networking technology to a wide variety of environmental monitoring applications giving scientists unprecedented access to the physical world. However, we believe that these specialized multihop networks are unlikely to provide a foundation for a large-scale general purpose Internet sensing presence. To achieve a wireless sensor edge for the Internet with true popular appeal, a new architectural approach is needed that not only supports very large scale, but focuses on people-centric sensing. In people-centric sensing applications, people are not only both the producers and consumers of the sensed data, but also enable extended sensing and communication opportunities through their mobility.

A focus on people-centric sensing shifts important design challenges away from those that apply to static, specialized, highly embedded networks. Some requirements are more strict. Because people are mobile, networks of sensing platforms (e.g., cell phones, motes-class devices) carried in large part by humans or human driven vehicles must deal with the challenges of mobility. Since the goal is a general purpose network, and because there is no centralized administrative control over all potential network users, platform heterogeneity is a certainty, and software version control is an issue. On the other hand, other requirements are relaxed. Due to mobility, it is no longer a strict requirement to use a wireless multihop transport to deliver data from the sensing target to information sinks. Also, many sensing devices will be carried by humans rather then embedded in the environment. This loosens the energy constraint as the devices can periodically be recharged *á la* cell phones.

In order to integrate heterogeneous sensing platforms into a single architecture, a common set of interfaces must be supported on each platform. At minimum, these include basic communication primitives related to sensing like tasking and collection. In a mobile environment, the fact that devices are likely to have widely varying radio ranges (e.g., cellular/WiMax, WiFi, Zigbee), strongly impacts the implementation of these communication primitives. A network of mobile Zigbee radio devices has sparse connectivity and must therefore use prediction-based tasking and multihop routing or muling for data collection. Conversely, a network of devices nearly always connected to a cellular backhaul can do “just in time” tasking and one hop data upload. However, it is important to note that regardless of the communication range of the device, the sensing range for a given sensing modality remains the same, and thus sensing itself still occurs opportunistically based on uncontrolled mobility. This interplay between sensing and communication ranges has implications on the design of emerging sensor network architectures.

An open question is what people-centric sensing applications will emerge as dominant in the next 10-20 years. We believe the following three classes of sensing are supported by an Internet-scale people-centric architecture: *personal sensing*, *peer sensing*, and *utility sensing*. Personal sensing has the individual as the object of sensing as well as the consumer of the sensed data, and includes applications like Nike^{+} [8], sensor-enabled cellphone applications [6], and health-related sensing (e.g., [9] [12]). Peer sensing has groups of people as its focus, and includes applications like urban gaming [11], and other ap-

WSW'06 at SenSys'06, October 31, 2006, Boulder, Colorado, USA.

plications with peer to peer information sharing. Utility sensing offers system-wide (e.g., college campus, city borough, metropolitan area) utility to a large population of potential users, and includes applications like noise mapping [10] and pollution mapping.

Clearly, these applications will have varying requirements in terms of timeliness of delivery, sampling rate, etc. For example, some applications (e.g., patient monitoring [9]) require real-time or near real-time sensing and data upload. For these, near ubiquitous coverage via dense WiFi APs or cellular/WiMax infrastructure is necessary. On the other hand, some applications (e.g., Nike$^+$ [8]) can tolerate hours or even days of data delivery delay, and do not require heavy on-sensor processing. For these applications a network of devices with mote-class (e.g., Tmote Invent [4]) radios is sufficient; very cheap sensor/radio devices will be produced in bulk and sewn into clothing and implanted in footwear; delay-tolerant/sparse-connectivity networks can be used for tasking and collection of data.

In the initial stages of the MetroSense Project, we have started looking at the delay-tolerant sensing application space, using mote-class sensing platforms. In Section 2 we discuss the MetroSense architecture [1], which supports heterogeneity but is designed to address the communication limitations of mote-class devices, allowing us to investigate an "opportunistic sensor networking" paradigm. We present our work to date on deploying a campus area sensor network testbed in Section 3, including two people-centric applications that contain both personal sensing and peer sensing elements. Section 5 summarizes our vision and current progress.

2 MetroSense Architecture

The MetroSense architecture incorporates a set of design principles, a hierarchical network structure, and a collection of useful techniques for enabling the addition of a sensor edge to the existing Internet infrastructure. The MetroSense architecture makes possible the realization of large scale, people centric sensing at low cost focusing on using mobile sensors and integrating select concepts from WSNs, mobile networking, and the scalability and reach of the Internet to enable new applications centered on data gathered from and around people. In the following we give an overview of the MetroSense architecture (described fully in [1]), starting with a set of overarching principles that guide the design of the architecture, promoting low cost, scalability, and performance.

2.1 Guiding Principles

MetroSense is governed by a number of design principles: *(i) Network Symbiosis*: New sensing infrastructure and service deployment should leverage existing traditional networking infrastructure and services. The symbiosis between networks should be managed to maximize the benefit to the participants of all associated networks. A sensor network architecture can benefit from the existing power and communications physical infrastructure, and from existing network services such as routing, reliable transport, and security. Users of the established networks should experience minimal service degradation (e.g., through resource sand boxing) and may be provided service enhancements due to the association. *(ii) Asymmetric Design*: Resource asymmetry that exists among members of the sensor network should be exploited by pushing computational complexity and energy burden to more capable nodes, while maintaining flexibility in the sensing applications that can be supported. Leveraging resource asymmetry may result in sub-optimal process flow in the provision of a particular service or operation. However, we are willing to accept this sub-optimality for the benefits of a simplified service model and a network that is easier to manage. *(iii) Localized Interaction*: Network elements should possess a highly constrained "sphere of interaction" within which they communicate with other network elements. We believe the loss of flexibility imposed by requiring localized interactions is outweighed by the increase in service implementation simplicity and communications performance. MetroSense relies on a probabilistic notion of reachability via opportunistic (mobility-enabled) interactions between nodes in the field.

2.2 Architectural Overview

The following outlines a three tier physical architecture for MetroSense based on a minimum required set of capabilities at each tier, followed by a discussion of *opportunistic sensor networking*, the conceptual foundation of the MetroSense architecture that allows for people-centric, very large scale sensing at a reasonable cost. A more detailed description of the MetroSense physical architecture, and a discussion of the MetroSense software architectural components exists in [1].

2.2.1 Tiered Physical Architecture

Server Tier. Members of the Server Tier are Ethernet-connected servers equipped with practically unbounded storage and computational power. These generic servers provide important service support to the architecture in the form of a set of *core components* and a set of *common components*. Core components provide management oversight and support to the tasking and collection activities within a particular MetroSense administrative domain, and include admission control, system state repository, localization and synchronization, and ground truth sensing. Common components are communal assets used to store and process the data output of sensing applications and are shared across MetroSense administrative domains. These include the spaces directory system, sensor data repository, sensor data mining, and sensor data anomaly detection.

SAP Tier. A sensor access point (SAP) offers high performance, high reliability, and secure gateway access to MetroSense services for sensor tier elements while in range. SAPs provide secure, trusted interaction with the sensor tier. When sensor tier elements are not under a SAP then there are little or no such assurances given. A SAP performs the role of sensing, acts as a sink point for data gathered by the sensor tier and programs sensors by loading small application components onto them. Following the design principle of network symbiosis, we envision many SAP implementations will exploit readily available infrastructure such as WiFi access points, PCs/laptops, cell phones [6].

Sensor Tier. A mobile sensor (MS) is a wireless sensor device entrusted to a custodian, such as a person or a vehicle. The sensor device performs application functions as the custodian moves within the sensor field. A static sensor (SS) is a wireless sensor device placed at a fixed location in the sensor field, typically to instrument infrastructure such as machinery or at specific locations to extend SAP coverage where wired SAPs are not possible. Mobile sensors support sensing of the sensor custodian via applications run on behalf of the custodian or others, and provide sparse sensing via mobility. Examples of sensors in MetroSense are Zigbee-compatible motes [4] and sensor-equipped cellular phones [6] supporting a common protocol stack.

2.2.2 Opportunistic Sensor Networking

In MetroSense, we leverage the uncontrolled mobility patterns of mobile sensors (i.e., human and vehicular mobility), mobility that comes at no cost to the sensing/communication infrastructure, to bridge gaps in static sensor coverage. This mobility gives rise to a suite of opportunistic processes that facilitate urban-scale sensing.

Opportunistic Tasking, Sensing, and Collection. In order for a given sensing operation to be successful it is necessary that a particular sensor has the right instruments (e.g., temperature sensing device) for the required sensing task, is loaded with the correct application, and has mobility characteristics that bring the sensor within the target area during the time window of interest. In an environment like MetroSense where most interaction between nodes is based on uncontrolled mobility we term the situation where the aforementioned requirements are met as *opportunistic sensing*.

Tasking of an appropriate mobile sensor for a given application by a particular SAP requires both that the mobile sensor moves within the sphere of interaction (e.g., radio range) of the tasking entity (e.g., a SAP, a mobile or static sensor that has been delegated the responsibility to task), and that the sensor remains within this sphere long enough for the tasking packet transfer to complete. Similarly, upload of a mobile sensor's data to a particular collection point (e.g., a SAP, a sensor acting as a data mule, a sensor with a multihop path to a SAP, a sensor that is in network end point or aggregator of particular data) requires both that the mobile sensor moves within the sphere of interaction of the collection point, and that the sensor remains within this sphere long enough for the data upload to complete. We introduce the terms *opportunistic tasking* and *opportunistic collection* to refer, respectively, to the methods by which sensor tasking and data upload can be completed in the face of uncontrolled sensor mobility.

Opportunistic Delegation Model. In any network architecture, principals have a set of designated responsibilities. Opportunistic delegation refers to the limited transfer of a subset of these responsibilities, when the transfer yields some advantage. Sensors may delegate responsibilities related to sensor tasking, and data collection. The transfer is limited in the sense that responsibilities are only delegated for a limited time or to perform a specific objective. The opportunistic element of the delegation is introduced by sensor mobility.

To make the notion more concrete, the following example presents a scenario where opportunistic delegation is used to extend the effective sensing range of a static sensor. We use the following as a running example through the rest of this section. Suppose an application requires data of type γ from region A of the field in the time interval $[t_1, t_2]$. Suppose, by some previous assignment, static sensor y has the responsibility to acquire this data, but its γ sensor range is such that region A is out of range. However, a mobile sensor z possessing a γ sensor exists with a motion vector v_z that intersects region A in the time interval $[t_1, t_2]$. Ideally z rendezvous with y prior to the intersection of v_z with A. In such a case, y delegates the target sense responsibility directly to z. Otherwise, “indirect delegation” can be used, whereby y delegates a third sensor w with the responsibility to task an appropriate sensor to execute the sensing. w, which can be a mobile or static sensor, in turn delegates the responsibility to sense the target to mobile sensor z. Such indirect delegation chains can grow as long as required. Assuming z is delegated to do the sensing in region A in time, it will acquire the γ data. At this point it is the responsibility of z to return the sensed data to the collection entity, say y. This responsibility can be fulfilled by z itself, or delegated to other sensors in a manner similar to the sensing responsibility just described.

Sensing Coverage. Opportunistic delegation provides only a probabilistic notion of sensing coverage, due to its strong dependence on both the temporal and spatial aspects of mobility. In particular, in the context of the running example in this section, a suitable sensor z (γ sensor equipped, v_z intersects A) for delegation must be available to y at the right time (relative to the intersection time of v_z with A) for the sensing delegation to be successful. Then, the data must make the return trip from z to the collection point, possibly using delegation.

Sensor Selection. In the previous example, sensor y may have several candidate sensors $\{z_1, \ldots, z_n\}$ for delegation of a given sensing task. In the general case, characteristics of the candidate sensor are not known to y, and consequently y may choose its delegate poorly. However, if y is allowed to choose more than one delegate, the success probability of the objective goes up. In fact, when such multiple delegation is combined with delegate chaining, a larger platoon of sensors can be assigned any particular objective, increasing the fault tolerance of the data sensing and collection effort, if desired. A more typical case might involve a constant or linearly growing platoon.

Sensor Data Collection. Opportunistic delegation in the data collection process, though semantically different, is functionally equivalent to data muling. Delegation in the collection process may enable collection where it is otherwise impossible (e.g., an isolated sensor). The best collection delegate at a given point is difficult to determine, yet the choice strongly impacts the ultimate success of data collection.

Sensor Data Fidelity. The impact of opportunistic

Figure 1. **A SAP implementation using a AP-70 WiFi access point, with a Moteiv Tmote Invent attached via USB interface.**

delegation on the fidelity of the sensed data is related to the sensitivity of the sensing instruments (e.g., motion detector range) on the sensor. Mobile sensing delegates may be forced, due to mobile sensor custodian mobility, to acquire data at a distance from the intended target beyond the optimal sensitivity threshold of the sensing instrument. In this case the quality of the sensed data often decreases monotonic with the sensor's distance from the target. Static networks are often deployed to provide complete sensing coverage over all regions of interest. However, this is not feasible at urban scale.

3 MetroSense Campus Area Sensor Network

In the following, we discuss our current progress in designing and deploying a campus scale sensor network based on the MetroSense architecture. We are using this testbed as a prototype to explore the challenges of large scale, sparsity, and mobility, as well as practical systems issues related to platform durability, maintenance, and component failure. We start with the design and current state of our SAP tier deployment, followed by a description of two applications we are currently studying.

3.1 SAP Design and Deployment

The Dartmouth College campus is gradually being covered by a network of SAPs that together with other mote class devices, that will typically be attached to people, will form a general purpose sensing infrastructure in support of a variety of sensing applications. The bulk of SAPs within this deployment will be created by leveraging the existing campus WiFi infrastructure. Standard Aruba AP-70 devices will be augmented with mote sensors (such as the Tmote Invent) via a USB connector (as seen in Figure 1) which provide both a network of static sensors and the necessary transceiver to connect mobile sensors with the back end components of MetroSense. The host Aruba AP-70s will continue to operate as normal WiFi APs due to the platform being open with the adoption by Aruba of the Open-WRT Linux distribution. This allows custom software to be added to extend the functionality of these devices to support the dual roles of acting as an Aruba WiFi AP and as a MetroSense SAP.

As this surrogate WiFi network will be the same network used for day to day operations, the SAP network will be highly representative of a realistic MetroSense deployment. Just as would be required for a MetroSense deployed in less controlled non-research institution circumstances, we will be forced to build both protection for device and network resource allocation such that network operations are not made unnecessarily brittle nor performance is not noticeably retarded. Similarly the potential for personal data to be collected by the MetroSense along with the shared nature of network and it's devices requires a means of providing verifiable guarantees that MetroSense collected data as well as control exchanges are sufficiently isolated from the campus network traffic, as well as it's users and administrators.

Figure 2. **An instrumented SkiScape. Dots mark fixed trail sense points. Squares mark collection points at the mountain base.**

Once complete the 800 SAP network will cover not only academic spaces but campus stores, the gymnasium, town streets, the town museum, and the medical school. This will offer the potential for a plethora of applications for use by faculty, students, employees, administrators, and townspeople. This deployment will present one of the largest general purpose, long-lived sensor networks built to date.

3.2 Current Applications

The bulk of non-academic/experimental wireless sensor network penetration to date has been into areas of environmental and industrial monitoring where people are out of the loop. Recreational sports is a domain where the benefits provided by sensor networks will spark a more general interest in wireless sensor network technology. In this section, we highlight two applications, SkiScape [2] and BikeNet [3], we are studying in the MetroSense Project as drivers to study the issues of people-centric sensing. SkiScape is an application for downhill ski resorts focusing on gathering semi-regular trail condition data for immediate feedback to the skier population, and also tracking skier mobility to enable both real-time response, and long-term trace analysis. With BikeNet, we investigate sensing and networking issues in deploying a sparse sensing system for cyclist experience mapping.

3.2.1 SkiScape Sensing

Skiers are interested in knowing current trail conditions (e.g., ice, bare spots, congestion) when at the base of the mountain in order to determine which lift to use to get to the desired trail head at the top of the mountain. Resort managers are interested in learning skier flow statistics to estimate wear on the terrain in order to en-

Figure 3. **Bicycle in the BikeNet testbed outfitted with sensor to measure pedal speed, wheel speed, road/trail slope angle, and compass direction.**

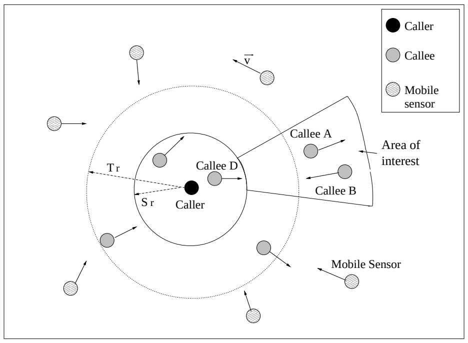

Figure 4. **The *Caller* is able to extend its effective range beyond its physical sensing range by means of opportunistic delegation.**

act preventative maintenance (e.g., close trail, make artificial snow). Safety/emergency personnel are interested in tracking skiers' location and speed in case of accidents (e.g., fall off trail, avalanche), and also to prevent accidents by speed policing. Skiers may be interested in tracking their own location or the location of their friends on the mountain, as well as avoiding lifts with long queues.

We are inspired by the resemblance of a ski resort trail map to a static sensor network data dissemination tree; many trails with heads at the top of the mountain funnel towards a small number of lift entry/collection points at the base of the mountain (Figure 2). In the SkiScape, ski lifts provide a continuous supply of data mules (skiers) to the trails at no cost to the sensing/communication infrastructure. Static sensors, mounted on light poles, sense data about the adjacent trail area; mobile sensors, mounted on skiers, can collect data in their locality as the skier traverses the mountain. Skiers opportunistically collect/carry data of interest as they travel along the trails to the data sinks at the base. In this way we leverage a sparse deployment of both static and mobile sensors to give a more complete picture over time of the field of interest at lower cost than would be required with a fully static deployment.

3.2.2 BikeNet Sensing

There is substantial interest in the cycling community in collecting data quantifying various aspects of the cycling experience. Existing commercial products have begun to integrate data from multiple local sensors (including biometric sensors and a GPS receiver) on a single user display, and even provide map software. A limitation of the currently available products is the inability to share data with other riders in real-time. Further, often real-time performance analysis of locally collected data is limited to local display of simple statistics like min, max and mean over the entire trip. When the road terrain is highly non-uniform and uphills can last a long time, comparing current speed against a trip-wide average loses significance.

With the BikeNet application, we work to address these limitations by allowing cyclists to share information about themselves and the paths they mutually traverse for real-time display. In BikeNet, information sharing occurs via short range radios, and can be direct (i.e., bike-to-bike) or indirect through neutral third-party entities called *rocks*. Rocks are untethered storage and aggregation devices that are placed along roads and trails frequented by cyclists; they store location-specific performance data on per-cyclist and aggregate bases.

In addition to sensor data sharing between cyclists, in BikeNet we investigate resource sharing between two or more bikes traveling within a common local radio range of each other. Each bicycle is equipped with a subset of sensors connected to form a wireless bike area network. Using localized resource sharing, all members of a group can benefit from the most capable and most expensive sensors attached to any member of the peloton. A location-stamped synchronization beacon can be broadcasted to the group from a cyclist outfitted with a GPS sensor. The short range radio confines the propagation of the beacon to an area on the same order of the positioning uncertainly of most price-accessible handheld GPS units, maintaining the usefulness of the broadcast for all recipients. Further, a high power long range radio mounted to one bike can be used for high priority transmissions (e.g., very low rate cycle tracking messages, medical emergencies).

3.3 Application Enablers

In the MetroSense Project, we are investigating a number of specific primitives to which the general paradigm of opportunistic delegation gives rise. These primitives can be viewed as virtual services [1] enhancing the sensing and collection capabilities of SAPs and mobile sensors. In the following, we highlight recent progress on extending the physical sensing range of a device to cover an arbitrary area using opportunistic delegation. We call this virtual sensing, and refer to this extended sensing range as a node's virtual sensing range (VSR) [7]. In the following we briefly describe VSR with reference to Figure 4, where we name the fixed nodes or gateways, *callers* and the mobile nodes *callees*; the transmission range T_r and the sensing range S_r are also shown. The caller is the sensor which has been tasked to provide sensing about its surrounding environment. VSR exploits the possibility that mobile sensors pass by the location of a caller.

The caller, which announces its presence by periodically beaconing, runs a selection algorithm to pick the best subset of mobile nodes to task for sensing beyond its S_r. The metric by which the caller selects mobile nodes is based upon two factors: *i)* the Link Quality Indicator (LQI) of mobile nodes' periodic beacons which start being delivered as soon as the caller's radio range is crossed, and *ii)* readings from a compass mounted on each node, piggybacked in each beacon the mobile node sends to the caller. The caller, which is also equipped with a compass, selects nodes heading toward the area of interest among all the mobile sensors in its radio coverage. As soon as the LQI of a candidate mobile sensor starts decreasing, indicating that the candidate is leaving the caller's radio range, the caller tasks the candidate if the candidate is still moving in the required direction. If, for example, in Figure 4 callee D is moving from left to right, callee D is a good candidate because it is heading toward the area of interest. As soon as callee D approaches the caller's radio coverage edge, callee D's LQI computed by the caller starts decreasing. At this point the caller tasks callee D to start collecting sensed samples.

The callee takes samples and stores them in the flash until the distance specified by the caller has been covered. The callee probabilistically relies on other mobile nodes heading in the opposite direction to mule data back to the caller once out of the caller's radio range. In Figure 4, callee A tasks callee B to provide sensing along its path toward the caller and, at the same time, callee A passes the data collected so far to callee B. Once within the caller's radio range, callee B delivers its own sensed data and callee A's sensed data to the caller. Through opportunistic delegation [1] and muling we have virtually extended the caller's sensing range to the area of interest, indicated in Figure 4; this area is beyond the caller's physically limited sensing range. VSR becomes more useful as the density of mobile nodes per time unit area increases. For example, in a dense urban environment VSR could be leveraged to reduce the cost compared to deploying a static sensor network infrastructure.

4 Related Work

There are a small number of academic and industry projects currently looking at implementing large scale and/or people-centric sensor networks. The TENET project at CENS [13] proposes an architecture for tiered sensor networks focused on leveraging device heterogeneity to promote scalability and simplicity of design and deployment. The Urban Sensing project at CENS [14] seeks to develop cultural and technological approaches for using embedded and mobile sensing to invigorate public space and enhance civic life. The goal of the CarTel project at MIT CSAIL [15] is to design and deploy a mobile distributed sensor computing system, specifically focusing on providing a simple application programming interface, handling intermittent and variable network connectivity, and handling data heterogeneity. SensorPlanet [6] is a Nokia-initiated cooperative project focused on building an open global mobile device centric research platform for large-scale wireless sensor network research. The Hourglass project at Harvard [16] is building a scalable, robust data collection system via Internet-based infrastructure to support geographically diverse sensor network applications. The ExScal project at Ohio State [17] is investigating issues confronting the deployment and operation of extreme scale static wireless sensor networks.

5 Conclusion

Future sensor network architectures should support people-centric applications as their focus, at very large scale, leading to a shift in the set of applicable design considerations. A people-centric architecture must accommodate and leverage uncontrolled human mobility, while at the same time power constraints are lessened since the person can recharge the sensing platform occasionally. Platform heterogeneity is a certainty, implying widely varying radio characteristics, but the sensing range is independent of the communications range; achieving sensing coverage will continue to be a challenge. Ongoing work in the MetroSense Project is studying an opportunistic sensor networking [1] approach to people-centric sensing, with the goal of providing a new wireless sensor edge for the Internet.

Acknowledgment

This work is supported by the Institute for Security Technology Studies at Dartmouth College under the 2006 appropriation funds (the NIST grant) award 60NANB6D6130 from the U.S. Department of Commerce, Intel Corp, and Nokia. The statements, findings, conclusions, and recommendations are those of the authors and do not necessarily reflect the views of the National Institute of Standards and Technology (NIST), the U.S. Department of Commerce, Intel, or Nokia. Shane Eisenman is currently supported by the Army Research Office (ARO) under Award W911NF-04-1-0311.

6 References

[1] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, and R. Peterson. People-Centric Urban Sensing (Invited Paper). In *Proc. of IEEE/ICST/CREATE-NET 2nd Int'l Conf. on Wireless Internet*, Boston, Aug 2006.

[2] S. B. Eisenman and A. T. Campbell. SkiScape Sensing (Poster Abstract). In *Proc. of ACM 4th Int'l Conf. on Embedded Networked Sensor Systems*, Boulder, Nov 1–3 2006.

[3] S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, G. S. Ahn, and A. T. Campbell. BikeNet: An Opportunistic Sensing System for Cyclist Experience Mapping. In *submission*.

[4] Moteiv Tmote Invent. http://www.moteiv.com/.

[5] TinyOS. http://www.tinyos.net/.

[6] Sensorplanet. http://www.sensorplanet.org/.

[7] E. Miluzzo, N. D. Lane, and A. T. Campbell. Virtual Sensing Range (Poster Abstract). In *Proc. of ACM 4th Int'l Conf. on Embedded Networked Sensor Systems*, Boulder, Nov 1–3 2006.

[8] Nike^{+}. http://www.nikeplus.com.

[9] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. Codeblue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. In *Proc. of Int'l Workshop on Wearable and Implantable Body Sensor Networks*, April 2004.

[10] Noise Mapping England. http://www.noisemapping.org/.

[11] Come Out and Play Festival 2006. http://www.comeoutandplay.org/.

[12] A. Wood, *et al*. ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring. Technical Report CS-2006-11, Dept. of Computer Science, Univ. of Virginia, 2006.

[13] TENET: An Architecture for Tiered Sensor Networks. http://research.cens.ucla.edu/projects/2006/Systems/Tenet/.

[14] Urban Sensing. http://research.cens.ucla.edu/projects/2006/Systems/Urban_Sensing/.

[15] CarTel. http://cartel.csail.mit.edu/.

[16] HOURGLASS: An Infrastructure for Connecting Sensor Networks and Applications. http://www.eecs.harvard.edu/ syrah/hourglass/.

[17] ExScal: Extreme Scale Wireless Sensor Networking. http://ceti.cse.ohio-state.edu/exscal/.