Open access • Proceedings Article • DOI:10.1145/1322263.1322273

The BikeNet mobile sensing system for cyclist experience mapping — Source link

Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald Peterson ...+2 more authors

Institutions: Columbia University, Dartmouth College

Published on: 06 Nov 2007 - International Conference on Embedded Networked Sensor Systems

Topics: Systems architecture and User interface

The BikeNet Mobile Sensing System for Cyclist Experience Mapping

S. B. Eisenman,[†] E. Miluzzo,[*] N. D. Lane,[*] R. A. Peterson,[*] G-S. Ahn,[†] A. T. Campbell[*]
[*]Computer Science, Dartmouth College, {campbell,miluzzo,niclane,rapjr}@cs.dartmouth.edu
[†]Electrical Engineering, Columbia University, {ahngang,shane}@ee.columbia.edu

Abstract

We describe our experiences deploying BikeNet, an extensible mobile sensing system for cyclist experience mapping leveraging opportunistic sensor networking principles and techniques. BikeNet represents a multifaceted sensing system and explores personal, bicycle, and environmental sensing using dynamically role-assigned bike area networking based on customized Moteiv Tmote Invent motes and sensor-enabled Nokia N80 mobile phones. We investigate real-time and delay-tolerant uploading of data via a number of sensor access points (SAPs) to a networked repository. Among bicycles that rendezvous en route we explore inter-bicycle networking via data muling. The repository provides a cyclist with data archival, retrieval, and visualization services. BikeNet promotes the social networking of the cycling community through the provision of a web portal that facilitates back end sharing of real-time and archived cycling-related data from the repository. We present: a description and prototype implementation of the system architecture, an evaluation of sensing and inference that quantifies cyclist performance and the cyclist environment; a report on networking performance in an environment characterized by bicycle mobility and human unpredictability; and a description of BikeNet system user interfaces. Visit [4] to see how the BikeNet system visualizes a user's rides.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless Communications; J.3 [Life and Medical Sciences]: Health.

General Terms: Design, Experimentation, Performance.

Keywords: Applications, Bicycling, Recreation, Systems.

1 Introduction

There is substantial interest in the mainstream recreational cycling community in collecting data quantifying various aspects of the cycling experience, mirroring the broader interest in fitness metrics among exercise enthusiasts and other health conscious individuals. Existing commercial bike-sensing systems targeting this demographic measure and display simple data such as wheel speed, and provide simple inferences such as distance traveled and calories burned. These systems have become increasingly more sophisticated and miniaturized. This trend is continuing and bicycles in the future will be sold with embedded fitness/performance-related sensing systems. However, the data collected by current systems fails to capture a more comprehensive picture of the cyclist experience. Among recreational cyclists there is a spread in the level of interest about various characteristics of a ride. Some are competitive with their friends for the sake of bragging rights, and may want to initiate challenges to set up virtual competitions among geographically separated cyclists; some focus on health-related aspects such as personal fitness; many view bicycling as a time to relax while getting some moderate exercise and are most interested in finding routes that are safe and quiet; others want to simply archive statistics about their rides for later analysis [5]. In this paper, we design and implement the prototype of a system not only to give context to the cyclist performance as part of a user-targeted application (e.g., health and safety), but also to collect environmental data as part of communal projects (e.g., pollution monitoring/mapping). We quantify aspects of cycling performance and environmental conditions that the mainstream recreational cyclist can appreciate and afford, akin to the Nike+iPod kit, a system [6] for recreational runners that logs exercise history.

BikeNet represents the first working mobile networked sensing system for bikes. Contributions of our work include:

- **Disconnected Operation.** BikeNet utilizes an opportunistic networking paradigm, whereby mobile sensing platforms are tasked and data is muled or uploaded according to the opportunities that arise as a result of the uncontrolled mobility of the cyclists. The BikeNet system operates in a *delay tolerant sensing* mode by default, where cyclists go on trips, collect sensed data, and upload their data when they return to home, possibly using the assistance of data mules (as discussed in Section 2.2.3). This default mode is akin to the Nike+iPod [6] sensing system for runners. In this case, BikeNet represents a pure delay tolerant mote-based solution. However, if the cyclist carries a cell phone (such as the Nokia N80 in our implementation) BikeNet automatically integrates the cell phone into the system as a mobile sensor gateway (i.e., a mobile sensor access point (SAP)) and offers real-time interaction between the back end and the cyclist in support of *real-time sensing*.
- **Cyclist Performance/Fitness Measurement.** The system collects and stores data about the following baseline cycling performance metrics: current speed, average speed, distance traveled, calories burned. In addition, the system collects and

SenSys'07, November 6–9, 2007, Sydney, Australia.

stores the following advanced metrics: path incline, heart rate, galvanic skin response (a simple indicator of emotional excitement or stress level). All data sensed by the system is at least stamped with time and location metadata.

- **Environment/Experience Mapping.** The system provides quantitative guidance to cyclists about the healthiness of a given route in terms of pollution levels, allergen levels, noise levels, and roughness of the terrain. These measurements, together with data from cyclist performance measurements, are correlated to create a holistic picture of the cycling experience. This environmental data is also provided to the larger community.

- **Long Term Performance Trend Analysis.** Collected data persists beyond the ride on which it is collected. The system enables the upload of data traces into a personal repository that can be selectively shared with other individuals, or into a public database. The data is archived in such a way as to facilitate spatio-temporal trend analysis.

- **Data Collection and Local Presentation.** BikeNet allows the cyclist to customize, via a profile of preferences, what data is collected by the system, when it is collected, where it is collected, and under what correlated conditions sensor data capture occurs (e.g., increase the sampling rate of the heart rate when the path incline is above a threshold). The profile also indicates how data is to be presented, both locally (e.g., on a handle bar-mounted cell phone LCD) on the bicycle when *en route* and through access and presentation methods once the data has been delivered to the back end repository.

- **Data Query and Remote Presentation.** The system provides a web-based portal [4] on the back end as a means to inject queries into the system to request particular bicycling-related data of interest to the back end system user. Also, the portal can be used as a place to publish/share data with friends/competitors about themselves and the paths they traverse for real-time or delayed display. In so doing, we provide a useful tool to network members of the cycling community through data of mutual interest.

In the following sections, we describe our experiences deploying a sensing system for cyclist experience mapping, leveraging opportunistic sensor networking principles and techniques [22]. We discuss the system architecture, design, and implementation in Section 2. Section 3 describes our cyclist experience mapping application, including sensing accuracy and inference techniques, communication protocol performance, and feasibility results. Related work is discussed in Section 4 before concluding with a summary and a discussion of possible extensions in Section 5.

2 System Architecture and Design

BikeNet is a network characterized by mobile sensing and sparse radio network connectivity. Given these characteristics, and the application requirements for the system, we design the BikeNet system as an instantiation of the architecture presented in [22]. The architecture offers a people-centric paradigm for large-scale sensing at the edge of the Internet using an opportunistic sensor networking approach. This approach leverages mobility-enabled interactions and provides coordination between people-centric mobile sensors, static sensors (e.g., [27]) and edge wireless access nodes (i.e., SAPs) in support of sensing, tasking, and data collection. Figure 1 shows a pictorial overview of the BikeNet system. Details of a prototype implementation are included to make the architecture and design descriptions more concrete.

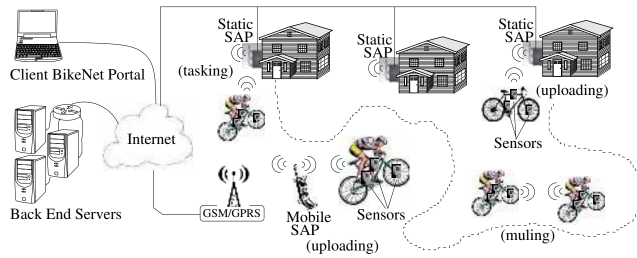


Figure 1: BikeNet System Overview. Sensors collect cyclist and environmental data along the route. Application tasking and sensed data uploading occurs when the sensors come within radio range of a static sensor access point (SAP) or via a mobile SAP along the route. Sensed data muling can occur when cyclists come within mutual radio range. We collect data about the cyclist (heart rate, galvanic skin response), about the cyclist's performance (wheel speed, pedaling cadence, frame tilt, frame lateral tilt, magnetic heading), and about the cyclist's surroundings (sound level, carbon dioxide level, cars).

2.1 Hardware

The BikeNet system hardware is organized into three tiers, the back end server tier, the sensor access point (SAP) tier and the mobile sensor tier. In the following, we discuss the design and implementation of each tier, along with information on ruggedizing and calibration of the hardware.

2.1.1 Mobile Sensor Tier

The mobile sensor tier incorporates a number of bicycle-mounted and human-mounted Moteiv Tmote Invent [8] mobile sensing platforms. Together these sensors gather data concerning cycling performance, cyclist health and fitness, and the environment surrounding the cyclists' routes. The Tmote Invents mounted to a particular bicycle, along with those mounted to the human riding the particular bicycle, constitute a BAN. Intra-BAN communication occurs via short range IEEE 802.15.4 radio. The BAN architecture is designed in a modular way such that sensing components can be added or subtracted simply according to user preferences (dynamically) set in software. Figure 2 shows a logical representation of the *bicycle area network* (BAN), and Figure 3 shows a prototype sensor-enabled bicycle.

We use the native sensors provided by the Tmote Invent: a two axis accelerometer, a thermistor, a photodiode, and a microphone. We also interface a number of additional sensors to the Invent. We process the accelerometer data to measure the angle of incline, and lateral tilt of the bicycle. To measure the angular velocity of the wheel and pedal, forward speed, and distance traveled, we attach a magnet-triggered reed relay mounted across the Invent's user button. Every time the relay closes (every pedal/wheel rotation) a TinyOS [11] interrupt event is generated. To measure direction and devia-

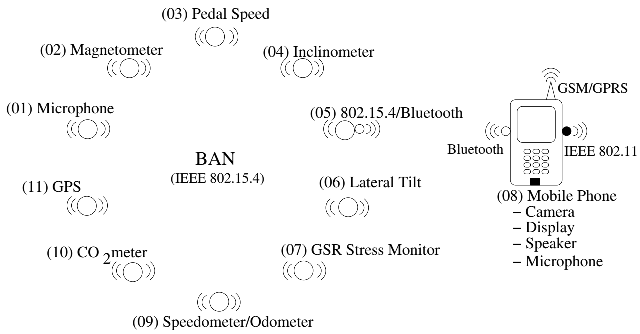


Figure 2: Logical representation of bicycle area networking. Sensors share a common IEEE 802.15.4 channel. A mobile phone plays a dual architectural role depending on whether its cellular radio is active/connected. If connected to the cellular back end the mobile phone acts as a mobile sensor access point (SAP) facilitating real-time sensing; else it acts as a local member of a BAN engaged in delay tolerant sensing.

Figure 3: Physical implementation of the BikeNet system. Numbered sensors installed on the bicycle map to the sensor types labeled in the logical BAN representation in Figure 2.

tion with respect to the Earth's magnetic field, we add a dual axis magneto-inductive sensor (Honeywell HMC1052L) by connecting the sensor output to two ADC channels on the Invent and connecting a free I/O pin from the Invent's MSP430 microcontroller configured as output to act as a digital control line. We further process the magnetometer data for use as a metal detector, and in particular for automobile detection. To provide a common notion of absolute time and location within a BAN, we connect a Garmin Etrex 12 channel GPS unit (Figure 4(b)) via the UART0 port of the Tmote Invent's MSP430. The Garmin Etrex provides time and location data at the fixed rate of once per two seconds via its RS232 interface. To measure the carbon dioxide levels in the environment surrounding the cyclist, we interface the standard Tmote Invent with the Telaire 7001 CO_2/Temperature Monitor, via an ADC port of the Tmote Invent's MSP430. To measure the galvanic skin response of the cyclist, we use an ArcherKit Biofeedback Monitor connected to a Tmote Invent. Wires connected to the fingers of the cyclist measure epidermal microcurrents.

The Tmote Invents and external sensors are powered using rechargeable batteries in our prototype. A commercial implementation could leverage ongoing work in energy harvesting (particularly from pedaling and frame vibration) to reduce the need for external recharging. This is outside the scope of the current work.

2.1.2 *SAP Tier*

The SAP tier offers high performance, high reliability, and secure gateway access from the sensor tier to the back end servers. This access allows sensed data to flow to the system repositories, and provides a point of command for the architecture to task available sensors with user application requests/queries. When possible, these gateways are symbiotically implemented on the back of existing network infrastructure by plugging a short range radio module into the existing network element (e.g., IEEE 802.11 access point), allowing it to communicate with the sensor tier. SAPs can be static and wired directly to the Internet, or can be mobile and use a data service over a wide area radio access network to provide connectivity to the back end (e.g., mobile phone with GSM/GPRS). We study both tasking and uploading via both static and mobile SAPs in our implementation. SAPs are also equipped with sensors to provide context and validation for uploaded data.

The static SAP is implemented using an unmodified Tmote Invent plugged into the USB port of an Aruba AP-70 IEEE 802.11a/b/g access point. The Aruba is running a customized version of OpenWRT, an embedded Linux variant. The BikeNet SAP is implemented as an overlay of tools requiring only user privileges. Certain kernel module support is needed; modules are loaded at run time if necessary. The tools distribution is cleanly encapsulated in a single tarball making symbiotic deployment of a BikeNet SAP on to a standard WiFi access point easy to manage. The mobile SAP is implemented using a Nokia N80 paired to a custom built Bluetooth/802.15.4 gateway via its Bluetooth radio. The N80 SymbianOS uses a serial device emulation of the Bluetooth SPP profile to read and write from the Bluetooth/802.15.4 gateway. The back end interface of the SAP uses GSM/GPRS to the BikeNet repository and back end services. This is done with a combination of SMS messages from the back end pushed to the phone, and TCP connections initiated by the N80 to transmit responses to a back end server that translates data uploads to SQL commands to insert data into the repository.

The use of a personal device like a cell phone as a mobile SAP gives rise to an interesting dual role for the N80 in our system. Architecturally, there is a clean separation between SAP and sensor tiers, but in the case of a mobile phone owned by the cyclist the BAN to which the cyclist belongs may have continuous access to the SAP services and resources whenever GPRS service is available. A mobile phone thus plays a dual architectural role depending on whether its cellular radio is active/connected. If connected to the cellular back end the mobile phone acts as a mobile SAP facilitating real-time sensing; else it acts as a local member

(a) A two-axis magnetometer is attached to a Tmote Invent via its ADC.

(b) An external GPS unit is attached to a Tmote Invent via its UART0 port.

(c) A BikeNet static SAP is a WiFi AP with an Invent inserted in the USB port.

(d) Waterproof OtterBox. Wires are fed through drilled holes that are then filled with silicone sealant. Wires have crimped connectors for easy disconnect.

(e) Ground truth video/sound/photo helmet with four N80s and GPS receiver, only for use in debugging our system and validating our inference techniques.

(f) BikeNet mobile SAP implementation. The Nokia N80 Bluetooth radio associates with a custom-built Bluetooth/802.15.4 gateway.

Figure 4

of a BAN engaged in delay tolerant sensing. GPRS pricing and performance also comes into play when using the cell phone as a SAP, but we set aside this problem for future work.

2.1.3 Server Tier

Members of the back end are Ethernet-connected servers equipped with practically unbounded storage and computational power. These provide a number of services to the architecture, some of which are described in Section 2.2.6. In particular, it is to the back end servers that system users connect to submit application requests/queries for execution in the sensor tier, and to retrieve and visualize sensed data.

2.1.4 Ruggedizing the Hardware

Because of the outdoor nature of the BikeNet testbed we take steps to protect the Tmote Invents from the weather (e.g., rain, snow) by enclosing each in an OtterBox 1600 Case. The OtterBox comes with adhesive foam that is customizable to a degree that allows us to secure the Tmote Invents inside the cases without any slipping. A number of sensors require running wires from the Tmote Invent out of the OtterBox to other places on bicycle or cyclist (e.g., the WheelSensor's reed relay is wired to the front fork of the bicycle). For these we drill holes through the OtterBox 1600 and fill the holes with silicone gel after passing through the wires to maintain waterproofing. We cut the wires inside the box and crimp/solder on connectors (see Figure 4(d)) to allow a quick disconnect of the Tmote Invents for recharging. Additionally, the Otterbox cases are securely fastened to the bicycle frame, using a system of steel mounting bars and steel hose clamps, since bicycling implies often severe vibration and jolting. The OtterBoxes are screwed to these mounting bars and the screw holes are sealed with silicone gel. In determining the geometry and placement of the mounting bars we have attempted to minimize vibration and unwanted degrees of freedom for the sensors (a picture of a sensor-enabled bicycle appears in Figure 3).

2.1.5 Calibration/Validation

Despite efforts to mount the accelerometers at perfect right angles (in two dimensions) with the ground, we find that calibration is required for each bicycle in order to correctly understand the measured values. Even if the error angle of the mounting bracket is small it can lead to a large skew in the calculated slope, because of the non-linear nature of the inverse tangent function used to calculate the slope. Stationary calibration is done in the lab by matching the bicycle-mounted accelerometer outputs against a set of known inclines to derive a calibration curve for each device. To validate this static calibration in the field, we manual measure a 0.75km section of the road containing slopes from 0 to 7 degrees using a laser level (model TUV EPT-97A, 650nm) at 30m intervals. We receive excellent correlation between manual measurements and those made using the accelerometer (the TiltSensor role).

We find that calibration is also necessary for the magneto-inductive sensors due to the steel frame of the bicycle, and the steel mounting bars. This is done by executing a hard/soft iron calibration [28] for each bicycle, and adding the correction for the magnetic declination of Hanover, NH, USA.

We infer cyclist fitness level using a combination of the lateral tilt, slope, and pedal speed to wheel speed ratio. To check our inference technique against a more direct physiological measure of cyclist fitness, we use the Garmin Forerunner 301 Heart Rate Monitor/GPS. A positive correlation between our inferred cyclist fitness level, and that indicated by the actual cyclist heart rate validates our technique.

To provide richer context for the sensor measurements and inference we do in BikeNet, we attach four Nokia N80 phones on a bicycle helmet, i.e., facing front, back, left and right (see Figure 4(e)). Using continuous video capture (both visual and audio) throughout the ride we are able to validate that events sensed/inferred by BAN sensors are at least reasonable/probable and depending on the measurement type we can definitively validate the data (e.g., car passing the bike or not).

To validate detection-based inferences, we use a standard Tmote Invent programmed to write the (time, location) 2-tuple to the Flash every time the user button on the Tmote Invent is clicked. We term this the *ButtonMote* for ease of reference. For example, in testing the MetalDetector (Section 3.1.1) we manually click the ButtonMote user button every time we pass a parked or moving automobile or an automobile passes us, and compare the time/location-aligned MetalDetector trace with the ButtonMote trace to determine detection accuracy.

2.2 Software

Figures 5(a), 5(b) and 5(c) show how the BikeNet software system maps to the three tier hardware architecture, respectively defining the mobile sensor, SAP and back end software sub-architectures. In our implementation, communication between the SAP and back end sub-architectures is via either a TCP/IP stack (for static SAPs) or a GPRS/GSM stack (for mobile SAPs). Primary software elements are discussed in the following.

2.2.1 BikeNet Role Assignment

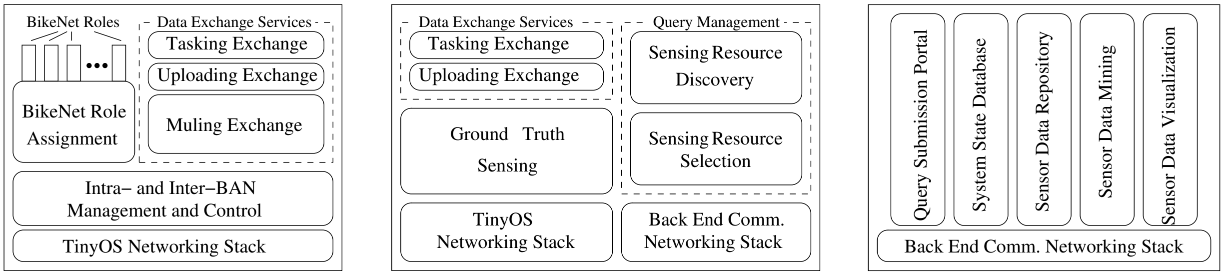
For purposes of modularity the functional requirements within a BAN are divided into logical *roles*. The *PedalSensor* and *WheelSensor* roles measure the angular velocity of the pedal and front wheel, respectively. From these the current and average speed, distance traveled, pedaling cadence and gear ratio is measured or inferred. The *TiltSensor* role measures the angle of incline of the bicycle frame with respect to the gravitational force vector, allowing for real time slope calculation and a mapping of the terrain along a cyclist's route. The *LateralTiltSensor* role measures the lateral angle of incline of the bicycle frame. The *CompassSensor* role measures the instantaneous angle of the bike frame with respect to the Earth's magnetic field. The *MetalDetector* role measures distortions in the the Earth's magnetic field caused by nearby ferromagnetic metals, allowing inference of the amount of passing automobile traffic. The *SyncSprinkler* role provides a common absolute notion of time and location to all members of the bike area network via periodic short range broadcasts. The *LocalDisplay* role provides a means to locally display sensed data. The *CO_2Sensor* role measures

BAN Hardware	BAN Roles
Invent	PersonalNode
Invent using its accelerometer	TiltSensor
Invent using its accelerometer	LateralTiltSensor
Invent using its microphone	SoundSensor
Invent + magnetometer	MetalDetector, CompassSensor
Invent + reed relay mounted on pedal	PedalSensor
Invent + reed relay mounted on wheel	WheelSensor
Invent + GPS	SyncSprinkler
Invent + 802.15.4/BlueTooth Gateway + N80	CameraSensor, LocalDisplay
Invent + CO_2 meter	CO_2Sensor

Figure 6: Mapping between BAN hardware and logical roles.

the carbon dioxide content in the atmosphere surrounding the bicycle, allowing the system to infer whether the cyclist is passing through an urban area (more CO_2 from auto exhaust) or a rural area (less CO_2 due to plant respiration). The *SoundSensor* role measures the volume of noise in the environment surrounding the cyclist, and is used for voice triggered sensing and audio annotation of a cyclist's ride. The *CameraSensor* role provides triggered capture of an image, or a video clip of specified duration. The *PersonalNode* role provides control via short range radio over the other sensing roles, including executing user preferences within the BAN (e.g., required sensors, sampling parametrization), and signaling the start and stop of a cycling trip. Each cyclist necessarily possesses a PersonalNode, but all other roles are optional, depending on the sensing preferences of the cyclist. Figure 6 shows the mapping between BikeNet roles and the sensing hardware, where each row represents a different (set of) devices on a fully equipped prototype bicycle.

We assume that each cyclist possesses a mobile personal computing device (e.g., Tmote Invent, Nokia N80, radio-equipped Apple iPod) at all times that can be tasked by the SAP to take on the PersonalNode role. In our prototype system, each cyclist carries a Tmote Invent preconfigured with the PersonalNode role. The PersonalNode role includes a list of user preferences that dictate what additional sensing roles are desired to quantify the cyclist fitness/performance/environment. These sensing roles are split into two lists, required and preferred, that are included into a *hello* beacon periodically broadcast by the PersonalNode. The *hello* beacon also includes the required sensing parameterization (e.g., sample rate). Each available mobile sensing platform (i.e., Tmote Invent) that receives the beacon replies with a *hello reply* if its sensing capabilities match either a required or preferred role requested in the *hello* beacon. The *hello reply* indicates which role(s) the respondent is offering to fill. However, a recipient of the *hello* that is already associated with another PersonalNode will not reply. Upon receiving a *hello reply*, the PersonalNode first registers the respondent and the role(s) it is offering to fill, and then sends a *hello reply ack* to complete the association. The *hello reply ack* contains a list of identifiers reflecting the current BAN membership. Subsequent *hello* beacons sent by the PersonalNode do not request sensing roles that are already being filled by associated mobile sensing platforms. If a *hello reply ack* is not received in response to a *hello reply*, the *reply*



(a) Mobile sensor architecture. (b) Sensor access point (SAP) architecture. (c) Back end architecture.

Figure 5

is retransmitted up to *N* times. If after *N* times the *ack* is not received, then it is assumed that mobility has carried the PersonalNode and potential sensor out of range and the partial association state is purged.

Though generally a mobile sensing platform may, depending on sensing capabilities, be able to take on more than one BikeNet role, for our current prototype implementation we allow only one role per sensing platform to work within hardware limitations of the Tmote Invent. For example, there are a limited number of free configurable I/O pins and ADC channels available for external sensors on the Tmote Invent, and a shortage of Flash/RAM. In the future with more capable hardware, we will be able to condense our current implementation, assigning multiple roles to a single sensing platform and greatly reducing BAN complexity and costs (i.e., monetary, radio congestion, energy consumption). In the rest of the paper, unless stated we treat the logical sensing role and the mobile sensing platform to which the role is assigned as synonymous.

2.2.2 Intra-BAN and Inter-BAN Management

Localization and Synchronization. The SyncSprinkler role provides, via a periodic broadcast within the BAN, periodic samples of the instantaneous absolute time and location. In our implementation these are obtained from a GPS unit. The SyncSprinkler controls its transmission power to limit the scope of its beacons, thereby maintaining a higher location accuracy for all broadcast recipients. All BAN members' time estimate is updated externally with the values contained in the SyncSprinkler broadcasts, and internally via a local clock set to provide higher time resolution between received SyncSprinkler broadcasts.

Sensing Control. When the PersonalNode has established associations with sensing platforms (i.e., Tmote Invents) sufficient to meet all the roles specified by the user preferences, an LED on the PersonalNode indicates a "Ready" state. A button click on the PersonalNode when in this "Ready" state sends a *start* message broadcast from the PersonalNode indicating that the ride is beginning and sensors should start collecting data with their prescribed parametrization. This message is acted on by mobile sensing platforms that are associated with that PersonalNode, moving both the PersonalNode and the associated mobile sensors into the "Started" state. If associated sensors do not receive a *start* message within a timeout period, the association times out and the mobile sensors are free to associate with another PersonalNode at that time. A subsequent PersonalNode button click while the PersonalNode is in the "Started" state sends a *stop* message broadcast, signaling the end of the ride. The *stop* message causes mobile sensors associated with that PersonalNode and are in the "Started" state to cease sensing.

Event-triggered Sensing vs. Continuous Sensing. Sensing is set up to occur either continuously or only when triggered by other events. In the continuous case, the user preferences executed by the PersonalNode parametrize the sensing capture (e.g., sampling rate, duration, local processing functions) that takes effect immediately upon receiving the *start* message of the sensing control protocol. Continuous sensing in BikeNet is appropriate for roles such as the TiltSensor where terrain mapping should be continuous. On the other hand, some sensing operations may be too energy expensive for a mobile sensing platform to do continuously, or may not have meaning except under certain contexts (e.g., certain locations of interest, or under certain sensed circumstances). Triggers are defined by dynamically updatable user profiles executed by the PersonalNode that specify the conditions under which sensing should occur.

The BikeNet implementation support of triggered sensing includes methods to define and submit sensing triggers and actions to the PersonalNode for execution within the BAN. Upon receiving the triggered sensing definition, the PersonalNode breaks apart the conditions that must be met for the action to take place, and reliably transmits each condition (e.g., "slope $>$ 5 degrees") to the BAN member suited to evaluate the condition (e.g., the TiltSensor). When a condition evaluates to true, the BAN member signals the PersonalNode. The PersonalNode initiates the action when all conditions for a given triggered action are met. We are currently focusing on triggered photography, video and audio capture using the camera and microphone on the N80, when certain conditions in the BAN are met, but we also implement support for a number of other actions such as sending data to be displayed on the LocalDisplay, sensing something at a different parametrization than the current one, playing a sound on the N80 or Tmote Invent speaker, transferring sensed data from one Tmote Invent to another, and blinking LEDs.

Real-time Feedback/Display. The local display protocol is used by the LocalDisplay to query other BAN members for values to display. The LocalDisplay is provided by a handlebar-mounted N80 mobile phone, via the Bluetooth/802.15.4 gateway shown in Figure 4(f). The Tmote

Invent's hardware design shares the SPI bus between radio and flash, and the same physical microcontroller pins are used for UART0. Since the Bluetooth-to-Serial converter is connected to the Tmote Invent UART0 port, this precludes simultaneous radio communication and display updating. Hence, BikeNet uses a simple TDMA-like time slot assignment on top of the TinyOS CSMA MAC to improve communication between the roles generating sensor data and the LocalDisplay. The LocalDisplay periodically broadcasts a query for data and the sensor roles register the first LocalDisplay they hear a query from as the only LocalDisplay they will reply to thereafter. This association times out after a period if no queries are heard from the LocalDisplay. The data that a given sensor role returns to the LocalDisplay is a matter of user policy, but a typical display includes speed, distance traveled, bike frame tilt angle, pedal RPMs, and time of day. To support flexibility in the user configuration of the display, data is represented in the packet with (type,length,value) format for flexibility.

2.2.3 Data Exchange Services

Three types of data exchange occur in the BikeNet system: tasking exchange, uploading exchange, and muling exchange. The tasking and uploading data exchanges take place between mobile sensing platforms and SAPs. The muling data exchange takes place only between members of the mobile sensing tier (e.g., Tmote Invents). As default, BikeNet uses a delay-tolerant mode where a BAN's PersonalNode mules data for sensing roles in its BAN (up to the limits of its available storage) and uploads the data via wire or a wireless upload protocol. Inter-BAN muling and in situ uploading via either mobile SAPs or opportunistically encountered static SAPs support queries from back end user applications that may want data in a more timely manner.

In the BikeNet tasking exchange, a SAP interacts with available mobile sensing platforms (e.g., Tmote Invents) to first instantiate a PersonalNode programmed with a cyclist's BAN preference profile. Based on this profile, the PersonalNode assembles a BAN by tasking other available mobile sensing platforms with the required sensing roles as discussed in Section 2.2.1. Aside from this BAN bootstrapping, the tasking exchange also includes the handling of user queries/requests for data by back end system users, received via the SAP. The PersonalNode responds to these queries by invoking the necessary continuous or triggered sensing (Section 2.2.2) within its BAN.

In the muling exchange, sensed data is transferred between mobile sensors outside of the radio range of either a mobile or static SAP. A simple muling protocol is implemented on every Tmote Invent, but the option to activate muling (i.e., spend Flash space to carry others' data) is set by cyclist preference. The protocol uses an *advertisement-accept-data* exchange, where the *advertisement* specifies the amount of data the provider wants to have muled, the *accept* message indicates the amount of data the consumer is willing to mule (based on Flash constraints) and the *data* message represents a burst of data packets from the producer to the consumer. In addition, Stop-and-Wait ARQ with a maximum of three resends provides for reliable transfer of the data packet burst. If a producer still receives no acknowledgment after three resends of the same packet it will assume the session is over and begins advertising anew. Our implementation includes support for replication of sensed data (i.e., via the muling exchange) but the replication of muled data is not allowed. Restricting the right to replicate to the data origin allows it to maintain control over the number of copies of its data that are circulating and also to vet (in terms of trustworthiness) all candidate mules.

In the uploading exchange, when a BAN comes within the radio range of a mobile or static SAP, the Tmote Invents composing the BAN attempt to upload sensed data to the back end data repository. The upload protocol message exchange is identical to that of the muling protocol. When a SAP receives data packets, they are forwarded (in both the mobile SAP and static SAP cases) to the back end repository. The decision to accept new upload sessions is made based on channel congestion around the SAP.

2.2.4 Ground Truth Sensing

In the BikeNet sensing system, SAPs are equipped with certain sensors and can provide *ground truth* measurements. Ground truth[1] sensing refers to a trusted, high fidelity, always accessible stream of data. One use of ground truth data is as a filter applied to data uploaded from a sensor before the data is passed by the SAP to the back end repository. The ground truth filter can be applied to validate or invalidate uploaded data when the uploaded data samples and ground truth data samples have a high expected correlation (e.g., temperature sampled at the same location and at the same time, samples triggered by the same set of circumstances). Further, ground truth sensing can be used to satisfy queries coming from a back end system user that have only coarse location context requirements. Ground truth data is also used to satisfy queries coming from a BAN in the radio range of a SAP. In this case the BAN can ask for readings from the SAP's ground truth sensors, e.g., as part of a self-calibration routine.

2.2.5 Query Management

The query management component on the SAP handles queries both from the back end system user, and from the PersonalNode of a BAN. It invokes a sensing resource discovery routine to determine what sensing resources are available to meet the sensing request. The routine checks both any ground truth sensors on the SAP itself (Section 2.2.4) and available sensing resources on any BANs that may be within radio range of the SAP. Once a list of available sensing resources is compiled, the SAP invokes a sensing resource selection routine to decide which resources will be tasked in order to satisfy the request, and invokes a tasking routine to execute the necessary request (i.e., a simple function call if the resource is on-SAP, or via the tasking exchange (Section 2.2.3) if the resource is in a BAN in radio range. In the BikeNet implementation, we have experimented with handling queries to the SAP, originating both from the back end and from a BAN in radio range, for ground truth data. In particular, using a cellular phone as a mobile SAP, we have

[1] While related in principle, this notion of ground truth sensing should not be confused with sensing for experimental validation and debug (e.g., using the quad capture video helmet in Figure 4(e)).

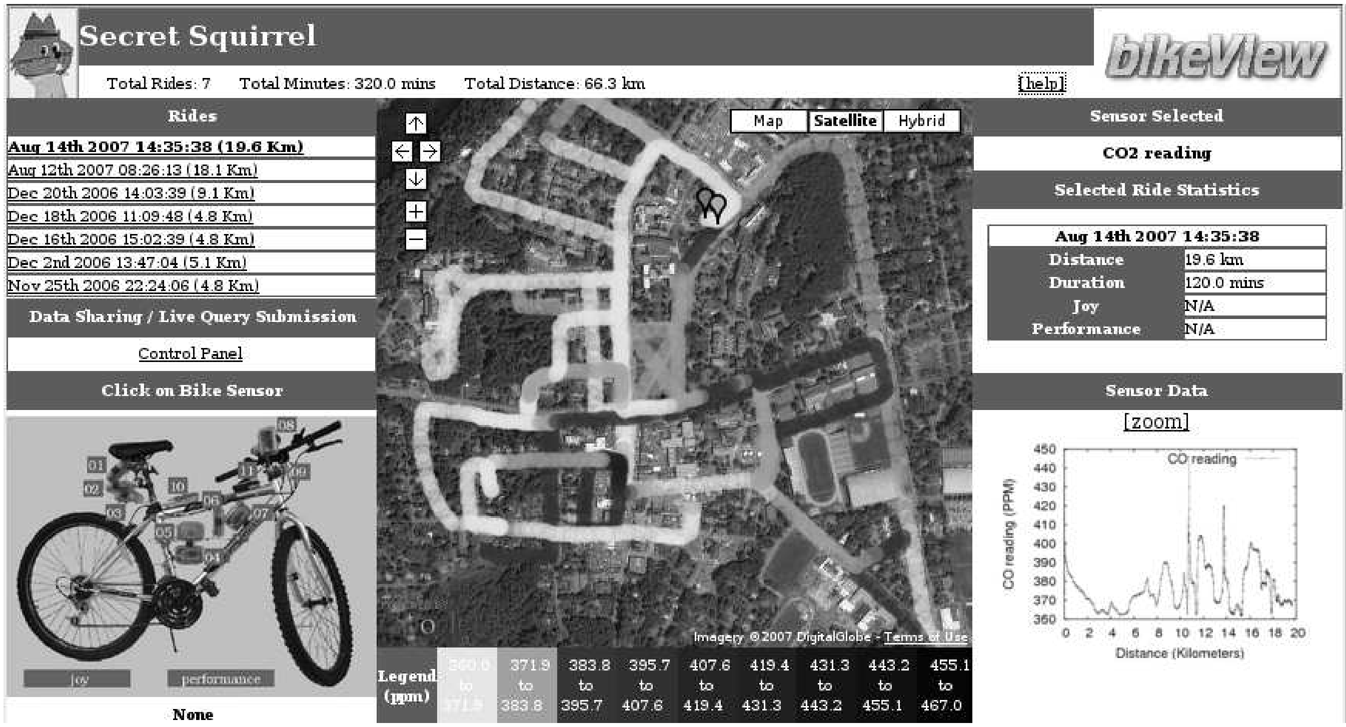


Figure 7: BikeView portal [4] for data display and query submission. A CO_2 map of Hanover, NH, USA streets on a summer weekday afternoon is shown.

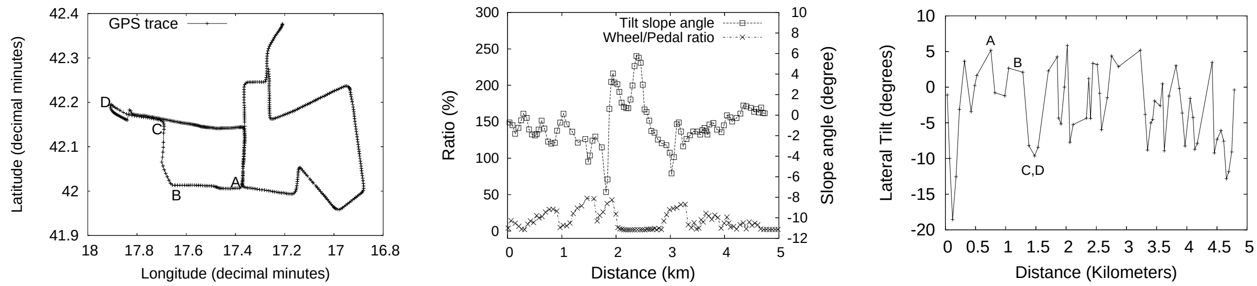
experimented both with event triggered capture of images, sound and videos requested by the BAN (e.g., an audio annotated ride); and with direct requests from the back end BikeNet web portal for image, sound and video samples.

2.2.6 Back End Services

Query Submission Portal. The BikeNet back end includes a web portal (*BikeView* [4]) containing a graphical presentation of a cyclists data, but also allowing for the real time querying of BANs using a GPRS connection via the N80, if the cyclist is using such a device for their PersonalNode. The user can select the BAN of interest and assemble a query to submit to the query manager component of that SAP using a collection of pull down menus. A final mouse click transmits the query over the cellular network to the selected mobile SAP. We implement the ability to query a BAN's location, capture a camera image, and sample the microphone via this portal interface by sending SMS over GPRS to the N80.

Sensor Data Storage, Processing and Visualization. The sensor data repository provides a location for the long term storage of cyclist experience data on a per-cyclist basis and also provides a convenient location for the aggregation of all long term trace data for all participating cyclists. Access to particular data is a matter of a the policy that each cyclist registers with the repository (or a separate access control entity). The sensor data mining component provides a set of standard statistical functions and reusable calculations/ data transformations that a user (e.g., cyclist) can invoke to control the retrieval and presentation of data. For BikeNet we use a number of data interpretation and inference tools and techniques, including scatter plots to look for data correlation, fast Fourier transforms (FFTs) to look for periodicity, running averages to smooth data to look for trends, and interpolation to align samples according to distance. For example, we use a method to filter spurious vibration data from the TiltSensor when calculating the slope of the cyclists path. Currently, the data handling is done in a non-automated way by storing raw data streams in flat text log files, processing these files using Awk scripts to extract data of different types and apply methods for smoothing, averaging, scaling, and using an external tool for FFT analysis. Further, we develop scripts to transform data values into *BikeView* [4] visualizations. With BikeView (see Figure 7), we present summarized collected data sorted by user, and sorted by ride within each user account (akin to the presentation of "My Runs" on the Nike+iPod web page [6]). Detailed sensed data can be obtained by simple mouse hovers and clicks over the graphic representations of different rides. The vision is to provide back end sharing between users facilitated by dynamic creation of group pages that are visible to all users in the group and to which all group members can publish data.

System State Database. The system state database contains both static state information (e.g., Tmote Invent physical address and sensing capabilities, PersonalNode human custodian information) and dynamic state information (e.g., last known position of a mobile SAPs, and BANs, SAP load average) about elements in the network. In particular, the system state database tracks which BANs are currently in radio range of mobile and static SAPs. This facilitates proper query routing from the back end query submission portal to



(a) The mapped GPS trace of the cyclist route, comprising roads in the vicinity of Dartmouth College in Hanover, NH, USA.

(b) Cyclist fitness is inferred by correlating the gear ratio inferred from the wheel/pedal ratio and the measured road slope.

(c) Lateral tilt plotted versus distance. Changes in lateral tilt are correlated with turns along the route shown in Figure 8(a).

Figure 8

particular BANs, for BAN-specific user queries. The information is also valuable more generally for debugging and management of the network.

3 System Evaluation

We build five fully equipped BikeNet bicycles, implement all of the aforementioned sensing roles using Tmote Invent motes and Nokia N80 mobile phones, build a number of static and mobile SAPs, and implement a functional back end web portal offering query submission and data retrieval services. In this section, we present selected results from several groups of experiments respectively targeted at: quantifying the cyclist experience from sensed data collected about a single cyclist and his environment; looking at performance aspects of key BikeNet subsystems; and measuring the real-time performance of a deployed system across the Dartmouth campus and in adjacent areas of the town of Hanover, NH, USA. We use a common path that we call the ground truth route. This route includes a variety of urban cycling terrain, including built up busy roads in the town center with lots of cars and pedestrian traffic and quiet back roads with little or no traffic. The route exposes cyclists to a variety of flat terrain, gradual down hill and steep uphill sections. Typically the ground truth route takes 25-30 minutes to ride and is nearly 5km long. The experiments are conducted at rush hour and in the middle of the day when there is less traffic and activity. We conducted many experiments over the period August 2006 - August 2007 collecting a typical data set of 0.8 MB per ride per bike. We record the runs using video from the video helmet (Figure 4(e)) that collects quad-directional video of a ride for ground truth validation of our correlation/inference methods (not part of standard BikeNet equipment). BikeView [4] contains an example of one such video recording.

3.1 Cyclist Experience Mapping

3.1.1 Inference and Cyclist Fitness Sensing

In this section we present a series of plots characterizing cyclist behavior and the environmental conditions encountered during a ride. We collect data from each of the sensing roles mentioned in Section 2.2.1, and apply fusion techniques and trend analysis to extract additional information from the raw data.

Figure 8(b) shows the measured slope profile of the route calculated from TiltSensor readings versus distance (the Wheel/Pedal ratio curve is explained later). The slope s is calculated according to $s = arctan(x/y)$, where x and y are the TiltSensor's measured x- and y-channel accelerometer readings, respectively. We register accurate measurements when the bike is stationary; error increases with speed and terrain roughness due to unfiltered vibrations and cyclist behavior. The slope profile which matches the manually measured ground truth road segment (c.f. Section 2.1.5) well (less than 10% deviation from the ground truth slope measurements).

Figure 8(c) shows the lateral tilt plotted versus distance. The lateral tilt is calculated in the same manner just described for the TiltSensor. A cyclist's aggressiveness in turning is inferred. From the plot we correlate the increases in lateral tilt magnitude shown on the y-axis, with corner turns expected from the mapped GPS trace shown in Figure 8(a). Positive lateral angle indicates a right-side lean whereas negative angles indicate left-side leans. In Figure 8(c), we label (viz. A, B, C, D) a sample of the lateral tilts that can be correlated with corner turns in the cyclist GPS trace (see Figure 8(a)), where at A, B, C, and D the biker makes, respectively, a right, a right, a left, and a left turn. The sharp left tilt (almost -20 degrees) is due to mounting the bicycle at the start of the ride.

The quantitative aspects of the cyclist fitness include the slope of the road/trail that the cyclist covers on his ride, the speed profile of the cyclist, the gear used when traveling up a given slope, and the location of the route. Figure 8(b) shows the slope profile of the road traversed on the cyclist's trip, and the ratio of the tire/wheel speed to the pedal speed. This ratio infers the approximate gear the bicycle is in at a given point in the route, and provides a notion of the fitness of the cyclist. This indicator is most accurate when the cyclist is going uphill, since when coasting downhill the pedals may not be moved much. A strong cyclist can use a higher Wheel/Pedal ratio when climbing hills. In Figure 8(b), intervals where the cyclist changes gears to climb hills are evident (from roughly 1 to 1.25km and from roughly 2 to 2.7km), where the Wheel/Pedal ratio is nearly 1.

Knowledge of the sensed path slope combined with the

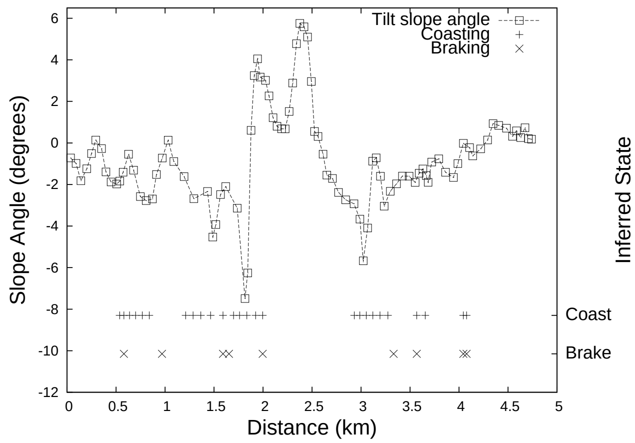


Figure 9: Periods of coasting and strong braking can be inferred from relationships between pedal RPM, wheel RPM and road slope.

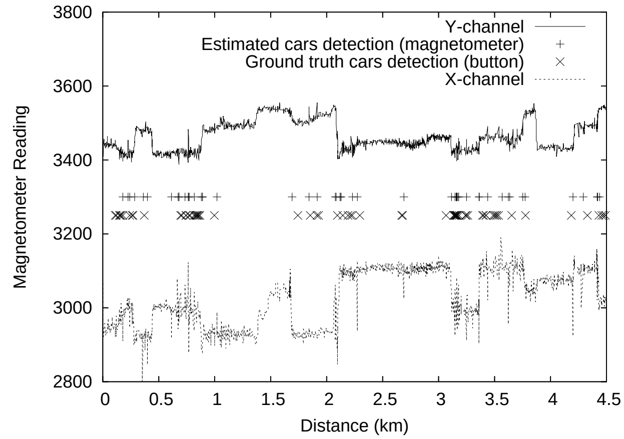


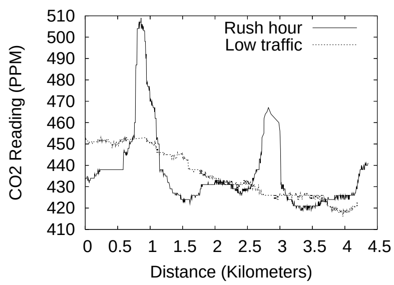
Figure 10: From measured magnetic field distortions, we use a thresholding method to infer locations with dense patches of cars.

measured pedal speed and wheel speed allows us to infer when a cyclist is coasting or braking. On a given bicycle there is a finite discrete set of pedal speed to wheel speed ratios possible when the bicycle chain is engaged with a gear and providing thrust to the bicycle. The cardinality of this set is equal to the number of gears the bicycle has. If the measured ratio of pedal speed to wheel speed does not match one of the allowable values we can infer that the cyclist is coasting. Braking can be inferred in a similar fashion to coasting. It is likely a cyclist is braking if the measured wheel speed slows while the slope is negative (downhill). Further, braking is likely when going uphill if the measured wheel speed slows faster than dictated by the slope of the hill. However, this is more challenging to detect since inference of uphill braking is also dependent on unknown quantities such as the combined mass of the bicycle and the cyclist, and the route surface composition/coefficient of rolling friction.

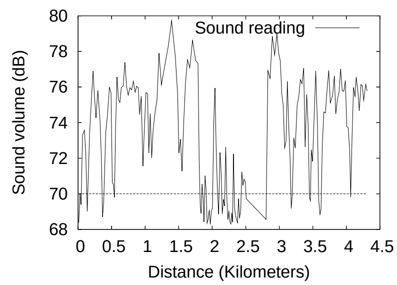
Figure 9 shows a plot of the road slope versus distance along the ground truth route. Applying the simple inference technique of observing decreasing speed when the slope is negative (downhill), we infer sharp braking intervals at 1.6km, 3.3km and 4.1km, which are verified by our known cyclist behavior. In these cases, we see a sharp decrease in wheel speed concurrent with a sustained downhill slope. Similarly, if the pedal speed is near zero and the wheel speed is high, we can easily infer the cyclist is coasting. In the figure, we infer periods of coasting from roughly 1.25 to 2km and 2.7 to 3.3km.

Aside from route topography and personal performance metrics, cyclists are interested in the ambience and safety of a route as a determinant in the overall enjoyment of the cycling experience. We take steps towards quantifying the ambience in terms of automobile traffic, air quality, and sound level. The presence of vehicles is often undesirable for cyclists who have concerns about safety, noise, or pollution. To infer automobile traffic along the cyclist route (Figure 8(a)), each BAN is equipped with a MetalDetector. When the MetalDetector passes close to any large metal body the Earth's magnetic field is deformed and the presence of a car is inferred. To collect ground truth data for the experiment to compare against the inference from the MetalDetector, the car rendezvous event is manually logged by the cyclist with a ButtonMote click. We include the cases when the bike passes a car (parked or moving), and when a car passes the bike. We find that detection of cars more than 2m away is unreliable with our hardware, so we do not try to log when a car passes in the oncoming lane (about 3m away). Each click generates a record of GPS time and location information. Figure 10 shows, the raw x-channel and y-channel readings of a MetalDetector's magnetometer plotted versus the distance covered along the ride. ButtonMote events and positions of inferred cars are overlaid on the same plot. The inference algorithm is run against both x and y channel data and works as follows. First, the exponentially weighted moving average of the magnetometer reading is calculated. If the difference between the current value and the moving average is greater than a threshold, and the current value is a peak (greater than both the preceding and succeeding values), a car is inferred. The threshold values for x and y channels and the moving average weight are learned by training with the readings from 0 to 1 kilometers. These trained values are then used along the rest of the route from 1 to 4.5 kilometers. While discrepancies between the ground truth data and the output of the detection algorithm exist, we note that our aim is not counting the exact number of cars but to identify areas of high automobile concentration. Using this simple thresholding technique we are able achieve a level of detection accuracy that supports this aim.

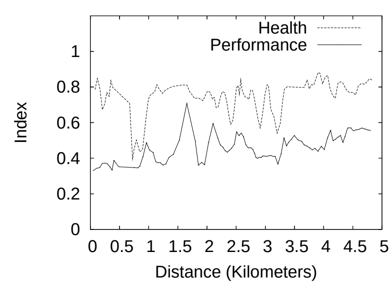
To provide a measure of air quality along the cyclist's route we conduct experiments using a sensor measuring the level of carbon dioxide in the air surrounding the cyclist. Figure 11(a) shows a trace of the carbon dioxide sensor readings along the route shown in Figure 8(a) for two different cases, namely, rush hour and low traffic. The peaks in the rush hour case occur when the biker was cycling on downtown roads with a considerable presence of cars. In fact, variation in carbon dioxide levels measured on roadways is likely to be the result of automobile exhaust. While carbon dioxide has low



(a) CO_2 level along the ground truth route. Large spikes as the cyclist passes through the center of town at rush hour.



(b) Sound level along the ground truth route. Even small town traffic exceeds the long term health threshold (70dB).



(c) Digesting the data: health and performance metrics to visualize a cyclist's experience.

Figure 11

toxicity at all levels we recorded during our experiments, it can act as a predictor of other noxious automobile exhaust constituents such as hydrocarbons, nitrous oxides, and particulate matter. Thus, from readings of the carbon dioxide sensor we can infer how enjoyable the traveled route is for a cyclist from the standpoint of pollution. The portal snapshot in Figure 7 shows a CO_2 map of the Hanover streets on a summer weekday afternoon.

Another way to detect the presence of high vehicle density, and to characterize the ambience of a route, is by measuring the sound volume. Sound in decibels is plotted versus distance in Figure 11(b) for a ride along the ground truth route. The sound volume peaks near 80dB when the route passes through the main intersections of town where the automobile traffic is more prevalent.

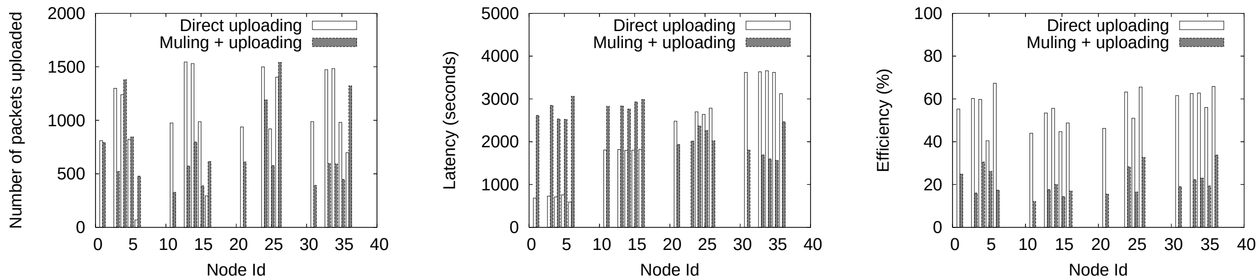
3.1.2 *Interpreting Cyclist Experience*

In Figures 8(b) - 11(b), we present a large amount of raw data and first level inferences. In this section, we introduce two example metrics to help cyclists and other system users understand and make use of the types of data that a BikeNet system provides. The metrics are weighted combinations of various sensor data types. In the metrics introduced below (health and performance), we constrain ourselves to incorporating sensors for which we collect data in our prototype BikeNet implementation, though there are other appropriate sensors that might reasonably be added. The weights (e.g., **a** in the expression for Health below) comprise two subweights: the user-defined preference/importance and the normalizing factor for each element. The user-defined preferences reflect relative personal sensitivities (e.g., a cyclist with asthma might weight the CO_2 higher) to the elements composing a given index score. With the second subweight, we normalize each element (e.g., CO_2) according to its maximum dynamic range measured along routes about which sensed data has been collected so far. At the back end, or on a user's local display, index scores can be plotted versus distance to see the variations across the route to identify critical/interesting sections. For example, the plot of health in Figure 11(c) shows a large dip in the health index just before 1km where the CO_2 spikes (ref. Figure 11(a)). Secondly, users can compare the average index value among different routes at different times to identify the most favorable routes for a given aim (e.g., joy-riding, exercise). These index values can be mapped to colors and routes can be visualized as a color-coded playlist. As the number and coverage of route segments are built up, a lookup service that returns the most healthy route at the desired time between two endpoints becomes possible. By sharing index values for routes of interest, and the user-defined preference weights, cyclists are able to learn from each other about where the good cycling is.

Health. Air pollution and its effect on public health is of great interest in many urban communities. In Austria, France and Switzerland, by measuring particulates specifically from motorized traffic the effect of air pollution on public health is estimated to account for >20,000 adult deaths, more than 290,000 episodes of bronchitis in children, and more than 500,000 asthma attacks each year [20]. Noise pollution is also a factor in urban areas. According to the Environmental Protection Agency's Office of Noise and Abatement Control in order to protect from hearing loss, one should not be exposed to more than 70dB for an extended period of time. Meanwhile the average city traffic is 85dB and in larger cities like New York, the noise level often exceeds 90db. 87% of America's city dwellers are exposed to noise so loud it has the potential to degrade hearing capacity over time [21]. Even in the small town of Hanover, NH, USA (see Figure 11(b)) the noise level is often above 70dB on the main streets at certain times. BikeNet sensing supports not only communal pollution mapping, but on a more personal level it supports the categorization of cycling routes according to their potential impact on a cyclist's health. We define a *health index* that combines data that indicate safety, noise and air pollution (either directly or through inference) as follows:

$$Health = 1.0 - a_1 * CarDensity - a_2 * CO_2Level - a_3 * SoundLevel.$$

From the raw values obtained from the MetalDetector we infer the density of cars along the route, along with raw values of CO_2Sensor and sound levels from the SoundSensor, we derive values for the health index of routes that a cyclist travels. A higher CO_2 level and derived car density imply there are more cars near the cyclist, creating an unpleasant



(a) Data uploaded directly and after muling for the twenty nodes in the four BANs.

(b) Delivery delay comparison between direct uploading and muling + uploading.

(c) Delivery efficiency comparison between direct uploading and muling + uploading.

Figure 12

experience due to exhaust, noise, and increased danger, driving the health index down. Similarly an increase in noise level indicates more traffic, people, wind, shouting, etc., reducing the health index. Figure 11(c) shows the details versus distance of how the characteristics of the route affect the cyclist, highlighting areas that should be avoided on future rides. We use an equal user-defined preference weight of $\frac{1}{3}$ for each of the three elements that are included in the score. The dynamic ranges (i.e., measured difference between max and min) for car density, CO_2 level and sound level are 12, 100 and 70, respectively. Therefore, we use $a_1 = \frac{1}{3} * \frac{1}{12}$, $a_2 = \frac{1}{3} * \frac{1}{100}$, and $a_3 = \frac{1}{3} * \frac{1}{70}$. The average value of the health index over the entire route is 0.746 and the standard deviation is 0.096. As previously mentioned, this average value can be used to help a user rank his routes according to his own preferences and also to share with his peers.

Performance/Fitness. Some cyclists' primary purpose in riding is for exercise or for competition. For these riders we calculate a *performance index*, using the values obtained from the WheelSensor, PedalSensor, and TiltSensor. We compute a unitless measure of performance using the following equation:

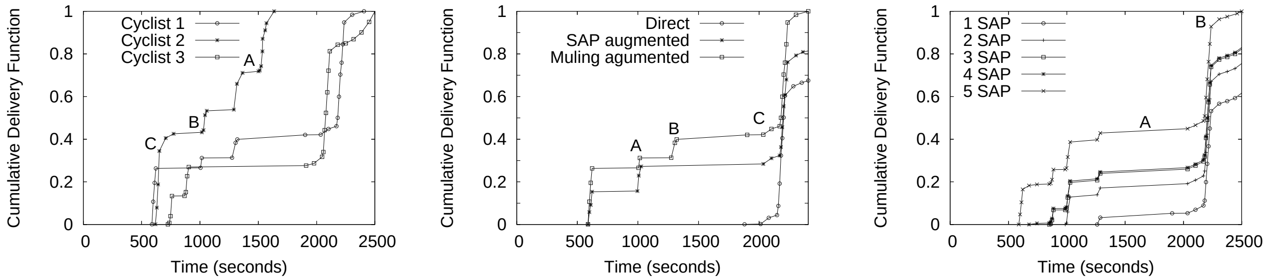
$$Perf. = b_1 * HillAngle + b_2 * WheelSpeed/PedalSpeed + b_3 * Distance.$$

When HillAngle is positive the performance index goes up; when it is negative the index goes down. When the wheel/pedal ratio is high this indicates the bike is in a higher gear (the wheel goes further with fewer pedal turns) and the index increases. The further a rider travels (larger *Distance*) the higher the performance index. The plot in Figure 11(c) shows the performance of a cyclists traveling the ground truth route. The weights are set according to the same rationale as for the health expression, with values: $b_1 = \frac{1}{3} * \frac{1}{13}$, $b_2 = \frac{1}{3} * \frac{1}{80}$, and $b_3 = \frac{1}{3} * \frac{1}{5}$. With these user weights, the average value of the performance index over the entire route is 0.253 with a standard deviation of 0.094.

3.2 Data Muling and Uploading

Throughout the conducted experiments, sensed data is always stored in the local Flash memory of the Tmote Invent. Ultimately, this data must be transferred to the back end data repository. Depending on bicycle and cyclist mobility this transfer may happen directly from BAN to SAP using the upload protocol (c.f. Section 2.2.3), or may happen indirectly via the muling protocol (c.f. Section 2.2.3). Each BAN member knows the identifiers of the other mobile sensing platforms in its BAN as a result of the role assignment protocol (see Section 2.2.1), and will only mule data for mobile sensors not in its BAN. The exception to this rule is that the PersonalNode does mule data on behalf of its associated BAN members. There is no intra-BAN coordinator for either upload or muling exchanges; BAN members transfer their data independently and contend using CSMA for the wireless channel. There are two main scenarios to consider: (i) the bicycle enters in range of a SAP in which case the mobile sensors in the BAN upload their data to the SAP directly or (ii) the bicycle is traveling out of range of the SAP, and must rely on probabilistic mobility of other people or BANs to mule the data to a SAP.

Since the Stop-and-Wait ARQ reliable transfer mechanism used in the muling and upload protocols is well known we omit any evaluation of this mechanism per se. Rather, we aim to characterize the opportunistic sensor networking environment provided by our initial prototype implementation of BikeNet. In Figures 12(a), 12(b) and 12(c), we show results from a multi-bicycle experiment where each cyclist follows a prescribed path; the paths intersect giving rise to inter-bicycle muling opportunities. The paths followed by each of the four bicycles used in the experiment follow the perimeter of a central grassy area called "the Green" commonly used for recreation at Dartmouth College. The area of the Green is approximately 150m by 100m. Two cyclists ride clockwise around the Green and the other two cyclists ride counter clockwise. The transmission range of the Tmote Invent reaches fewer than 50m so the connections among the cyclists are intermittent. However, there are data exchange opportunities when the cyclists who are moving in opposite directions pass by each other. Rendezvous durations are 10.75 seconds on average. After ten minutes of circling around the Green, each cyclist leaves the square in turn at fifteen minute intervals and parks the bicycle within the radio range of the SAP installed at the Sensor Systems Lab in the Computer Science building which is 250m away from the northeast corner of the Green (i.e., out of range).



(a) Data delivery traces for three cyclists using direct upload from BAN to SAP.

(b) Comparison of different delivery methods for single cyclist's data.

(c) Evaluation of delivery delay versus number of SAP encountered en route

Figure 13

In Figure 12(a), we show the number of data packets directly uploaded and the number of data packets muled to the SAP. Data records have unique identifiers to allow filtering of duplicates between muled and directly uploaded packets. The x-axis of the plot shows the identifiers of the Tmote Invents in each BAN. The Tmote Invents with identifiers 1 through 6 belong to BAN-1, 11 through 16 belong to BAN-2, 21 through 26 belong to BAN-3, and 31 through 36 belong to BAN-4. This x-axis is the same for Figures 12(b) and 12(c). Figure 12(a) shows that the rendezvous intervals in the experiment support a substantial amount of muling exchange. Overall, more data packets are directly uploaded than muled, but a considerable amount of data are muled before uploading (e.g., almost half of BAN-1's data).

Figure 12(b) presents the latency of direct uploading and muling. We measure the latency as the time difference from the time the sensor data packet is generated to the time the packet is uploaded to a SAP, either by the originator of the packet or by a mule. The result clearly shows the benefit of muling. The data from BAN-3 and BAN-4 are muled by the Tmote Invents on BAN-1 and BAN-2 which enter within radio range of the SAP earlier than BAN-3 and BAN-4. As a result, the data from BAN-3 and BAN-4 are delivered earlier with muling than with direct uploading. In particular, data from BAN-4 experience an average of 1500 seconds less delay with muling than with direct uploading.

Muling implies a performance penalty due to the additional radio transmissions that are required on the path from origin to mule(s) and mule(s) to SAP. Figure 12(c) illustrates the transfer efficiency for both muling and direct uploading as a function of the packets transferred. Here we define efficiency as the ratio of data bytes transferred to the total bytes sent (including data packet replicas and retransmissions). A higher efficiency for a given transferred packet reflects a number of possible factors, including a higher quality radio link and less congestion between sender and receiver, both of which lead to fewer retransmissions. The data origin always stores a copy in its local Flash hoping to upload the data directly to a SAP. In our experiments, the muling replication degree is one, meaning we allow the originating Tmote Invent to transfer only a single copy of the data to another mobile sensor via the muling protocol. Further, we do not allow multi-hop muling; only the originating node may replicate data. Figure 12(c) shows that the muling efficiency is less than uploading efficiency as expected. While the uploading efficiency ranges from 40% to 68%, the muling efficiency ranges from 12% to 33%. The difference between the muling efficiency and the uploading efficiency represents the cost for improving the data delivery delay. We are currently studying the effect of replication degree on performance; we expect to see that more replication leads to improved delivery delay performance, but also lower efficiency.

3.3 Town-scale System Scenario

We build five sensor bikes and implement a small-scale BikeNet testbed with seven static SAPs at a number of points across the Dartmouth College campus and in the town of Hanover to validate and evaluate an operational BikeNet system. In what follows, we present results from data collected by a group of three cyclists on the morning of November 20, 2006. We have collected a significant amount of data from over 50 different BikeNet experiments starting in summer 2006 but here only present data from a single-shot experiment with the three cyclists. The three cyclists' routes and the times they started their rides are pre-planned. Cyclists 1 and 2 live near each other and ride much of the way toward campus from the town together.

Before getting to the campus they rendezvous with cyclist 3 before cyclists 1 and 3 depart toward the library while cyclist 2 heads to the Computer Science building. The longest journey time for a cyclist is 40 minutes. The results presented in this section provide insights into how live sensor data collected by each of the bikes is either muled or directly uploaded to a passing SAP, in an opportunistic manner.

We first consider the time taken for each cyclist to upload its data into the back end data repository via direct upload to a SAP and present the results in Figure 13(a). Note that in this case the data uploaded to a SAP by a cyclist's BAN includes data originating in the local BAN and any data muled on behalf of other BANs. Figure 13(a) shows the cumulative delivery function versus the trip time. We define *cumulative delivery function* as the cumulative fraction as the trip proceeds of the total data packets delivered to the back end. For example, at 520 seconds into the experiment cyclist 2 has delivered 40% of its data including any data it may mule on behalf of cyclists 1 and 3. The initial lack of any data delivery evident in the plots is due to the initial absence of

SAPs along the route from the homes of cyclists 1, 2 and 3. Cyclists 1 and 2 encounter a SAP along Main Street in Hanover at approximately 520 seconds into their ride. Even though only a modest number of SAPs are deployed we can see from the plot that all bikes are capable of delivering sizable fractions of their data before their trips end. For example, cyclists 1 and 2 deliver approximately 50% and 40%, respectively, of their data to the back end repository before the half way point of the trip time. From Figure 13(a), we observe that data is transferred between bikes and SAPs in quick bursts. The amount of data transferred in strongly influenced by the short contact times which are a product of the short range radios used in the BikeNet experiments.

Figure 13(b) shows the delivery of sensor data that is generated by only a single cyclist - in this case cyclist 1. We consider three scenarios of possible transfer between cyclist 1 and the data repository: *Direct*, which is where cyclist 1 keeps all its data, does not replicate and only uploads to the SAP at the Computer Science building (the cyclist's destination); *SAP augmented*, which is where cyclist 1 opportunistically transfers data to SAPs it encounters along the way with no help from mules; and finally, *Muling/SAP augmented*, which exploits muling and opportunistic use of SAPs to transfer cyclist 1's data to the repository. Figure 13(b) shows the performance of these three types of communication across the trip time. From the plot we can observe the direct benefit of muling: cyclist 1 is disconnected for approximately 15 minutes between points A and C in the plot of the cumulative delivery function, but at the intermediate point B the delivery of cyclist 1's packets continues.

Next, we evaluate the impact of the incremental addition of SAPs to the system on the average delay of data delivered from the three cyclists to the back end repository. As the number of SAPs increases from one to five, we plot the cumulative delivery function versus time. We run five trials at each SAP level, each with the three cyclists riding prescribed routes from their respective homes to the Computer Science department. Figure 13(c) shows the cumulative delivery functions for each of the five cases, truncated at the time when 100% of data is delivered by all cyclists for the case of five SAPs. The plot shows that at the time all data is uploaded in the five SAP case, only 78% and 70% of data is uploaded in the three SAPs and one SAP cases, respectively. When the cyclists return to the Computer Science building they become stationary and upload their remaining data. This is reflected in the steep step in all the curves at point B in Figure 13(c), representing a large delivery of the remaining data from the bicycles to the SAP. In contrast, the flat portions of the curves, e.g., in the area of point A, represent periods when cyclists 2 and 3 are disconnected from the network with no other mobile sensors acting to mule data to the SAPs. From Figure 13(c), we can also observe that the addition of a new SAP yields a non-uniform improvement in the data delivery performance. The impact of adding SAPs to the system on the data delivery delay is highly dependent on many factors including the SAP deployment density, and the location of the SAPs in relation to routes frequented by the cyclists.

4 Related Work

A number of companies (e.g., [9] [7]) have begun to offer products that integrate data from multiple sensors on a single user display, including biometric, advanced cycloperformance and GPS location data, showing the interest in quantifying performance. These range from rather limited \$40 devices to very capable \$500 devices [7]. Products with a slightly different focus offer integrated hardware and software solutions (e.g., [9]) to help cyclists with pre-ride route planning, and in-ride navigation cues via predownloaded maps combined with real-time GPS data. Others (e.g., [10]) offer offline planning software packaged with an online GPS tracking service available via a select set of cellular providers. BikeNet goes beyond any of these commercial offerings by adding environmental sensors to give context to performance. Further, the BikeNet system incorporates a dual mode wireless networking approach for data delivery to a back end analysis and visualization tier, providing usable and understandable information to users. By providing this dual mode architecture, we aim to support both delay tolerant query/delivery as well as expedited query/delivery models. BikeNet supports delay tolerant interaction for the sake of the biker doing post-ride analysis of his entire ride, and real time interaction for more time-sensitive operations by community consumers of the data.

The wearable computing and personal area networking fields have produced numerous examples [12], [13], [14], [15] of wireless networks that operate on and near the human body and interact with the wearer's surroundings or other people's devices. There has also been work in delay tolerant networking [16], [17], [18] to improve data transfer in networks that are often disconnected as BikeNet is. BikeNet synthesizes these ideas, using opportunistic rendezvous among personal computing devices (e.g., cell phones), embedded bicycle sensor networks, and sensor access points. BikeNet adds an implementation of back end data storage, analysis and visualization services, to translate raw sensor data streams into meaningful information about personal health and performance, and community health. BikeNet provides a complete opportunistic sensing system targeted at a cyclist experience application, and adds a new dimension to delay-tolerant networking by investigating both human-to-bike and bike-to-bike data transfer.

Mobile sensing systems have been proposed in other application contexts. Zebranet [1] monitors zebras wearing sensor collars using a mobile jeep-mounted radio gateway. The Cartel project [3] provides a mobile communications infrastructure based on car-mounted communication platforms exploiting open WiFi access points in a city, but unlike BikeNet it does not integrate either a sensing or sensed data analysis component. SATIRE [2] presents a software architecture for smart clothing, similar to the BikeNet three tier architecture. The MetroSense Project [30] proposed a mobile sensing system for skier-based sensing. BikeNet differs in its application scope and its inclusion of bicycles as mobile sensing platforms for personal and environmental sensing. Further, we have developed BikeView, our community oriented web-based portal for visualization and sharing of cycling health and performance information.

Tightly bound to the domain of people-centric sensing, is the issue of privacy [26]. Though the bulk of sensed data collected by the BikeNet system may seem innocuous (especially since it may be later shared with the community anyway), concerns about personal performance, and especially location tracking must be addressed. While back end data sharing is regulated by the user, in-network techniques such as muling along with the possibility of wireless snooping [29], mandate some privacy solution. While we do not integrate privacy protection into our prototype, we note the work of others can likely be leveraged. We conjecture that concepts such as virtual walls [25] can be used when deciding what information to reveal in real time via the back end or to peer BANs. Light-weight encryption (e.g., TinySec [24], MiniSec [23]) may used for intra-BAN communications and to protect against data prying on the part of mules.

5 Conclusion

In this paper, we have presented the detailed design, implementation and evaluation of the BikeNet mobile sensing system, adding to the growing body of work exploring opportunistic sensor networking techniques. BikeNet represents the first comprehensive mobile sensing system quantifying the cyclist experience. BikeNet provides for the collection and analysis of personal performance and communal environmental sampling. BikeNet supports two modes of operation in support of delay-tolerant and real-time sensing, and collected data can be presented both locally to the cyclist or to others via back end services. The BikeView portal concept promotes social networking among cyclists, and to the broader community. Initial results are encouraging and demonstrate some of the value that mobile wireless sensor networks can bring to our lives, including how we are impacted by our environment and how we can regulate our activity patterns to improve our quality of life. While our current experiments have concentrated on sensing for the cyclist and bicycle mounted sensors we conjecture that the BikeNet system could be implemented on other vehicles such as cars with little modification to the software system.

Acknowledgment

This work is supported in part by Intel Corp., Nokia, NSF NCS-0631289, ARO W911NF-04-1-031, and the Institute for Security Technology Studies (ISTS) at Dartmouth College. The authors thank Xiao Zheng and Hong Lu for their help with data collection, Peter Boda and Chieh-Yih Wan for their support of this project, and Peter Corke who served as the shepherd for this paper. ISTS support for this research is provided under award 60NANB6D6130 from the U.S. Department of Commerce and award 2005-DD-BX-1091 from the U.S. Bureau of Justice Assistance. The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the National Institute of Standards and Technology (NIST), the U.S. Department of Commerce, or the U.S. Department of Justice.

6 References

[1] T. Liu, C. Sadler, P. Zhang, M. Martonosi. Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet. In *Proc. of 2nd Int'l Conf. on Mobile Systems, Applications, and Services*, pp. 259–269, Boston, MA, USA, Jun 7-9 2004.

[2] R. Ganti, P. Jayachandran, T. Abdelzaher, J. Stankovic. SATIRE: A Software Architecture for Smart AtTIRE. In *Proc. of 4th Int'l Conf. on Mobile Systems, Applications, and Services*, pp. 110–123, Uppsala, Sweden, Jun 19-22 2006.

[3] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *Proc. of 4th Int'l Conf. on Embedded Networked Sensor Systems*, pp. 125–138, Boulder, CO, USA, Nov 1-3, 2006.

[4] BikeView - the BikeNet Web Portal. `http://bikenet.cs.dartmouth.edu`. Username: bikeview, Password: bikes

[5] Beating Heart Blog. `http://turbulence.org/Works/beatingheart/blog/`.

[6] Nike+iPod Sports Kit. `http://www.apple.com/ipod/nike/`.

[7] CicloSport HAC5. `http://www.ciclosportusa.com/hac.html`.

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

[9] Garmin Edge 305 and Garmin Emap, Garmin International, Inc. `http://www.garmin.com`.

[10] Adventure Planner, Trimble Outdoors. `http://www.trimbleoutdoors.com/biking.aspx`.

[11] TinyOS. `http://www.tinyos.net/`.

[12] M. Feldmeier and J. A. Paradiso. Giveaway wireless sensors for large-group interaction. In *Extended Abstracts on Human Factors in Computing Systems*, pp. 1291–1292, Vienna, Austria, Apr 24-29 2004.

[13] R. DeVaul, M. Sung, J. Gips, and A. Pentland. MIThril 2003: Applications and Architecture. *Proc. of the 7th Int'l Symp. on Wearable Computers*, pp. 4, White Plains, NY, USA, Oct 21-23 2003.

[14] T. Zimmerman. Personal Area Networks: Near-field Intrabody Communication. *IBM Systems Journal*, 35(No. 3,4), 1996.

[15] B. Gyselinckx, C. Van Hoof, J. Ryckaert, R.F. Yazicioglu, P. Fiorini, and V. Leonov. Human++: Autonomous Wireless Sensors for Body Area Networks. In *Proc. of 27th Conf. on Custom Integrated Circuits*, pp. 13–19, San Jose, CA, USA, Sep 18-21 2005.

[16] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A Networking Architecture Designed Around Mobile Users. *Proc. of 3rd Ann. Conf. on Wireless On-demand Network Systems and Services*, pp. 78–86, Les Ménuires, France, Jan 18 2006.

[17] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin. Intelligent Fluid Infrastructure for Embedded Networks. In *Proc. of 2nd Int'l Conf on Mobile Systems, Applications, and Services*, pp. 111–124, Boston, MA, USA, Jun 7-9 2004.

[18] Y. Wang and H. Wu. DFT-MSN: The Delay Fault Tolerant Mobile Sensor Network for Pervasive Information Gathering. In *Proc. of 25th Int'l Conf. on Computer Communications*, pp. 1-12, Barcelona, Spain, Apr 23-29 2006.

[19] Nokia N80 Mobile Device. `http://www.nokia.com/nseries/index.html`.

[20] N. Kunzli, et al. Public-health Impact of Outdoor and Traffic-related Air Pollution: A European Assessment. *The Lancet.* Vol. 356, Iss. 9232, pp. 795-801, Sep 2000.

[21] J. Orlando. Vehicle Noise and the Toll on People. `http://www.culturechange.org/issue19/vehicle_noise.htm`.

[22] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson. People-Centric Urban Sensing. In *Proc. of 2nd Ann. Int'l Conf. on Wireless Internet*, Boston, MA, USA, Aug 2-5 2006.

[23] M. Luk, G. Mezzour, A. Perrig, and V. Gligor. MiniSec: A Secure Sensor Network Communication Architecture. In *Proc. of the 6th Int'l Conf. on Information Processing in Sensor Networks*, pp. 479–488, Cambridge, MA, USA, Apr 25-27 2007.

[24] C. Karlof, N. Sastry, and D. Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *Proc. of the 2nd Int'l Conf. on Embedded Networked Sensor Systems*, pp. 162–175, Baltimore, MD, USA, Nov 3-5 2004.

[25] A. Kapadia, T. Henderson, J. J. Fielding, and D. Kotz, Virtual Walls: Protecting Digital Privacy in Pervasive Environments. In *Proc. of the 5th Int'l Conf. on Pervasive Computing*, pp. 162–179, Toronto, Canada, May 13-16 2007.

[26] P. Johnson, A. Kapadia, D. Kotz, N. Triandopoulos. People-Centric Urban Sensing: Security Challenges for the New Paradigm. *Dartmouth Technical Report TR2007-586*, Feb 2007.

[27] CitySense. `http://www.citysense.net/`.

[28] M. J. Caruso. Applications of Magnetic Sensors for Low Cost Compass Systems. In *Proc. of IEEE Position Location and Navigation Symp.*, pp. 177–184, San Diego, CA, USA, Mar 13-16 2000.

[29] T. S. Saponas, J. Lester, C. Hartung, and T. Kohno. Devices That Tell On You: The Nike+iPod Sport Kit. `http://www.cs.washington.edu/research/systems/privacy.htm`.

[30] MetroSense Project. `http://metrosense.cs.dartmouth.edu`.