# Distributed Wireless Control for Building Energy Management*

Alan Marchiori and Qi Han
Department of Mathematical and Computer Sciences
Colorado School of Mines
{amarchio,qhan}@mines.edu

## Abstract

Automated building energy management systems are essential to enabling the development of mass-market, low-energy buildings. In existing and future buildings, the impacts of occupant behaviors contribute significantly to the total energy efficiency. As building technologies and materials improve, the relative impact of behavioral factors is more significant. We propose a general framework where building systems can share information in order to optimize performance. To be successful, such a system must be responsive, intuitive, robust, and scalable. As a first step toward achieving these goals, we present a prototype distributed control system for building energy management that uses wireless sensor network-class nodes. Using protocol independent multicast, sensors and controllers are allowed to efficiently share information in a distributed peer-to-peer fashion. Our prototype system achieved an energy savings of 7.1% - 14.6% by implementing a relatively simple control policy. Based on the results of this this work we have identified three key areas for future work.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

## General Terms

Design, Experimentation, Performance

## Keywords

Multicast, Distributed Wireless Control, Peer-to-Peer

* This work was supported in part by NSF grant CNS-0855060 and the U. S. Department of Energy (through the National Renewable Energy Laboratory under contract number DE-AC36-08GO28308).



*BuildSys 2010* November 2, 2010, Zurich, Switzerland.



## 1 Introduction

The U.S. department of energy reports that buildings were responsible for 39% of the total energy consumption in the U.S. in 2009 [16]. Because energy consumption is closely tied to occupant behavior, numerous building monitoring systems have been recently developed [1, 7, 9] to provide occupants with detailed energy consumption information. The impact of this data is significant; however, monitoring alone does not always result in savings. A recent study observed an initial 31.9% reduction in energy consumption immediately after installing a monitoring system; however, after a month the reduction fell to only 3.7% [10]. This illustrates that while significant savings are possible, relying on occupants to change their long-term behavior may be difficult. One alternative solution is to build systems that automate the energy saving behaviors.

Automated building management systems (BMS) are expected to save an average of 5% to 10% in residential households [2, 3, 15]. Although these savings are significant, it equates to only $5-$10 per household per month [17]. To be practical, a BMS must pay for itself within a few years, which means it should cost no more than a few hundred dollars. As a result, deploying a WSN with sensors and actuators dedicated exclusively for building management is too costly. However, if we leverage existing sensors already in the home, we can significantly reduce the cost of the BMS. For example, many homes have security systems which sense the states of doors and windows and detect motion. Existing HVAC systems sense temperature. Everyday household appliances have numerous on-board sensors ranging from the simple refrigerator door switch to the complex sensing techniques possible with an idle PC's microphone and camera. It is even possible to collect device-level energy usage in many appliances for free [4]. Of course none of these systems currently share this very useful information. One reason is that there are no standards defining how to share and consume this information.

Our proposed solution uses a wireless sensor network (WSN) to share this information. Wireless sensor networks utilize low-powered low-cost wireless nodes communicating over an ad-hoc network. Standardization is emerging in the form of IEEE 802.15.4 [8] and 6LoWPAN [14]. The dominant communication paradigm in WSNs is from the sensor nodes to a base station for processing and storage. Using this approach we could easily construct a centralized build-

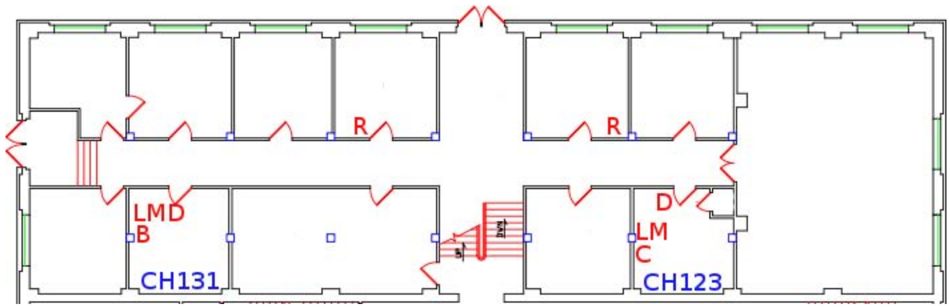



**Figure 1. A building management system prototype deployment area. The offices with energy controllers are CH131 and CH123. Letters indicate the approximate location of nodes. Node types are abbreviated as: Light, Motion, Door, Energy Controller, Relay, and Base-station.**

ing management system that would process all sensed data and make optimized control decisions. However, for a BMS to be more responsive, intuitive, robust, and scalable, a distributed approach is essential. We define a distributed BMS as one where control decisions are made locally at the point of control using information received directly from any other nodes in the network. For example, a light might receive sensor data from a wall switch, motion sensor, and light sensor. The control policy could be to turn on the light if the switch is on and there has been motion in the last 15 minutes and the ambient light level is below 200 lux; otherwise the light will be off to save energy.

Implementing a WSN-based distributed control system requires an efficient means of sharing information between devices. The two general approaches to information sharing are to pull or push the data. In a pull-based system, controllers would periodically poll the sensors, pulling the relevant information into the controller. This places the burden on the control point to collect necessary information in a timely manner while the sensor only needs to respond to requests. In a push-based system, the sensors disseminate information to the controller when it is available. This makes the controller's job much easier by transferring more responsibilities to the sensors. The ZigBee Smart Energy Profile 2.0 allows both forms of information sharing [18]. For both cases it is implemented in the application layer with sequential unicast communications, which creates redundant messaging when two nearby nodes are consuming the same information. For example, two (or more) lights in the same room might rely on the same set of sensors. Using unicast communication requires unique messages for each sensor used by each controller. However, the broadcast nature of wireless communication, makes it possible to improve efficiency by allowing any interested node within a shared communication area to receive the same information. We have implemented this approach using standard IP multicast that we have adapted for WSNs. The result is that sensors can push information that is then efficiently delivered to all interested control points. The use of IP multicast distributes the responsibility for information sharing to the *network* rather than either the sensor or controller. Because this is implemented at the network layer, redundant packet transmissions can also be eliminated, which improves energy efficiency, information timeliness, and network utilization.

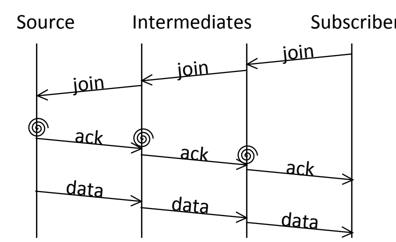



**Figure 2. Illustration of the basic PIM processes. Spirals indicate when a node updates its subscription list. Each node then uses its subscription list to determine if it must forward data.**

## 2 Efficient Multicast for IPv6 WSNs

Our multicast implementation is named PIM-WSN. PIM stands for Protocol Independent Multicast [5], the most common multicast found in wired IP networks. Figure 2 shows the basic operation. To receive multicast data an interested node (subscriber) sends a unicast join message to the source. The source responds with a unicast acknowledgement. Nodes along the path of the acknowledgement become forwarding nodes. If overlapping forwarding paths are set up only one packet is transmitted on shared links, improving efficiency.

The novel feature of PIM-WSN that makes it well-suited for WSN-based BMSs is constant memory usage with an unlimited number of source and subscriber nodes. This allows networks to easily scale from a few nodes to hundreds, without the need to reconfigure or recompile. A detailed analysis of PIM-WSN is available in [12] where we show that PIM-WSN achieves 1) high packet delivery rate (over 97%), 2) low latency per hop (less than 5 ms), and 3) lower radio utilization than three other multicast protocols (by more than 50%). To implement PIM-WSN we use TinyOS 2.x with the Blip IPv6 networking stack as base. Compiled for the TelosB platform, PIM-WSN requires an additional 5,978 bytes of ROM and 235 bytes of RAM.

## 3 Prototype System

Our prototype BMS using PIM-WSN is currently deployed in two graduate student offices on our campus. The deployment is depicted in Figure 1. The portion of the build-

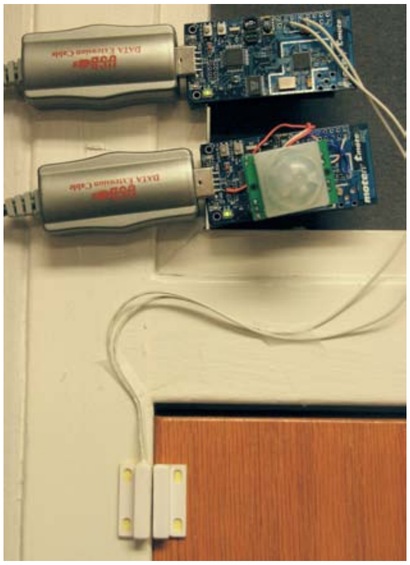


**Figure 3. Door and PIR motion sensors.**

ing shown is approximately 120 feet by 40 feet. We configured the transmit power on each mote to -10 dBm to simulate a physically larger and more interesting network topology. This results in a maximum hop count of 5 from the base station in CH131 to the motes in CH123. The control algorithm is distributed and implemented directly on each energy controller. The base station is required by Blip to provide multihop routing for unicast packets. Multicast data is then forwarded along the routes selected by unicast routing protocol.

## 3.1 Sensors and Controllers

Each office is outfitted with three sensors: 1) passive infrared (PIR) motion sensor, 2) door sensor (magnetic reed switch), and 3) ambient light sensor. To consume this data, each office has an energy controller node. The energy controller plugs into a standard electrical outlet and provides two outlets: one switched and one non-switched. The energy controller also contains a power meter that measures real-time power usage and total energy usage independently for each outlet. A power strip is plugged into each outlet and all essential devices (PCs, refrigerator, network equipment, etc.) are plugged into the non-switched power strip. Non-essential devices (LCD, printer, coffee pot, microwave, etc.) are plugged into the switched power strip. The sensors and energy controller each use a modified TelosB mote programmed with TinyOS 2.x and our implementation of PIM-WSN. The premise is that the energy controller will detect and process the available sensor data and intelligently control the switched outlet to save energy by switching off non-essential devices when the office is unoccupied.

**1) Motion sensor**

The motion sensor is a PIR sensor (Parallax #555-28027) with a motion detection range of approximately 20 feet (lower mote in Figure 3). It is configured to send repeated pulses when there is continuous motion. The sensor output is attached to the TelosB's expansion connector on an interrupt-enabled GPIO pin. This allows the sensor to wake up the TelosB when motion is detected. Every transition of the GPIO pin causes a single multicast packet to be transmitted indicating the motion sensor output (motion or no motion). Because the sensor requires 3.3V-5.0V to operate reliably, it is powered directly from the USB interface.

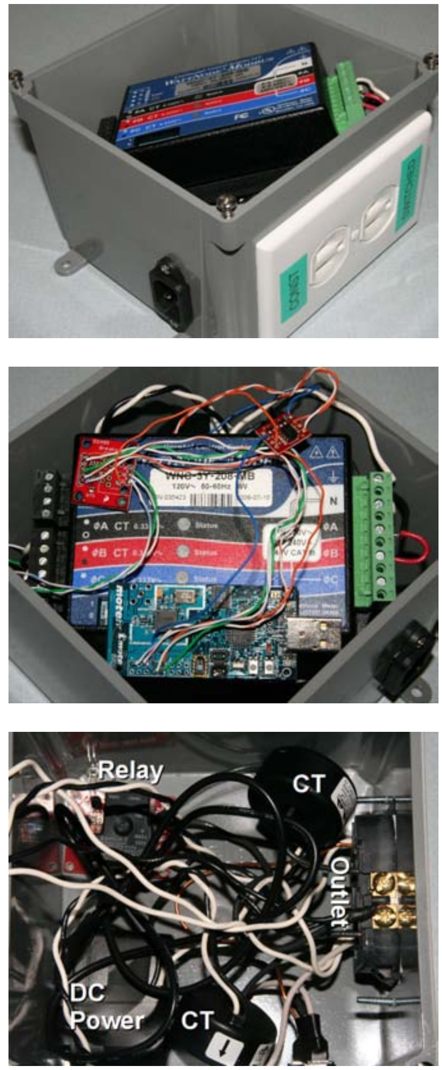



**Figure 4. Energy controller node.**

**2) Door sensor**

The door sensor is a magnetic proximity switch (C&K Components #MPS45WGW) attached to each office door (upper mote in Figure 3). The switch is interfaced to the TelosB in the same configuration as the motion sensor. An interrupt-enabled GPIO pin allows the sensor to wake the mote when the state of the door changes. When the switch changes value a single multicast packet is sent indicating the current state of the door (open or closed). Although multicast packet delivery is generally very reliable, it is not guaranteed. Because it is likely to only send one or two packets (unlike the motion sensor that generally sends several), we also use a periodic timer to send the door state once per minute. This allows the energy controller to miss packets and still maintain acceptable functionality.

**3) Ambient light sensor**

The ambient light sensor is the TelosB's on board Hamamatsu S1087 photodiode read by the MSP430's internal ADC. The photodiode is polled randomly once every $0.75 \pm 0.25$ seconds. Typical ADC readings range from 0 (dark) to the several hundred for normal office lighting. If the sampled light value is more than 25 higher or lower than the last reported value it is multicast immediately. Otherwise, one sample is sent every 100 readings. This results in a data rate of at least one packet every $75 \pm 25$ seconds while still being responsive to rapid changes, such as the lights being turned on or off.

**4) Energy Controller**

The energy controller is a TelosB with a Tyco relay (# T9AS1D12-5) that is rated for 220V@30A and a WattNode[1] energy meter to allow power monitoring. We use a standard `6"x6"x4"` electrical box to house this equipment. Power comes in through an IEC C14 connector, passes through a 15A current transformer (CT), and travels in parallel to the non-switched side of a standard NEMA 5-15 outlet and the AC relay. The output of the relay passes through another 15A CT and then on to the switched side of the outlet. The WattNode, relay, and TelosB are powered from the non-switched AC supply so the energy controller's power usage is included in the non-switched power measurement. Because of this configuration, we are measuring total power and switched power; the non-switched power usage is simply the difference between the two measurements. The power meter is configured to average instantaneous power readings over a 20 second sliding window. The TelosB then samples the real (W) and reactive (VAR) power, line voltage (V), frequency (Hz), and total energy used (kW h) once every five seconds for each phase (total and switched). The sensor data is then augmented with the current occupancy value (true/false) and encoded in a single packet and transmitted sequentially as a multicast packet and as a serial packet (for logging).

Figure 4 shows the energy controller. The top figure shows the power input and outlets. The middle figure is TelosB with interface circuitry. The WattNode uses the Modbus [13] serial communications protocol over an EIA-485 physical link, so an EIA-485 adapter was added on `uart0` of the TelosB. The relay requires 200 mA to activate, so a supplemental AC/DC power supply was also added. Because the mote was powered via USB (to collect diagnostic information) an opto-isolator was used to interface to the relay (the WattNode's EIA-485 interface is already isolated). The internal wiring is shown in the bottom figure.

## 3.2 Control Algorithm

Each energy controller is preprogrammed with the room number it is deployed to so that it can search for sensors (see Section 3.3) in the same room (each sensor is also programmed with its room number). When a door, motion, or light sensor is detected in the same room, the energy controller subscribes to that node's multicast and begins receiving sensor data. Our control algorithm (Figure 5) relies on detecting when the occupancy changes and then switching the relay on or off accordingly. Occupancy detection is a difficult problem and not our focus, so we use a simple but effective algorithm tailored to our office environment. Each office has a single door. We assume that the office is occupied if the door is open and that the door is shut when unoccupied (this is nearly always true). As a result, occupancy can only change after a door-close event. Therefore, when the door is open, the energy controller switches to the occupied mode. After detecting a door-close event it starts a 60-second timer. While this timer is running it counts the number of motion events received. When there is constant motion the motion sensor will send one motion event per second, but even when there is no motion one or two (false) motion events per hour. To reduce the impact of false motion events, we use a threshold of 5 or more motion counts in the 60-second interval to indicate that the room is occupied. In practice this algorithm works very well in our offices and could be easily applied in residential homes by interfacing to a security system with door and motion sensors.

```
if door open then
  // room is occupied
  setRelay(close)
else
  // assess occupancy
  count = 0
  for 60 seconds do
    wait for motion event or timeout
    count += 1
  end for
  if count ≥ 5 then
    // room is occupied
    setRelay(close)
  else
    // room is not occupied
    setRelay(open)
  end if
end if
```

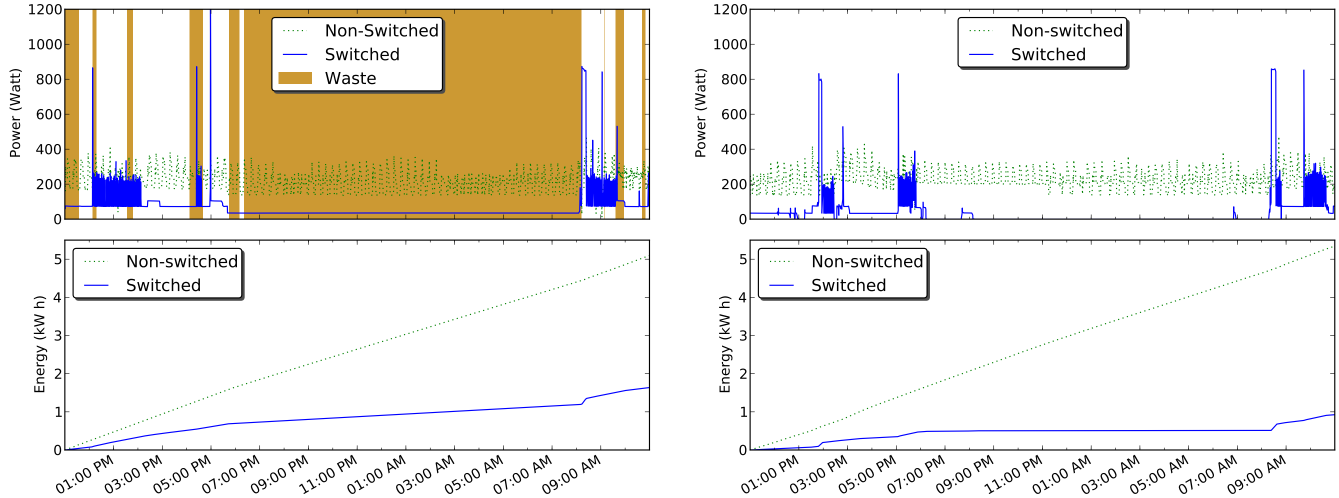
**Figure 5. An occupancy detection algorithm executed each time the door state changes**

## 3.3 Service Discovery

One remaining challenge is to decide how the energy controller initially subscribes to the multicast from each sensor. One approach is to hard-code the source address of each sensor, by definition this is not a very flexible solution. Instead, we have implemented a service discovery protocol that is similar to the Simple Service Discovery Protocol (SSDP) [6]. SSDP is used by Universal Plug and Play (UPnP) to detect other UPnP devices. It uses HTTP formatted messages over a predefined multicast group. In our implementation we use two fixed format messages, rather than the variable format HTTP messages, to simplify processing. We have also assigned a special multicast group in PIM-WSN where all nodes are assumed to subscribe. This effectively allows PIM-WSN to broadcast the service discovery messages to every node in the network. The two messages are: *advertisement* and *query*. Common to both of the messages are two 8-character fields defining the sensor type and domain. The sensor types we used are: motion, light, door, and energy. The domain is used to indicate the room number of the sensor: CH123 or CH131.

There are two ways to detect a sensor. The first is at

[1] `http://www.ccontrolsys.com/products/wattnode_modbus.html`




(a) Monitoring only. (b) Monitoring and control.

**Figure 6. Experimental results over a typical day.**

startup, because up it will initially transmit several *advertisements* to the service discovery multicast group. If a node is interested in the advertised sensor it can then join the multicast immediately. The second way to detect a sensor is to have the interested node send a *query* message to the service discovery multicast group. The query message allows wildcard searches on the type and domain values. All nodes that receive the query will check to see if their service description matches, and if so, the node replies with a unicast advertisement.

## 4 Experimental Results

Figure 6 shows the power and cumulative energy usage logged by the energy controller in CH131 over two typical days. On day one (Figure 6a) the relay was disabled to detect and compute wasted energy. On day two (Figure 6b) the relay was enabled to control the wasted energy. The non-switched devices are: two PCs, one laptop, two small refrigerators, and an Ethernet switch. The switched devices are: two LCD displays, a laser printer, two powered speakers, a desk lamp, a microwave, and a coffee pot. The large spikes in the switched data are due to the coffee pot, microwave, and laser printer. The oscillations in the non-switch data are due to the refrigerators.

Figure 6a shows the first day where the room was occupied for 7h 52m 46s. The switched devices consumed a total of 1.6362 kW h and the non-switched devices consumed 5.0972 kW h. Of the switched total, 0.7008 kW h was consumed (wasted) while the room was unoccupied. The minimum power usage was 33 watts. This reveals a potential savings of 42.8% of the total energy used by switched devices or equivalently 10.4% of the total (switched plus non-switched) measured energy consumption. Performing the same analysis on the other office yields a potential savings of 6.89%.

On the next day the relay was enabled and the results are shown in Figure 6b. On this day the room was occupied for 9h 56m 21s. The switched devices consumed a total of 0.9120 kW h wile the non-switched devices consumed 5.3426 kW h. The non-switched devices consumed 4.8% more on this day, most likely due to the increased occupancy. Despite this fact, the switched devices now consumed 0.7242 kW h less than the previous day. The total energy consumption (switched plus non-switched) was 6.2546 kW h or 7.1% less than the previous experiment. The same analysis for the other room shows that the total energy consumption was reduced by 14.6%.

On the first day we measured that 0.7008 kW h of electricity was wasted. On the second day we controlled the devices to reduce waste and the total measured energy usage was 0.7242 kW h less than the total on the previous day. These results are very consistent between the two days. If we then assume an average daily savings of 0.7 kW h and then multiply by 365 for a conservative estimate of the yearly savings (because unoccupied time, and therefore savings, is expected to be greater on weekends and holidays) the result is 255 kW h. Then, if we assume this savings is typical over all 41 offices in our building, the estimated building-wide savings becomes approximately 10 MW h per year. This equates to an annual reduction of approximately 7.8 tons of $CO_2$ [2] and a savings of approximately $1,000.

Over the last year our building's total energy consumption was approximately 300 MW h; however, this includes HVAC and lighting. We could expand our system to include these systems or for calculation exclude them from the total energy usage. According to [16] in an average building HVAC and lighting represent 53% of the building's total energy consumption. This can be used to compute the total energy consumption excluding HVAC and lighting as 141 MW h. Fully deployed, our BMS is expected to reduce this by 10 MW h, or over 7%. Because this value is close to our achieved savings, it gives confidence that the offices used in this study are representative of the average energy consumption in the building.

If we assume the building already has sensors able to

[2] 1.5 lbs $CO_2$ per kW h

detect room occupancy (and they share this data), the only additional hardware required to implement this system is a simplified energy controller node for each office (the energy monitor function is not needed). To achieve a one year payback period (assuming 0.7kW h per day savings), the resulting budget is $25 per node. This is more than the cost of our TelosB motes, but, commercial IEEE 802.15.4 devices like the XBee are currently priced around $20 each. The relay that we used is currently priced at $1.40 each. This gives us confidence that this type of distributed control system could achieve a one year payback period.

## 5 Conclusion and Future Work

Advanced building management systems will eventually become common in residential and commercial buildings because occupant behaviors have a significant impact on the total energy consumption. To be successful these systems must be responsive, intuitive, robust, and scalable. Our approach is a fully distributed architecture using WSN-class nodes coupled with an efficient multicast communication protocol. This allows each controller to autonomously locate and receive relevant sensor information from other nodes in the network. Because control decisions are made at each control point, if a sensor or communication link fails the controller can still make reasonable control decisions. Our prototype system achieved an energy savings of 7.1% - 14.6% by implementing a relatively simple control policy. Based on the results of this this work we have identified three key areas for future work.

**Recovery from network disconnections** - Although PIM-WSN achieves good packet delivery ($>97\%$ under normal conditions [12]), missing just one packet can cause a control algorithm to fail. In our case the "door open" packet was crucial to receive or the occupant could walk into a room with all their devices powered off. We consider any packet delivery failure as a network disconnect, even if it is a transient event. To solve this, first we need a robust way to detect these disconnections. Second, once the node regains communication the missed packets should then be delivered to the node. This is reminiscent of the Trickle algorithm [11]. To be applied in this domain the algorithm must support rapidly changing data from many sensors in the network.

**Low power multicast** - To minimize the number of packets sent, PIM-WSN uses one-hop broadcast messages to deliver packets to multiple nodes simultaneously. Most modern low-power protocols focus on unicast rather than multicast or broadcast and as a result their performance in these cases is greatly reduced. In order to support battery powered or energy harvesting sensor nodes, efficient low power communication is essential. Synchronized low-power protocols represent one approach to alleviating this problem.

**Advanced control strategies** - Our prototype control strategy was admittedly very simplistic and only achieved energy savings while the room was unoccupied. The distributed control architecture is capable of implementing much more complex control strategies. We must identify and implement control policies that can reduce energy consumption even in occupied spaces. We are currently examining how different behaviors affect energy consumption in occupied homes. If we are able to identify behaviors that are common in low energy homes we can then devise automated control strategies to replicate these behaviors across all homes.

## 6 References


[1] Y. Agarwal, T. Weng, and R. K. Gupta. The energy dashboard: Improving the visibility of energy consumption at a campus-wide scale. In *Proceedings of The First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys)*, 2009.

[2] R. Anderson and D. Roberts. Maximizing residential energy savings: Net zero energy home technology pathways. Technical Report NREL/TP-550-44547, National Renewable Energy Laboratory, 2008.

[3] C. Barley, C. Haley, R. Anderson, and L. Pratsch. Building america system research plan for reduction of miscellaneous electrical loads in zero energy homes. Technical Report NREL/TP-550-43718, National Renewable Energy Laboratory, 2008.

[4] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler. Energy metering for free: Augmenting switching regulators for real-time monitoring. In *Proceedings of the 7th international conference on Information processing in sensor networks (IPSN)*, 2008.

[5] B. Fenner, M. Hadley, H. Holbrook, and I. Kouvelas. Protocol independent multicast - sparse mode (PIM-SM): Protocol specification (revised). RFC4601, 2006.

[6] Y. Goland, T. Cai, P. Leach, Y. Gu, and S. Albright. Simple service discovery protocol/1.0. draft-cai-ssdp-v1-03, 1999.

[7] Google Inc. Google PowerMeter. `http://www.google.org/powermeter/`, accessed 2009.

[8] IEEE Computer Society. Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). `http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf`.

[9] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In *Proceedings of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2009.

[10] X. Jiang, M. V. Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2009.

[11] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

[12] A. Marchiori and Q. Han. PIM-WSN: Efficient Multicast for IPv6 Wireless Sensor Networks. *Under review*.

[13] Modbus Organization, Inc. MODBUS protocol specification. `http://www.modbus.org/`, accessed 2010.

[14] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 packets over IEEE 802.15.4 networks. RFC4944, 2007.

[15] B. Neenan. Residential electricity use feedback: A research synthesis and economic framework. Technical Report 1016844, Electric Power Research Institute, 2009.

[16] U.S. Department of Energy. Buildings energy data book. `http://buildingsdatabook.eren.doe.gov/`, 2009.

[17] U.S. Department of Energy. Electric power monthly. `http://www.eia.doe.gov/cneaf/electricity/epm/epm_sum.html`, accessed 2010.

[18] ZigBee Alliance and HomePlug Powerline Alliance. Smart Energy Profile 2.0 Technical Requirements Document, April, 2010.