

# iPower: An Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks

Lun-Wu Yeh, You-Chiun Wang, and Yu-Chee Tseng

**Abstract**—Wireless sensor networks (WSNs) provide a convenient way to monitor physical environments. Exploiting the context-aware capability of WSN to achieve energy conservation in intelligent buildings is an attractive direction. We thus propose an *intelligent and personalized energy-conservation system by wireless sensor networks (iPower)* that combines WSNs and appliance control devices to provide personalized energy conservation services. A WSN is deployed in each room to monitor the usage of electric appliances and to help determine if there are electric appliances that can be turned off to conserve energy. The iPower system is intelligent and can adapt to personal need by automatically adjusting electric appliances to satisfy users' requirements. The design and implementation details of iPower are reported in this paper.

**Index Terms**—context awareness, energy conservation, sensor network, smart environment, wireless communication.

## 1 INTRODUCTION

THE discovery of electricity is one of the most important milestones in human history. Electricity is so essential in our daily life that many people cannot live without it. However, today, energy has been overly used and energy shortage has become a global concern. According to the report in [1], more than one third of electricity is spent on *HVAC systems*, which include heating, ventilating, air conditioning, lighting, and other related equipments. According to experiences, a large portion of energy consumed by HVAC systems is due to improper use of electric appliances. Therefore, how to exploit the context information of an environment to automatically control the usage of electric appliances has a great potential to reduce the waste of energy.

In this work, we propose an *intelligent and personalized energy-conservation system by wireless sensor networks (iPower)* to reduce energy consumption of HVAC systems by exploiting the context-aware capability of sensors. In iPower, WSNs are deployed in rooms of a building to collect environmental information. Such information is reported to a control server to determine whether to turn off unnecessary electric appliances. Such a system needs to be designed with user friendliness in mind to minimize the involvement of users in making decisions. As an example, when sensor nodes detect a low temperature or a high brightness in a likely unoccupied room, they can report to the server that the electric appliances in that room (e.g., air conditioners or lights) could be turned off. The server then sends an alarm signal to notify people in the room that the electric appliances could be turned off shortly. If there are still users in that room, they can signal the system that these appliances should not be turned off by triggering some events (such as making some voices, changing the light reading of any sensor, or moving any furniture attached with sensors). If there is no such intentional events made by human being detected in a predefined amount of time, the server will turn off the electric appliances through some power-line control devices.

In this way, the iPower system can work even if users are not wearing any particular badge.

In iPower, we also provide *personalized services* where electric appliances can be automatically adjusted to satisfy users' preferences. In particular, each user can create a profile to describe his/her favorite temperature and brightness. Such users are considered priority users and need to carry user identification devices so that our system can retrieve their profiles. When there are priority users in a room, the server will adjust the air conditioners and lights in that room according to the profiles of these users.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 presents the design of our iPower system. Section 4 gives the implementation experiences. Section 5 gives some simulations to evaluate our system. Section 6 concludes this paper and discusses some research issues in our system.

## 2 RELATED WORK

WSN has been widely used to provide context information in smart spaces/environments. How to automatically control electric appliances according to users' locations and their requirements has been intensively discussed for smart homes/offices. The work in [2] considers a ubiquitous computing architecture in which electric appliances are controlled by a SIP (session initiation protocol) [3] server, under which architecture users can make calls to communicate with the SIP server to control their electric appliances. In the *MaoHome* system [4], the mobility pattern of a user in a house is exploited and is forwarded to the system to provide advanced services (e.g., controlling the corresponding electric appliances) in the predicted locations of the user. In *Semantic Space* [5], the authors propose some semantics to describe the environment, which can be used to query the status of the environment where users are located. The work in [6] proposes a context-aware smart house in which electric appliances can be automatically adjusted according to the environmental information collected from sensors. Our work is motivated by observing that the issue of energy conservation, which is very critical to our environment, has not been well addressed.

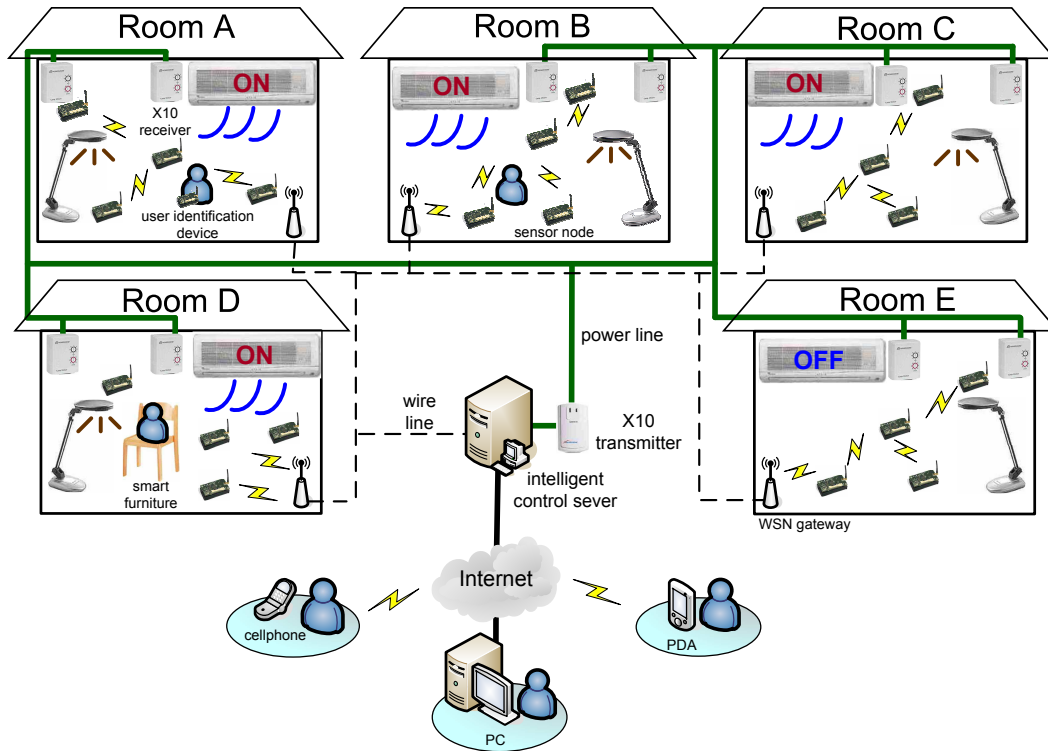


Fig. 1: System architecture of iPower.

### 3 DESIGN OF THE IPOWERSYSTEM

#### 3.1 System Architecture

The system architecture of iPower is illustrated in Fig. 1, which consists of many sensor nodes, several WSN gateways, an intelligent control server, some power-line control devices, and user identifications devices. Below, we describe the functions of each component separately.

**Sensor nodes:** In each room, we deploy sensor nodes to monitor the environment. These nodes will form multi-hop WSNs to collect information in the rooms. In our prototype, three types of sensing data are collected: light, sound, and temperature. An *event* is defined when the sensory input is higher or lower than a predefined threshold. To conserve the energy of sensor nodes, reporting events is reactive, in the sense that a node reports its sensing data only when some predefined events occur. Different events can be combined to describe a room's condition. For example, a low temperature (or a high brightness) together with some sound events in a room may indicate that the corresponding electrical appliances are turned on to serve users in that room; some sound events and change of the light degree may indicate that users in that room are moving around; and a low temperature (or a high brightness) with no sound event for a certain amount of time may indicate that the air conditioners (or lights) in that room are unnecessarily turned on because no one is in the room. We can include more types of sensors to provide more intelligence. For example, as shown in Fig. 2, a smart desk may include some pressure sensors underneath the cushion of a chair and some light sensors nearby the lamp on the desk. When someone is sitting on the chair, such an event can be detected by the pressure sensors, and the system can adjust the lamp based on the light degree nearby the lamp. When the user leaves the chair, the pressure sensors can detect

the disappearance of the user and make energy conservation decision by notifying the server to turn off the lamp.

**WSN gateways:** The set of sensor nodes in each room will form a WSN. For each WSN, there is a WSN gateway. A WSN gateway has a wireless interface to communicate with sensor nodes and a wire-line interface to communicate with the intelligent control server. It has four major functionalities: issuing commands to sensor nodes, gathering data from sensor nodes, reporting the room's condition to the intelligent control server, and maintaining the WSN. Specifically, the gateway will notify sensor nodes in the WSN to begin collecting environmental information when it receives a *start* command from the server. After gathering sensing reports from the WSN, the gateway will determine the room's condition and report to the server. In order to maintain the WSN, the gateway will periodically broadcast a *heart-beat* message to the network. A sensor node receiving such a message will reply an *alive* message to the WSN gateway. If the gateway does not receive any *alive* message from a sensor node for a predefined amount of time, it will notify the server that the node may be broken.

**Intelligent control server:** The intelligent control server is used to collect the system's status (e.g., rooms' conditions and sensors' states) and to perform power-saving decisions. It maintains a database of user profiles and periodically checks the states of electric appliances in each room. It will decide whether to turn off an electric appliance in a room according to the sensory data collected from that room. The server can also adjust the electric appliances in a room according to the profiles of users in that room. Such decisions or adjustments are achieved by sending commands through the power-line control devices to turn off or adjust electric currents of the corresponding electric appliances. The server also provides user interfaces to allow users to maintain the iPower system. In particular, users can modify their profiles and obtain the system's status through remote devices.

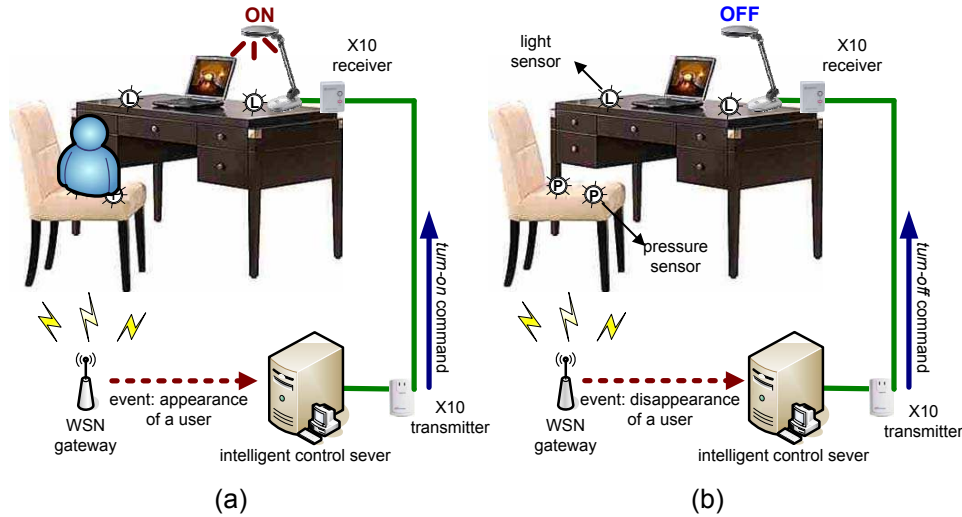


Fig. 2: The smart desk scenario. (a) The user sits on the chair. (b) The user leaves.

**Power-line control devices:** The power-line control devices allow the system to turn on/off or adjust the electric currents of appliances. In our current prototype, we adopt the X10 devices produced by SmartHome [7]. Such devices contain one X10 transmitter and several X10 receivers. The X10 transmitter can talk to X10 receivers via power lines. In the iPower system, the X10 transmitter is connected to the control server to transmit the server's commands.

**User identification devices:** The user identification devices are portable devices that can be carried by users so that the system can determine users' IDs and retrieve their profiles. It can be any identification device. In this work, we simply use the processor board of our sensor platform (without sensors) for user identification. When a user enters a room, his/her user identification device will join the WSN in that room and provide its ID to the server via the WSN gateway.

### 3.2 Energy Conservation Scenarios

Next, we give five scenarios to demonstrate how iPower works in an intelligent building. Let us consider the five rooms in Fig. 1.

**Room A: electric appliances are turned on and somebody is in the room (with a user identification device).** In this case, since the system can detect that the room is occupied, energy conservation commands will not be issued. So the electric appliances in room A will remain on.

**Room B: electric appliances are turned on and somebody is in the room (without a user identification device).** In this case, energy conservation commands will be given depending on whether some events (such as sound events) indicating that the room is occupied can be detected or not. If there are such events, the electric appliances will remain on. Without such events, some signals (such as beeps or blinking lights) will be triggered to warn users in that room. In response, users can do some actions to signal the system that the room is occupied (such as making some noise by clapping, covering any sensor with a light sensor to change its light reading, or switching on or off any electrical appliance that is under control of the iPower system). As long as any of such events can be detected, the server can realize that the room is still in use and thus will not turn off the electric appliances. Note that to avoiding bothering users too much, the interval to warn

users next time will be increased in an exponential manner after each intentional event being generated by users in that room. Further, after several warning signals without success, the system will stop trying (to make energy conservation decisions) for a long period of time.

**Room C: electric appliances are turned on but nobody is in the room.** In this case, since sensor nodes have detected a low temperature, a high brightness, and no sound event for some while, the WSN gateway will report to the control server that this room is *abnormal*, implying that electricity may be wasted in room C. The server will then send an *alarm* message to room C, which triggers the beepers attached to sensor nodes. These beeps are used to announce that the system will turn off air conditioners and lights in room C in a few minutes. Alternatively, we can blink lights on and off to signal users that appliances in that room will be turned off soon. This is to avoid our system to make wrong decisions. Since there is no one in the room, the server will turn off these appliances after timeout to conserve energy.

**Room D: electric appliances are turned on in the room with smart furniture.** If there is smart furniture in the room, they can help detect the existence of people in that room. For example, if there is a person sitting on a smart chair, the system keeps on reporting that someone is on the chair, so no energy conservation command is issued. If the smart furniture is not in use, the scenario in room B may be applied.

**Room E: electric appliances are turned off.** In this case, the WSN gateway will report to the server that the room is *normal* so the server will not take any action.

### 3.3 System Operations and Message Flows

Fig. 3 illustrates the message flow and the interaction of system components in the iPower system. The details are discussed below.

- 1) The control server starts checking the usage of electric appliances in a room by sending a *start* message to the WSN gateway in that room. Checking can be done periodically or at predefined time, according to the system configuration file.
- 2) On receiving the *start* message from the server, the WSN gateway notifies its sensor nodes by issuing

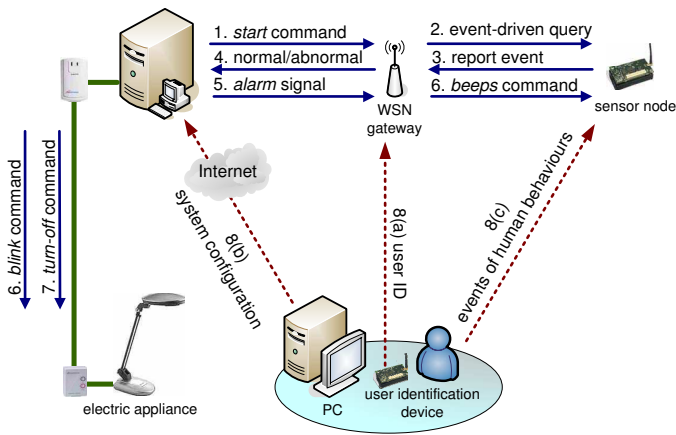


Fig. 3: Message flows in the iPower system.

some event-driven queries to collect information from the environment. The WSN gateway then sets a timer to wait for sensing reports from sensor nodes.

- 3) When a sensor node detects any event (such as a low temperature or a high brightness), it reports its sensing data to the WSN gateway.
- 4) If the WSN gateway receives any sensing report and any human behavior report from step 3 before its timer expires, it can determine the room's status according to the following rules:
  - a) If any piece of smart furniture reports that someone is using it (e.g., the case in Fig. 2(a)), the WSN gateway reports a *normal* status to the server. However, if it is reported that users leave the smart furniture, the WSN gateway resets its timer and goes back to step 2 to repeat the above procedure.
  - b) If sensor nodes report any human behavior (such as sound events or change of light readings), the WSN gateway reports a *normal* status to the server. However, it also notifies the existence of people to the server so that the system will check this room's status later on.
  - c) Otherwise, the WSN gateway reports an *abnormal* status to indicate that the electric appliances in the room may be turned on unnecessarily.
- 5) When the server receives an *abnormal* report from the WSN gateway, it warns the people (if any) in the corresponding room by sending an *alarm* message to the WSN gateway.
- 6) Once receiving the *alarm* message, the WSN gateway instructs one of its sensor nodes to turn on its buzzer to generate a beeping sound. Alternatively, the server can send a *blink* command to the X10 receiver to blink any light on and off for a short period of time. These actions are used to notify people in the room that the server will turn off the electric appliances after a short period of time (e.g., ten minutes).
- 7) If the server does not receive any human behavior event from the room after a predefined period of time, it knows that there is no one in that room and thus turns off the electric appliances by sending a *turn-off* command to the X10 receivers in that room.
- 8) If there is any user in the room hearing the beeping

```
<? xml version="1.0" encoding="UTF-8" ?>
- <User>
  <Id>007</Id>
  <Name>HSCC</Name>
- <Attribute>
  <Name>temperature</Name>
  <Value type="Float"/>
  <Range max="28" min="25">true</Range>
</Attribute>
- <Attribute>
  <Name>light</Name>
  <Value type="Float">70</Value>
  <Range max="0" min="0">>false</Range>
</Attribute>
</User>
```

Fig. 4: An example of the user profile.

sound or seeing blinking light, he/she can notify the server that the room is still in use by any of the following three methods:

- a) If the user has carried a user identification device, the device will directly inform the server (via the WSN gateway) his/her ID. In this case, the user does not need to take any action.
- b) If the user can access the Internet, he/she can login the web page of iPower to set up the next checking time of this room so that the server will not disturb the user before he/she leaves the room.
- c) Otherwise, the user can make some intentional events by changing the room's environment, such as making some noise by clapping or turning off and then turning on any light. In this way, sensor nodes will detect an unusual sound or change of light degree and thus report these events to the WSN gateway.

According to these reports, the WSN gateway can notify the existence of users to the server and thus the system will back off and check the room's status later on. The next checking time can be set manually by users, by any default value (such as one hour), or in any typical exponential backoff manner.

### 3.4 Personalized Services and User Profiles

The iPower system also provides personalized services in which electric appliances can be automatically adjusted to satisfy users' preference. In particular, each user can specify his/her favorite temperature and brightness. When a user enters a room, iPower can adjust the air conditioners and lights to meet the user's preference. To achieve this goal, the user has to create a profile in the server's database and carry a user identification device when entering our system. The user's location is determined by the WSN gateway which collects the user's ID.

In our current implementation, we follow the format of XML (*extensible markup language*) [8] to describe user's profiles. The current definition is illustrated in Fig. 4. Specifically, the profile includes user's ID, name, and several attributes with the user's favorite temperature and brightness. For example, Fig. 4 indicates that user's preference temperature is from 25°C 28°C and light is 70 lux.

```

<iPowerRule> = [ User <UserID> ] On <Time> Condition <Event> Do <Action>
<UserID>    = string
<Time>     = min/hr/date/mon/yr | anytime
<Event>    = <Sensor> AND <Event> | <Sensor> OR <Event> |
             <DeviceStatus> AND <Event> | <DeviceStatus> OR <Event> |
             <Sensor> | <DeviceStatus>
<Sensor>   = <SensorID> <SenseData>
<SensorID> = string
<SenseData> = <SenseType> <Range>
<SenseType> = temperature | sound | pressure | humidity | light | rssi
<Range>    = <Max> To <Min>
<Max>     = integer | float
<Min>     = integer | float
<DeviceStatus> = <DeviceID> <DeviceInfo>
<Action>   = <Device> AND <Action> | <Device>
<Device>  = <DeviceID> <DeviceInfo>
<DeviceID> = string
<DeviceInfo> = <DeviceAction> AND <DeviceValue> | <DeviceAction>
<DeviceAction> = on | off
<DeviceValue> = integer | float
    
```

Fig. 5: The iPower's EBNF-like recursive grammars.

```

Room A:
  User userID_1
  On anytime
  Condition ( sensorID_1 temperature 28 To 50 )
  Do device_aircon on
Room C:
  On anytime
  Condition ( sensorID_2 rssi 0 To 40 ) AND ( sensorID_2 rssi 80 To 100 ) AND
            ( device_lamp on )
  Do device_lamp off
Room D:
  On anytime
  Condition ( sensorID_3 light 0 To 20 ) AND ( sensorID_4 pressure 10 To 100 )
  Do device_lamp on
    
```

Fig. 6: Examples of the iPower's rules.

### 3.5 Events and Actions

One of the main components of iPower is its automatic rules. A rule can be composed of time, events and actions. A rule can be event-driven or time-driven. Actions can be triggered by *simple events* or *compound events*, where the latter are combinations of multiple simple events. For example, when someone is sitting on a smart chair near a smart desk with a low light degree, to automatically turn on the lamp on the desk, we need to combine events from pressure sensors and light sensors. Note that compound events can be combined through logical operations, such as 'AND' and 'OR'.

In Fig. 5, we list the definition of iPower's rules, which are written in the format of EBNF (*Extended Backus-Naur Form*) [9] recursive grammar. Each iPower's rule defines for a certain *User*, when some *Time* and some *Conditions* are matched, the corresponding actions to be taken. Terms quoted by [ · · · ] are optional. For example, when <UserID> in a rule is not specified, it means that anyone can match this rule. Fig. 6 shows the rules for rooms A, C, and D in Fig. 1. Note that here we use *RSSI* (*received signal strength index*) between 40 and 80 to indicate that a user's badge is within the range of a WSN.

### 3.6 Protocol Stack

To implement iPower, we have designed a protocol stack in Fig. 7, which consists of the following layers:

**User layer:** The user layer defines how a user can access the system through the user interface. Here we consider two kinds of users: *administrators* and *end users*. An administrator can add or remove equipments (e.g., electric appliances, sensor nodes, and power-line control devices) in the system, change

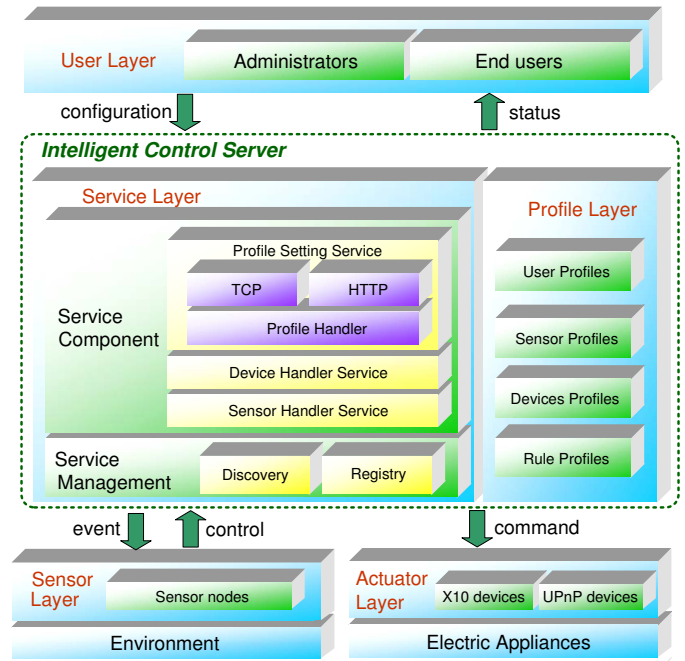


Fig. 7: Protocol stack of the iPower system.

their attributes and profiles, and manage end users. An end user can only create and modify his/her user profile.

**Service layer:** The service layer defines the rules by which the system provides and manages its services. We follow the interface defined in *OSGi* (*open service gateway initiatives*) [10], which is a service-oriented architecture for networked systems. An OSGi platform provides a standardized, component-oriented computing environment for the cooperating networked services. Using this architecture can help reduce complexity to build and maintain applications. Following OSGi, the service layer is separated into *service component* and *service management*, where the former defines the services provided by the system, while the latter provides a management mechanism to maintain these services. In our current implementation, three service components are defined, including *profile setting service*, *device controller service*, and *sensor handler service*. The profile setting service is used to create and modify a profile, while the device controller service and sensor handler service are used to control the power-line control devices and sensor nodes, respectively. To manage services, a new service component must be first *registered* to the server. The administrator can obtain the statuses of all service components in the system by the *service discovery* mechanism.

**Profile layer:** The profile layer maintains all profiles for users, sensor nodes, power-line control devices, and rules. The *sensor profiles* describe the locations and sensing types of sensor nodes. The *device profiles* describe the electric appliances controlled by the power-line control devices. Finally, the *rule profiles* define how the components in the iPower system interact with each other. All profiles are depicted by XML.

**Sensor layer:** The sensor layer controls the actions of sensor nodes. These include executing commands from the WSN gateway (such as to detect events and to generate beeping sounds) and reporting sensing data to the WSN gateway.

**Actuator layer:** This layer provides an abstraction of electric appliances to service layer and profile layer. In our implementation, we choose X10 and UPnP [11] as our device control protocols. Through these protocols, we can turn on, turn off,

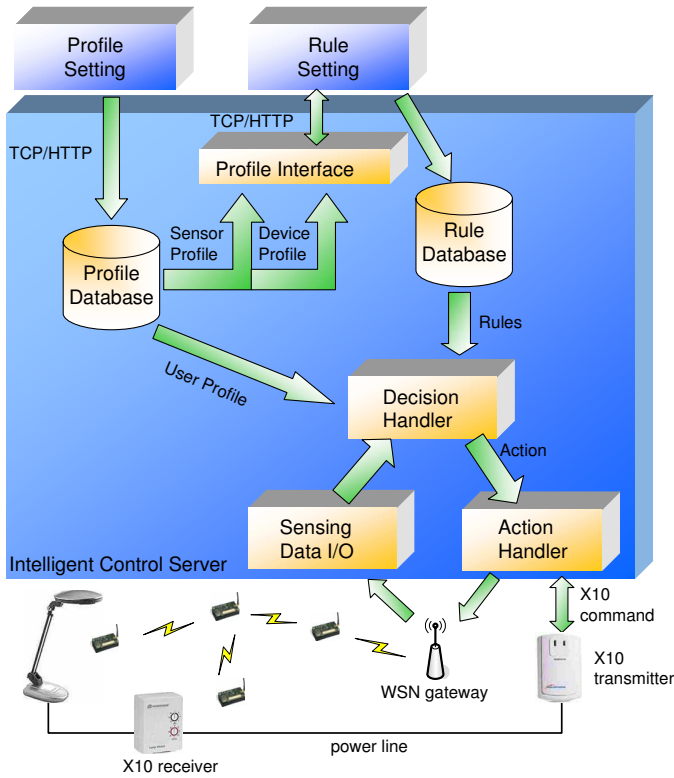


Fig. 8: Design of the intelligent control server.

and adjust the electric currents of appliances.

## 4 IMPLEMENTATION DETAILS

### 4.1 Hardware Specification

We use the MICAz Motes [12] as sensor nodes. The MICAz is a 2.4GHz, IEEE 802.15.4-compliant module that enables low-power operations and offers a data rate of 250 kbps with a DSSS radio. Each sensor node has a sensing board that can collect sensing data from their surroundings, including light, sound, and temperature. More sensors can be added on the board to increase the sensing capabilities. Each sensor node also has a buzzer to generate a beeping sound when they are commanded by a WSN gateway.

For the power-line control devices, we adopt the X10 products by SmartHome. The X10 devices consist of X10 transmitters and X10 receivers. They can communicate with each other by the X10 communication protocol, which encodes messages on the electric signal with a frequency of 60 Hz. With the X10 communication protocol, an X10 transmitter can send commands to an X10 receiver through a power line. To control electric appliances, we connect one X10 transmitter to the server via an RS-232 interface and connect all electric appliances with X10 receivers. Each X10 receiver has a unique address and at most 256 addresses can be selected.

### 4.2 Design of the Intelligent Control Server

The intelligent control server is the core of our iPower system. Fig. 8 illustrates the design of the server. The implantation details are discussed below.

- 1) An administrator can add a sensor profile or a device profile through the *profile setting* component. Related information such as sensing types and device attributes can be created in the profile database.

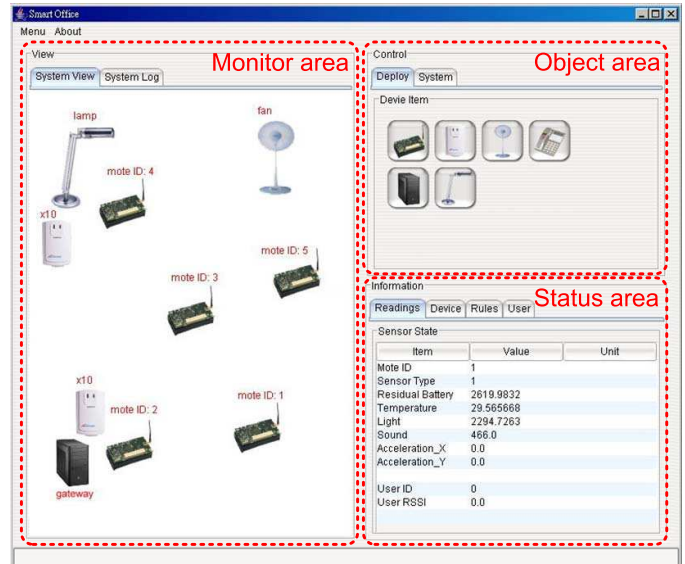


Fig. 9: The user interface at the intelligent control server.

- 2) An administrator can interact with the *profile interface* to create rules through the *rule setting* component.
- 3) A gateway can report environment information through the *sensing data I/O* interface.
- 4) The *decision handler* combines the user profiles, rules, and sensing data to generate proper actions.

The actions are sent to the *action handler*, which can generate commands to X10 devices or sensor nodes.

### 4.3 User Interface

We provide a user interface to manage the system and allow users to create their profiles at the server, as shown in Fig. 9. The user interface has an *object area*, a *monitor area*, and a *status area*. The object area provides an interface to deploy all devices in a room, including WSN gateways, sensor nodes, electric appliances, and X10 devices. This area also allows users to start or stop the system. In the monitor area, the administrator can visualize the deployment of sensor nodes and devices. He/She can add new objects in the room by dragging objects from the object area to the monitor area. The monitor area also shows the network topology and electric appliance in the room. In the status area, the administrator can observe the attributes and the current status of each sensor node.

## 5 SYSTEM EVALUATION

In this section, we present some simulation results to evaluate the system performance. We consider the energy consumption of an office with five people, one air conditioner, and five desk lamps, where each lamp is owned by one person. Table 1 lists the energy consumptions of different electric appliances. For the air conditioner, we assume that it will spend extra 100 watts when the temperature is decreased by 1 °C. A *two-state discrete Markov model* [13] is used to model a person's behavior during every hour, as shown in Fig. 10(a). A person can be either in one of the two states: *leave* or *stay*. When a person is in a *leave* state, the corresponding desktop lamp will be turned off. We use another Markov model to model the detailed behavior of a person when he/she in a *stay* state, as shown in Fig. 10(b). In particular, during every twenty minutes, the person may

| electric appliance | energy consumption        |
|--------------------|---------------------------|
| air conditioner    | 800 watts/hour (at 28 °C) |
| desk lamp          | 80 watts/hour             |

TABLE 1: Energy consumptions of electric appliances.

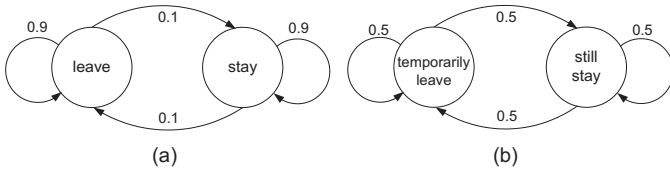


Fig. 10: Two-state discrete Markov models. (a) Model a person's behavior in each hour. (b) Model a person's detailed behavior in each twenty minutes.

decide whether to 'still stay' in the office or 'temporarily leave' the office. When the person decides to temporarily leave the office, his/her own desktop lamp will remain on if the iPower system is not applied. Table 2 lists the favorite temperatures of the five people. When there are two or more people in the office, the temperature of the air conditioner will be adjusted to the average of favorite temperatures of those people in the office. Note that without iPower, we only adjust the temperature of the air conditioner when people enter the office.

Fig. 11 shows the energy consumption with five people during ten hours. We can observe that without iPower, the energy consumption of the office is always higher than 1500 watts, even when there is no person in the office (i.e., hours 3, 7, 9, and 10). This is because when all people temporarily leave the office, the air conditioner and some desk lamps are still turned on. On the other hand, the iPower system can detect such situation and thus properly turn off some electric appliances to conserve energy. Fig. 12 compares the total energy consumptions of the office with different numbers of people in the simulation. As can be seen, our iPower system can save approximate 16.5% ~ 46.9% energy, which reflects its effectiveness.

## 6 CONCLUSIONS AND DISCUSSIONS

In this work, iPower has been proposed to conserve energy in an intelligent building and to provide personalized services for environment control. The iPower system can detect if there is possible waste of electricity by WSNs and then turn off these unnecessary electric appliances via the X10 power-line control devices with a user-friendly design. The iPower system also provides personalized services in which electric appliances can be automatically adjusted to satisfy users' requirements. We have presented the design and implementation details of iPower. Prototyping experiences and design issues are also given in this paper.

The prototyped iPower system can be further improved in several ways. First, since the X10 protocol is somewhat slow

| person | favorite temperature |
|--------|----------------------|
| A      | 25 °C                |
| B      | 27 °C                |
| C      | 26 °C                |
| D      | 22 °C                |
| E      | 24 °C                |

TABLE 2: Favorite temperatures of the five people.

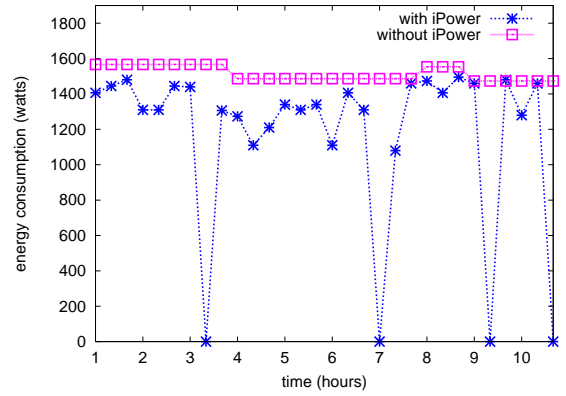


Fig. 11: Energy consumptions during 10 hours.

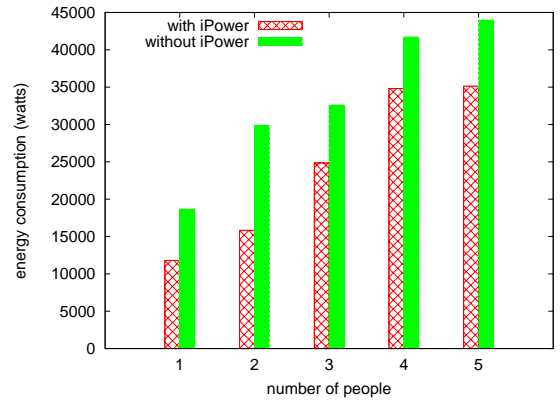


Fig. 12: Total energy consumptions with different numbers of people.

and sometimes unreliable, in the future we plan to replace X10 by *INSTEON* [14], which could be more reliable and could transmit at a higher speed. Also, we are considering integrating other intelligent furniture into our system. Below, we point out several important design issues that deserve attention.

**Conflicting profiles:** When two or more people are in the same room, their profiles may conflict with each other since each person may have different requirement or preference in temperature and light. To solve the profile-conflicting problem, we propose to assign a weight to each user and adopt the weighted average to determine the desired degrees of temperature and light. For example, suppose that two users have favorite temperatures of 23 °C and 26 °C in their profiles, and their weights are 3 and 2, respectively. Then the desired temperature will be  $\frac{3}{3+2} \times 23 + \frac{2}{3+2} \times 26 = 24.2$  °C. Note that the weight assignments can depend on the application requirements or user priorities.

**Privacy and security:** In the iPower system, the complete user profiles are stored in the control server. A user identification device only needs to transmit its ID to the control server to find out the corresponding profile. Thus, the personal information will not be exposed through the user identification device. The ID of a user can be represented either by the address or the network interface card or a higher-level identity. Since the network address must be in clear text in any communication, it is insecure to use such addresses as user IDs. So, the latter approach is preferred (which can be protected by any encryption algorithm).

**Message reliability:** Most of the signalling messages in

Fig. 3 require an acknowledgement mechanism to guarantee their delivery. Unfortunately, the X10 devices do not support such acknowledgement mechanism. To solve this problem, we can enforce sensor nodes to report their current environmental statuses to check whether the X10 devices have successfully deliver the commands from the control server. For example, in Fig. 3, suppose that sensor nodes report that there is nobody in the room and thus the control server will send a command to the X10 receiver to turn off the electric appliance (e.g., the desk lamp). If the turn-off command is lost due to channel errors in the power-line, the sensor node can maintain a timer to check whether the command from the control server has been reflected from its reading related to the desk lamp. Therefore, the message loss problem on X10 can be resolved.

**Incorrect sensing readings:** Due to environmental noises or errors, the readings of sensor nodes may not be accurate. This may mislead the control server to make incorrect decisions. To solve this problem, we can apply the solutions in [15]–[17] to alleviate the effects of these inaccurate sensing readings.

**Environmental factors:** Some environmental factors like sunlight can be considered to help conserve more energy. For example, the work in [18] suggests adjusting lamps according to users' requirements and the sunlight. Similarly, we can apply this extension in our system.

## REFERENCES

- [1] O. Gassmann and H. Meixner, *Sensors in Intelligent Buildings: Sensors Applications*, Wiley-VCH, 2001.
- [2] H. Schulzrinne, X. Wu, S. Sidiroglou, and S. Berger, "Ubiquitous computing in home networks," *IEEE Comm. Magazine*, vol. 41, no. 11, pp. 128–135, 2003.
- [3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R.S.M. Handley, and E. Schooler, "SIP: session initiation protocol," IETF RFC 3261, June, 2002.
- [4] S.K. Das, D.J. Cook, A. Battacharya, E.O. Heierman III, and T.Y. Lin, "The role of prediction algorithms in the mavhome smart home architecture," *IEEE Wireless Comm.*, vol. 9, no. 6, pp. 77–84, 2002.
- [5] X. Wang, J.S. Dong, C.Y. Chin, S.R. Hettiarachchi, and D. Zhang, "Semantic space: an infrastructure for smart spaces," *IEEE Pervasive Computing*, vol. 3, no. 3, pp. 32–39, 2004.
- [6] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, "The gator tech smart house: a programmable pervasive space," *IEEE Computer*, vol. 38, no. 3, pp. 50–60, 2005.
- [7] SmartHome, "X10-1132B," <http://www.smarthome.com/>.
- [8] WorldWideWeb Consortium (W3C), "Extensible markup language (XML) 1.0," <http://www.w3.org/TR/REC-xml/>.
- [9] R.W. Sebesta, *Concepts of programming languages (fourth edition)*, Addison-Wesley, 1999.
- [10] O. Alliance, "OSGi – The dynamic module system for Java," <http://www.osgi.org/>.
- [11] UPnP, "UPnP forum," <http://www.upnp.org/>.
- [12] Crossbow, "MOTE-KIT2400," <http://www.xbow.com/>.
- [13] L. Kleinrock, *Queueing systems, volume 1: theory*, John Wiley & Sons Inc., 1975.
- [14] INSTEON, "INSTEON, a smart labs technology," <http://www.insteon.net/>.
- [15] J. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," *Proc. IEEE Int'l Conf. Distributed Computing Systems*, 2006.
- [16] Y. Zhuang, L. Chen, X.S. Wang, and J. Lian, "A weighted moving average-based approach for cleaning sensor data," *Proc. IEEE Int'l Conf. Distributed Computing Systems*, 2007.
- [17] B. Sheng, Q. Li, W. Mao, and W. Jin, "Outlier detection in sensor networks," *Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing*, pp. 219–228, 2007.
- [18] V. Singhvi, A. Krause, C. Guestrin, J.H. Garrett, and H.S. Matthews, "Intelligent light control using sensor networks," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems*, pp. 218–229, 2005.