行政院國家科學委員會/經濟部能源局
「能源科技學術合作研究計畫」成果報告

無線感測網路應用於能源節約之個人化適性服務

計畫類別：■ 個別型計畫　　□ 整合型計畫

計畫編號：NSC 96-2623-7-009-002-ET

執行期間：96 年 1 月 1 日至 96 年 12 月 31 日

計畫主持人：曾煜棋

計畫參與人員：潘孟鉉、葉倫武、梁家銘、吳鈞豪、胡淑琮、陳志誠、盧奕丞

成果報告類型(依經費核定清單規定繳交)：□ 精簡報告
■ 完整報告

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，
□一年□二年後可公開查詢

執行單位：國立交通大學資訊工程系

中　華　民　國　96　年　12　月　17　日

摘要

無線感測器網路提供了一種非常方便的形式來監控環境中之資訊，故在近幾年來，無線感測網路已被大量運用在各項應用方面。本計畫使用無線感測器網路來監控室內之環境資訊，並結合控制系統至室內家電，來達到省電之效益。

本系統以節約能源為目標，並且考量到不同使用者均有不同偏好，且在從事不同行為時，所需之環境因素也不儘相同，且根據使用者事前設定好之偏好值，並加上無線感測器所收集之即時環境資訊，再透過一套可適性之決策演算法，計算出最適於目前環境之設定值，且透過幾項家電控制協定，來將室內之電器自動的調整至最適於使用者之狀態，以滿足所有使用者之偏好值。

關鍵字：節能、無所不在的運算、智慧型生活環境、無線通訊、無線感測網路

Abstract

Wireless sensor networks (WSNs) provide a convenient way to monitor the physical environment. In recent years, WSNs have been widely discussed in many applications. We combine wireless sensor networks and control systems to achieve energy saving.

Different users have different preference. When user does different activities, he needs different environmental factors. The goal of our system is energy saving. According the environmental information which is collected by wireless sensor networks, our system would do an adaptive decision algorithm. According the result of decision algorithm, our system would adjust the appliance automatically to achieve energy saving.

Keywords：energy conservation, pervasive computing, smart environment, wireless communication, wireless sensor network

一、前言

電力的發現是人類史上非常重要的里程碑，它不但推動了後續許多的發明，更豐富了我們的生活，直到今日，我們的生活沒有一天不能不使用到電力。雖然電力便利了我們的生活，但近幾年來，電力的過渡使用，造成了電力的供不應求，舉例來說，在炎炎夏日裡，電力往往因大家過度的使用，常超過了可負荷的極限，而發生了跳電的情形，過度用電不但容易造成用電緊張，更可能為了製造這些電力而造成環境的汙染。因此，為了響應日趨重視的環境意識,『節約用電』可說是非常重要的議題。

二、研究目的

在日常生活之中，電力消耗最多的莫過於暖氣機(Heating)、通風設備(Ventilating)與冷氣機(Air Conditioning)，除此之外，還有電燈也佔了不少耗電之比重，這些電器統稱為 HVAC 系統，根據資料顯示，HVAC 系統消耗了三分之一以上之用電量，倘若能夠節省這些電器之耗電量，勢必可以節省十分可觀之能源。

根據統計資料顯示，最常見之電源浪費情形，為使用者之疏忽，舉例來說，在一辦公室內，當人員離開時，常常忘了關閉其座位之檯燈，仰或是當辦公室人員全離去時，忘了關閉辦公室內的冷氣，此類的疏忽，所消耗之能源，一年累積下來，所以浪費之電力相當可觀，倘若能夠有一套良好的機制，能夠節省此類的能源，不儘能為地球之環保儘一份心力，也能夠節省公司電力之開銷。

除了使用者之疏忽以外，在室內之中，冷氣之溫度之高低與日光燈之強弱，也是影響用電量很重要的一環，舉例來說，室內人員偏好

之空調溫度並非非常低時，空調可不必開太冷，也可搭配電扇來使用，冷氣調高一度可節省之能源即相當可觀了，除了溫度之外，每位人員偏好光線強度也不儘相同，所提供之光度越高，所花費之電量亦越高，若能將室內之光線開啟在適當之光度之下，且能夠滿足所有使用者之偏好光度，也可節省不少用電。

為了達到上述的研究目的，我們使用無線感測器網路，利用無線感測器之特性，來收集環境中之資訊，並整合至室內之家電之中，來達到家電可調適性之自動化，並在符合每位使用者需求之下，做一項能源最有效之運用。

三、文獻探討

近幾年來，無線感測網路被廣泛的運用在各項領域之上，最貼近於人們日常生活之應用，莫過於應用於一般家電之上，主要為省電與滿足使用者偏好為主要之議題。如何根據使用者在室內之行為與位置，來決定家電之反應，國內外已有不少此類之相關研究，舉例來說，在[1]之中，作者提出了一套系統，藉由情境感知來建構出智慧型之生活環境，其主要是利用感測器來感測環境之資訊，來做為家電控制之依據，以達到情境感知之目標。而有些研究，則是著重在分析使用者之歷史記錄，在[2]之中，其記錄使用者多筆歷史記錄，藉由這些歷史記錄，可推測使用者未來之行為與位置等等，讓此環境能夠自動化的來因應使用者之需求。而有些則是提出了一套軟體架構，在[3]之中，其提出了一套一般性之軟體架構，來達到家電控制的功能。

上述的種種研究，主要為家電控制與家電自動化，而有些研究，則是集中在單一家電之上，舉例來說，在[4][5][6][7]裡，主要探討的

均為燈光之控制，如何調節室內之燈源，以滿足使用者需求，仰或是能達到省電之功能，在[5]之中，其定義了多種不同之使用者需求與其相關之代價函數，而其目標主要為最小化此代價函數，來滿足使用之需求。而在[6]之中，則是將燈光控制問題描述為一能源消耗與滿足使用者偏好光度之損益問題，其對每一位使用者偏好之光度值，均個別定義一項效益函數，而其目標為最大化所有使用者之效益函數，讓多數使用者均能夠有其偏好之光度值。而[4]和[7]則是著重在省電之議題之上，利用無線感測網路之優勢，來節省大樓內之各項能源消耗。

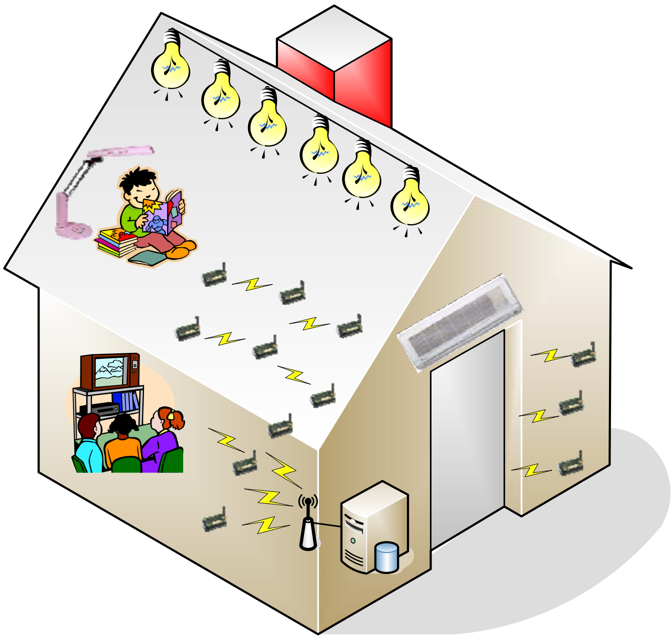
四、研究方法

本系統以省電為目標，來自動的關閉室內不必要開啟之電器，並能夠滿使用者之偏好，而在滿足使用者偏好方面，主要是集中在光度，來做為實驗與實做之項目，但本系統為一套一般性之系統，可直接套用於其他同性質之環境因素，如溫度、聲音等等，讓使用者能夠滿足其生活空間之的各項環境因素。

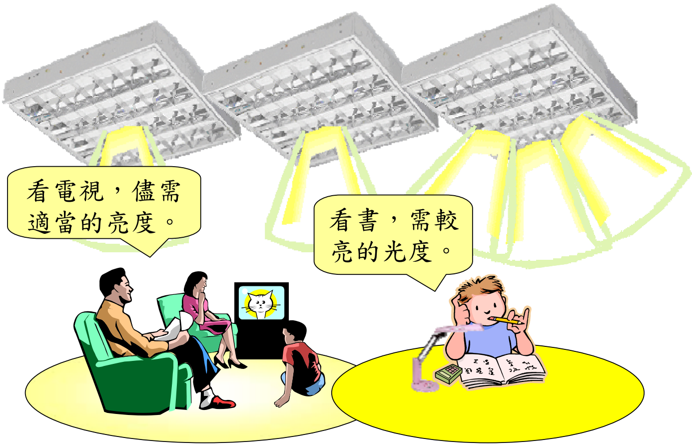
1、 系統情境

在環境中的使用者，其身上會備帶有一識別裝置，此裝置會發送無線訊號，讓本系統可得知此使用者之位置，以其辨別使用者身份，且使用者可透過此裝置，讓本系統得知其目前所從事之行為，根據不同行為，本系統可自動的來調節其需之電器，及其所需之光源強弱，用以節省能源。圖一為系統情境示意圖，室內有冷氣、檯燈以及天花板之日光燈，而本系統會根據使用者之行為，來自動之開啟或關閉電器，以及調節燈光亮度，以滿足室內所有使用者，而圖二為光度偏好

情境示意圖，不同之行為，所需之光度均不相同，舉例來說，看電視儘需適當之亮度，但看書時，則需要較強之光度，讓使用者能夠在其作業面上專心閱讀，而本系統會依據感測器數值以及使用者行為與喜好，來自動調整光源強弱，用以節省能源。



圖一：系統情境示意圖

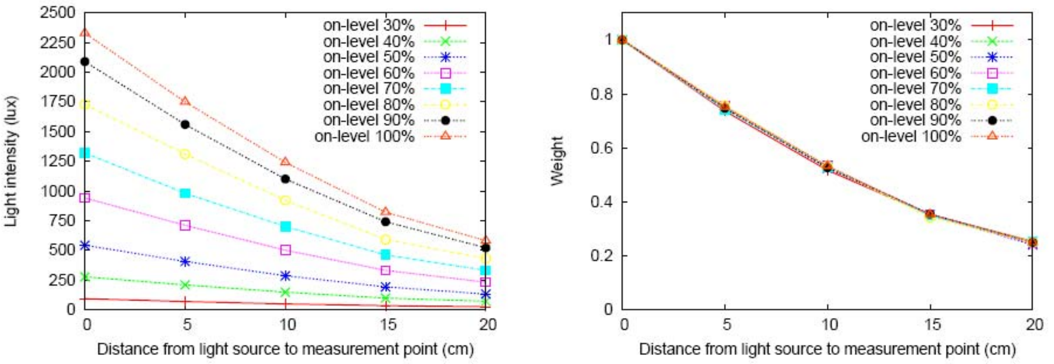


圖二：光度偏好情境示意圖

2、　光度特性

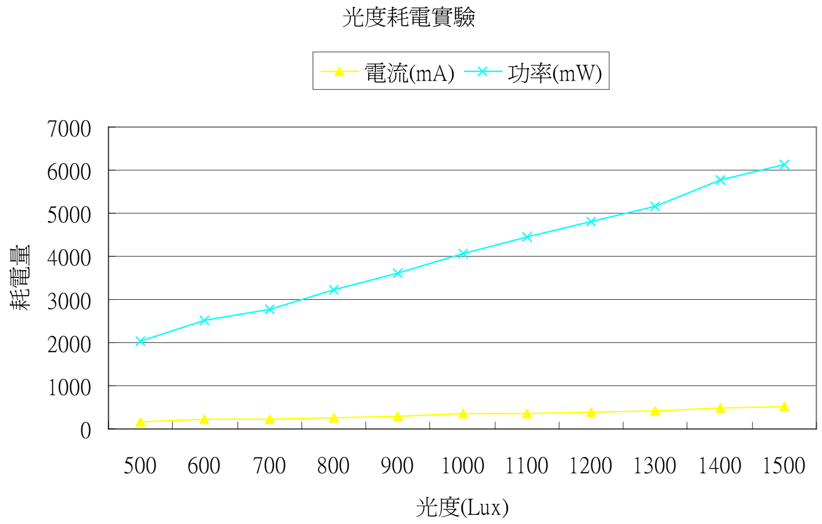
為了滿足每位使用者偏好的光度值，首先要了解不同光度在不同距離下所呈現之關系，以及不同光度，所消耗之能源為多少，故我們做了底下幾項實驗，以輔助本系統模型之設計與實作。

第一項實驗為距離與亮度之實驗，此實驗使用一盞燈源，並每次固定其亮度，但在不同距離之下(0、5、10、15 和 20cm)測量其光度值(lux)，並做多次實驗，每次調整不同之光度值，其結果為圖三(a)所示，在開啟同樣光度之條件之下，距離越遠，則感測到之光度越低，若以光源距離為 0 cm 為基準值(在此訂為 1)，而其他距離依此做標準化，則可得到圖三(b)，由圖中發現，所有亮度值之實測結果，都重合於同一線段上，由此可得知，若系統事先量測每定點每盞光源所以提供之最強光度，則之後可由此關系來推得該光源所開啟之強度。



圖三：亮度衰減實驗

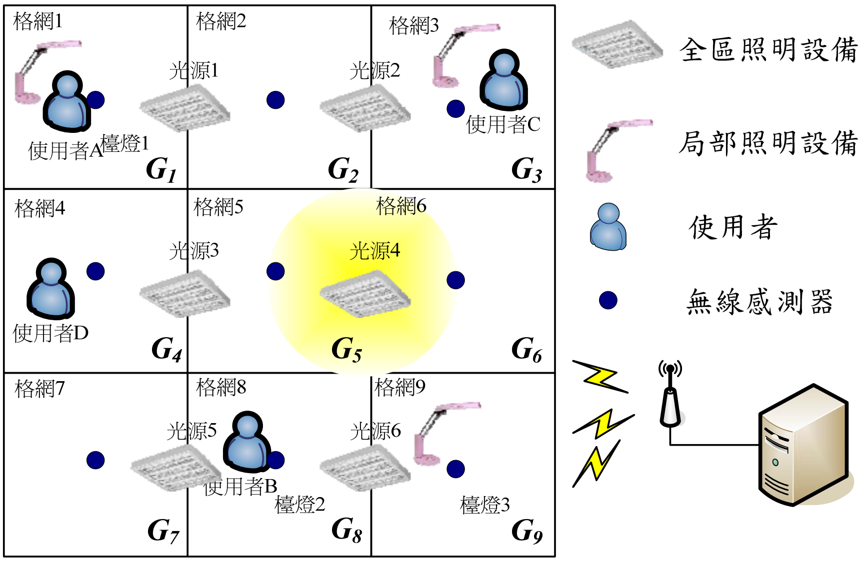
第二項實驗為亮度與耗電之實驗，此實驗為開啟一盞檯燈，固定電壓為 12V，每次將檯燈之光度增加 100 Lux，並記錄其電流值(mA)，和其對應之耗電功率，而由圖四可得知，光度越強，則耗電功率越大，故若能控制適當的亮度，則勢必能節省可觀之耗電量。



圖四：光度耗電之實驗

3、 系統模型

本系統是應用於室內之電器，舉例來說，如冷氣，其所影響之範圍為整間房間，故系統模型並不定義其有效範圍，但如光度，如圖三之實驗結果所示，其影響範圍與距離有密切之關係，故本系統將環境切割為多個格網，且定義每盞光源可影響之格網。

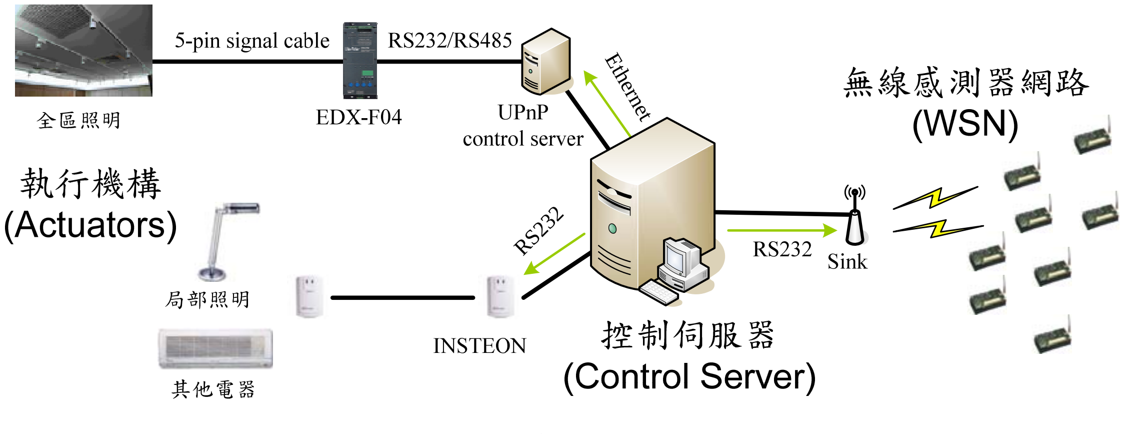


圖五：網路情境示意圖

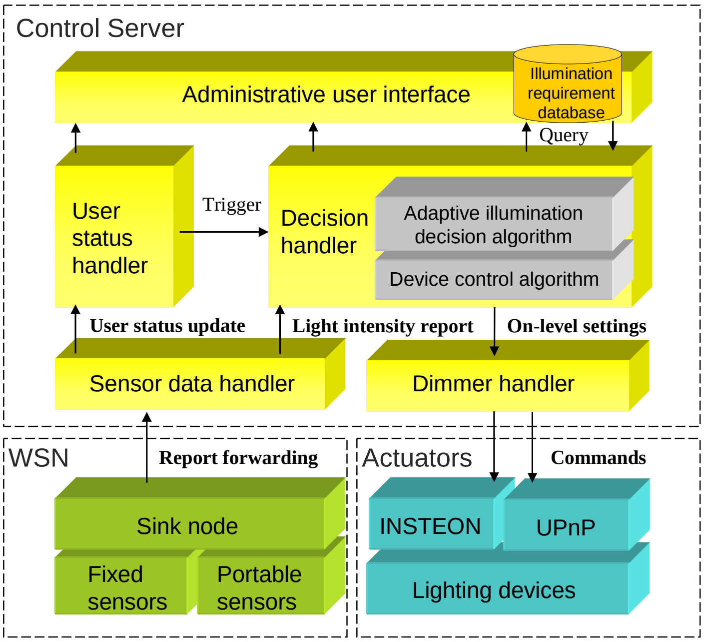
在本系統中，一般照明設備之照明可能會影響多個網格區域，例如圖五之光源4會影響網格五與格網六內之固定型感測器所感測之數值。由於光線也是隨著距離而衰減，因此我們假設在系統建置階段我們可以得知一般照明設備對網格內感測器亮度值之影響程度。例如：某一般照明設備A對網格G_j的亮度影響程度為0.7，則代表在網格G_j中感測器所感測之亮度值為一般照明設備所發出之亮度值的0.7倍。根據圖三顯示之亮度衰減實驗，由這實驗可知，無論燈光目前之調控值為何，光線衰減程度都可為一線性關係。在本系統中，我們將一般照明裝置對網格之影響程度記錄於一矩陣W中。我們可以利用W矩陣來求得目前一般照明設備所提供之亮度值，其計算方法如下：假設我們已知每個一般照明裝置所處之位置，我們利用所處位置之資訊由取出m條相對應之row，建構一矩陣W'，並取出一般照明裝置所處位置之感測器亮度值記錄於一列矩陣S_f，假定一般照明設備目前提供之亮度值為L_D，則S_f、W'及L_D有這下列關係：$S_f=W'L_D$。L_D可使用一簡單地以解聯立方程式求得。

4、 系統架構

本系統主要可分為三大部分，無線感測網路、控制伺服器以及執行機構。底下分述各項之細節。



圖六：系統架構圖



圖七：系統實作堆疊圖

- 無線感測網路

此部分我們使用了 Jennic[8]做為無線傳輸之節點，Jennic 為符合 IEEE 802.15.4[9]協定之無線感測模組，其運作在 2.4 GHz 頻帶之上，而在感測器之部分，我們自製了一塊感測板，其使用 Si photodiode[10]做為光度感測器，並在此感測板上設置數個按鈕，使用者可回報其目前之行為至本系統，而使用者則隨身攜帶此感測器，用以感測其作業面之照度，並可回報其行為。而在環境之中，我們佈建了許多無線感測器，這些感測器會定期回報環境中的資訊，如光度、溫度等等，並且藉由這些感測器之訊號強弱，本系統可定位出使用者目前所在之位置。環境中之無線感測器均將資料回傳到無線基地台(sink)，而此無線基地台以 RS232 連接至控制伺服器。

- 控制伺服器

此部分為系統之核心所在，主要採用 Java 程式語言來撰寫，可分為五大部分，即 user status handler、decision handler、sensor data handler、dimmer handler 以及 administrative user interface，底下分述各部分之細節。

i. Sensor data hander：此部分會定期的接收無線感測器網路所傳回之環境資訊，如光度、溫度等，以及使用者所回傳目前從事之行為。

ii. User status handler：根據 sensor data handler 所回傳之資訊，可計算出使用者目前所之位置，並予以記錄，當使用者位置或其行為有改變時，則通知 decision handler，以重新計算出最適於目前狀況之節能策略。

iii. Decision handler：此部分接收了由 sensor data handler 傳回之環境資訊，以及 user status handler 所維護之使用者資訊，再加上使用者事前定義好之環境偏好，如偏好之溫度或亮度等，綜合以上資訊來執行一項可適性調節演算法，來決定出目前環境中之電器應如何開啟或調整，並將此決策傳送至 dimmer handler。

iv. Dimmer handler：此部分會根據 decision handler 所計算出之結果，發送一對應之指令至室內之電器，在目前實作之版本中，有 INSTEON 協定與 UPnP 協定。

v. Administrative user interface：除了底層複雜的系統之外，我們還實作了一套完整的介面，讓管理者可以透過此介面來維護和監控整套系統，舉例來說，可藉由此介面來設定使用者偏好之光度值，或藉由此介面來觀看系統中所有感

測器之讀數與狀態等。

- 執行機構

在目前我們實作的項目裡，執行機構分成兩部分，一般電器和局部照明設備使用 INSTEON[11]來做為控制之協定，我們使用 INSTEON PowerLinc 控制器和 INSTEON LampLinc 調光器來控制檯燈，其中 INSTEON PowerLinc 以 RS232 連接至控制伺服器，而控制伺服器從 RS232 發送 INSTEON 指令至 INSTEON PowerLinc 後，此指令透過電力線傳送至 INSTEON LampLinc，而 INSTEON LampLinc 則根據指令內容，來關閉、開啟或調整電流大小，以控制相對應之電器。

而在全區照明設備方面，則是使用 UPnP[12]來做為控制之協定，我們實作了 UPnP Lighting Controls v1.0 之標準來控制室內之全區照明設備，控制伺服器發送 UPnP 指令至 UPnP 控制伺服器，而此 UPnP 控制伺服器則連接至 EDX-F04[13]調光器，此調光器可根據指令來調整全區照明之強弱，其調控刻度為 0 至 100。

五、結果與討論

在本節模擬之中，以一小型辦公室為情境，並假設有五位使用者、一台冷氣機與五盞檯燈，其中每盞檯燈均位於每位使用者之辦公桌上，表格一表示了不同電器所耗費之能源，而對於冷氣機，在此假設每降攝氏一度，則需要多增加 100 瓦特的用電。

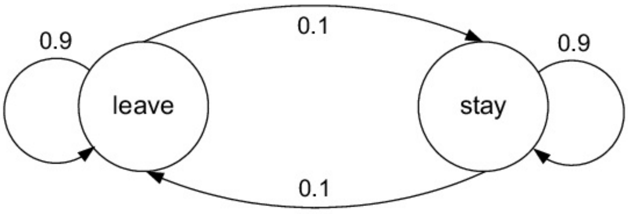
表格一：電器耗電量

電器	耗電量
冷氣	800 瓦特/小時 (28˚C)
檯燈	80 瓦特/小時

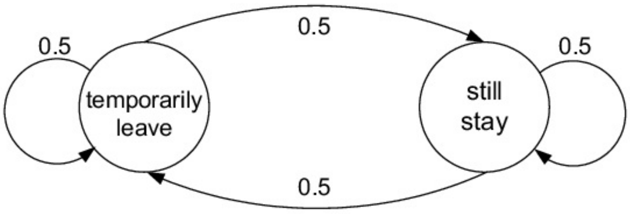
表格二：使用者偏好溫度

使用者	偏好溫度值
A	25˚C
B	27˚C
C	26˚C
D	22˚C
E	24˚C

對於辦公室內使用者之行為，我們以二狀態離散馬可夫模型(two-state discrete Markov)來模擬其每小時之行為，如圖八(a)所示。每位使用者有兩種狀態 leave 或 stay，當使用者狀態為 leave 時，其檯燈之狀態為關閉。而當使用者狀態為 stay 時，我們以另一個二狀態馬可夫模型來描述其每二十分鐘之行為，如圖八(b)所示，其狀態可為 still stay 或是 temporarily leave，當使用者之狀態為 temporarily leave 時，若不使用本系統，則其辦公桌上之檯燈為開啟，若使用本系統則為關閉。對於每位使用者，其均有不同之偏好溫度，如表格二所示，當室內有兩位以上之使用者，則冷氣開啟之溫度為使用者偏好溫度之平均值，當不使用本系統時，冷氣只有在使用者進入辦公室才會更動，當有使用者離開時並不會調整冷氣溫度。



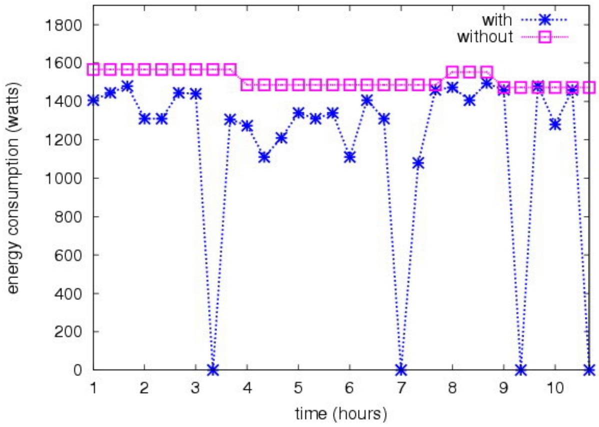
(a) model a person’s behavior during each hour



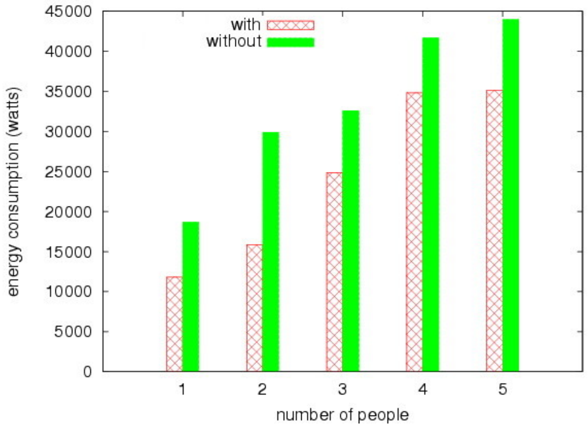
(b) model a person’s detailed behavior during each twenty minutes

圖八：使用者之二狀態離散馬可夫模型

圖九為十小時內之能源消耗，從圖九可觀察到，當不使用本系統時能源消耗均大於使用本系統，因為當使用者暫時離開辦公桌時，其檯燈之狀態仍然為開啟，而若是使用本系統，則本系統會自動偵測到無人在該辦公桌，則自動關閉該盞檯燈，用以節省不必要之能源浪費。圖十為不同人數之總能源消耗，由圖中可觀察到，使用本系統可節省約 16.5%~46.9%之能源。



圖九：十小時內之能源消耗



圖十：不同人數之總能源消耗

本系統將無線感測網路結合於一般日常生活中，以節約能源為目標，並且考量到環境中使用者之行為與其偏好，如光度、溫度等，在滿足室內使用者之需求之下，來達到省電的需求。並以模擬來驗證使用本系統確實可節省大量電源，且實做出系統之雛型。

六、計畫成果自評

- 熟悉並了解無線感測傳輸節點(Jennic)的各項功能，並可將收到的資訊加以處理，且回報至控制伺服器。
- 使用 Si photodiode 做為光度感測器，並實際設計出此電路板，且將感測到之資料傳送至無線感測傳輸節點。
- 設計一套無線通訊協定，來將無線感測器節點所偵測到的環境資料傳回至 sink，並回報至控制伺服器。
- 設計了一套省電之演算法，可根據使用者之行為，及其位置，來

決策出目前室內應開啟那些電器，及其強弱。

- 實作出 UPnP 燈光控制標準，用以整合至本系統，來控制全區照明設備。
- 結合 INSTEON 協定，用以控制一般家電其局部照明設備。
- 撰寫模擬，用以驗證所設計之演算法確實可節省非常可觀之用電量。
- 整合上述所有元件，並將其整套系統完整實作。

研究成果

[1] L.-W. Yeh, Y.-C. Wang, and Y.-C. Tseng, “iPower: An Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks”, Int'l J. of Sensor Networks, to appear

[2] Y.-C. Wang and Y.-C. Tseng, "Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multi-level Coverage", IEEE Trans. on Parallel and Distributed Systems, to appear (SCI, EI).

參考文獻

[1] S. Helal,W. M. H. E. 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.

[2] S. K. Das, D. J. Cook, A. Battacharya, E. O. H. III, and T. Y. Lin, “The role of prediction algorithms in the mavhome smart home architecture,” IEEE Wireless Communications, vol. 9, no. 6, pp. 77–84, 2002.

[3] H. Schulzrinne, X. Wu, S. Sidiroglou, and S. Berger, “Ubiquitous computing in home networks,” IEEE Communications Magazine, vol. 41, no. 11, pp. 128–135, 2003.

[4] F. O’Reilly and J. Buckley. Use of wireless sensor networks for fluorescent lighting control with daylight substitution. In Proc. of Workshop on Real-World Wireless Sensor Networks (REANWSN), 2005.

[5] H. Park, M. B. Srivastava, and J. Burke. Design and implementation of a wireless sensor network for intelligent light control. In Proc. of Int’l Symposium on Information Processing in Sensor Networks (IPSN), 2007.

[6] V. Singhvi, A. Krause, C. Guestrin, J. H. Garrett, and H. S. Matthews. Intelligent light control using sensor networks. In Proc. of ACM Int’l Conference on Embedded Networked Sensor Systems (SenSys), 2005.

[7] Y.-J.Wen, J. Granderson, and A. M. Agogino. Towards embedded wireless-networked intelligent daylighting systems for commercial buildings. In Proc. of IEEE Int’l Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2006.

[8] Jennic, JN5121 IEEE 802.15.4 Evaluation Kit. http://www.jennic.com/.

[9] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks specific requirements part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs), 2003.

[10] S1133 si photodiode. http://cn.100y.com.tw/PNoInfo/10155.htm.

[11] INSTEON, Lamplinc and Powerlinc controller. http://www.insteon.net/.

[12] UPnP Forum. http://www.upnp.org/.

[13] EDX-F04. http://www.liteputer.com.cn/china/liteputer-tw/product.asp.

附件

- 附件一：L.-W. Yeh, Y.-C. Wang, and Y.-C. Tseng, “iPower: An Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks”, Int'l J. of Sensor Networks, to appear
- 附件二：Y.-C. Wang and Y.-C. Tseng, "Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multi-level Coverage", IEEE Trans. on Parallel and Distributed Systems, to appear (SCI, EI).

附件一

L.-W. Yeh, Y.-C. Wang, and Y.-C. Tseng, “iPower: An Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks”, Int'l J. of Sensor Networks, to appear

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

Lun-Wu Yeh
Department of Computer Science,
National Chiao-Tung University, Hsin-Chu, 30010, Taiwan
E-mail: lwyeh@cs.nctu.edu.tw

You-Chiun Wang
Department of Computer Science,
National Chiao-Tung University, Hsin-Chu, 30010, Taiwan
E-mail: wangyc@cs.nctu.edu.tw

Yu-Chee Tseng*
Department of Computer Science,
National Chiao-Tung University, Hsin-Chu, 30010, Taiwan
E-mail: yctseng@cs.nctu.edu.tw
*Corresponding author

Abstract: Wireless sensor networks (WSNs) provide a convenient way to monitor the physical environment. Exploiting the context-aware capability of WSN to achieve energy conservation in intelligent buildings is an attractive direction. We thus propose an *iPower (intelligent and personalized energy-conservation system by wireless sensor networks)* system which 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 for energy conservation. The iPower system is quite 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.

Keywords: context awareness; energy conservation; sensor network; smart environment; wireless communication.

Reference to this paper should be made as follows: Yeh, L.W., Wang, Y.C. and Tseng, Y.C. (2007) 'iPower: An Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks', Int. J. Sensor Networks, Vol. 1, Nos. 1/2/3, pp.XX–XX.

Biographical notes: Lun-Wu Yeh received his M.S. degrees in Computer and Information Science from the National Chiao-Tung University, Taiwan, in 2005. He is currently pursuing Ph.D. in Computer Science from the Chiao-Tung University, Taiwan. His research interest is wireless sensor network.

You-Chiun Wang received his Ph.D. in Computer Science from the National Chiao-Tung University, Taiwan, in 2006. He is currently a postdoctoral research associate at the Department of Computer Science, National Chiao-Tung University, Taiwan. His research interests include wireless communication, mobile computing, and sensor network.

Yu-Chee Tseng received his Ph.D. in Computer and Information Science from the Ohio State University in 1994. He was an Associate Professor at the Chung-Hua University (1994∼1996) and at the National Central University (1996∼1999), and a Professor at the National Central University (1999∼2000). Since 2000, he has been a Professor at the Department of Computer Science, National Chiao-Tung University, Taiwan, where he is currently the Chairman. His research interests include wireless communication, mobile computing, and parallel and distributed computing.

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 Gassmann et al. (2001), 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 therefore 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 the iPower system, WSNs are deployed in rooms of a building to collect information of the environment. Such information is reported to a control server to determine whether to turn off those unnecessary electric appliances in the building. 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 the iPower system, we also provide *personalized services* in which 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 works. 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 Works

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 Schulzrinne et al. (2003) considers a ubiquitous computing architecture in which electric appliances are controlled by a SIP (session initiation protocol) (Rosenberg et al., 2002) server, under which architecture users can make calls to communicate with the SIP server to control their electric appliances. In the *MavHome* system (Das et al., 2002), 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* (Wang et al., 2004), 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 Helal et al. (2005) 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.

3 Design of The iPower System

3.1 System Architecture

The architecture of our iPower system is illustrated in Figure 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 current prototype, three types of sensing data can be collected, including 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 of events is reactive, in the sense that a node will report 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

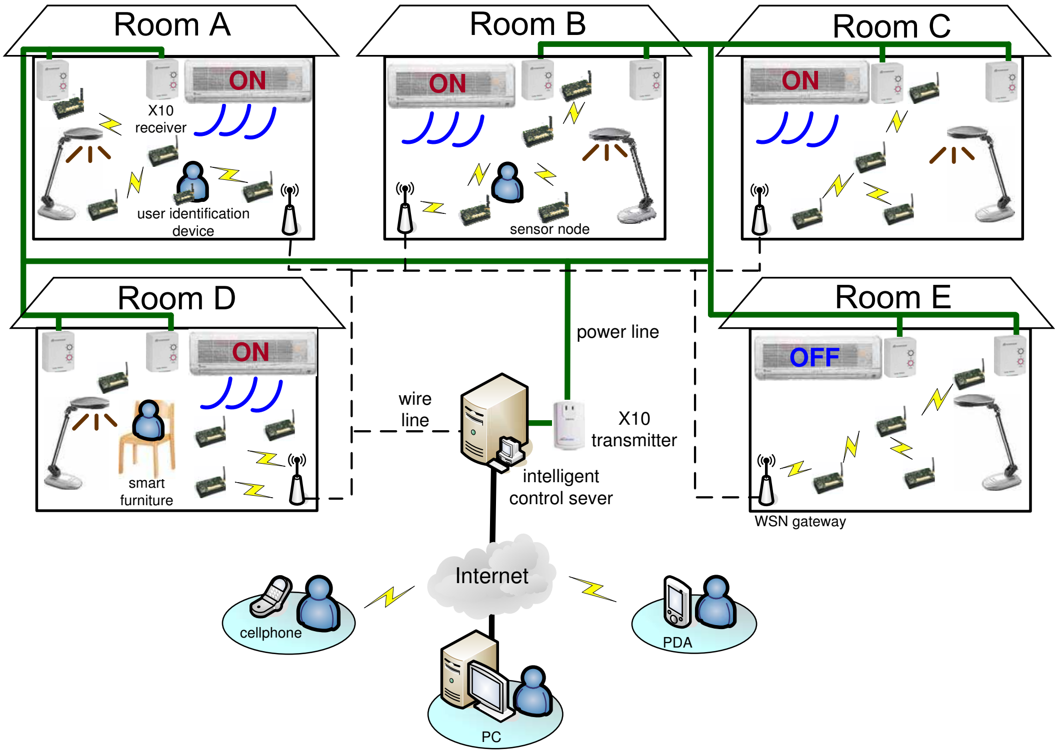


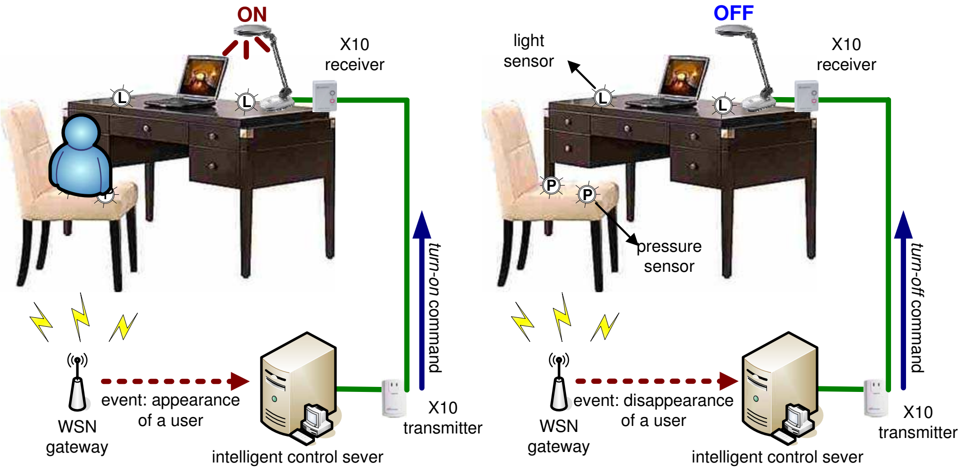
Figure 1: System architecture of the iPower.

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 Figure 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 according to 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.

- **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



(a) the user sits on the chair

(b) the user leaves

Figure 2: The "smart desk" scenario.

by SmartHome (2006). 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 the iPower system works in an intelligent building. Let us consider the five rooms in Figure 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 reduce 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 will keep on reporting that someone is on the chair, so no energy conservation commands will be issued. If

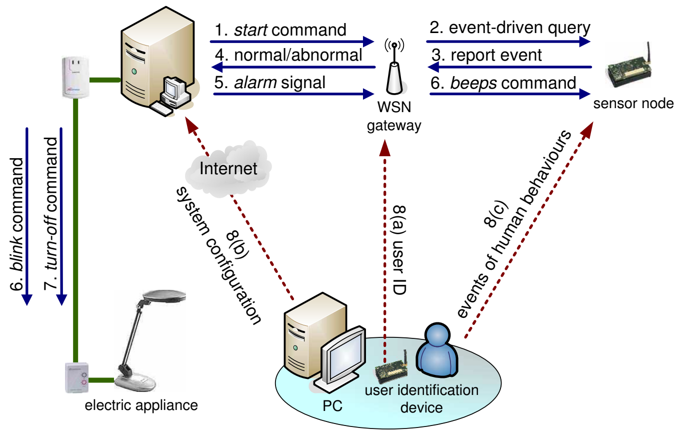


Figure 3: Message flows in the iPower system.

the smart furniture is not in use, then 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

Figure 3 illustrates the message flows 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 will notify its sensor nodes by issuing 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 will report 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 Figure 2(a)), the WSN gateway will report a *normal* status to the server. However, if it is reported that users leave the smart furniture, the WSN gateway will reset its timer and go back to step 2 to repeat the aforementioned procedure.

    (b) If sensors report any human behavior (such as sound events or change of light readings), the WSN gateway will report a *normal* status to the server. However, it will also notify the existence of people to the server so that the system will check this room's status later on.

    (c) Otherwise, the WSN gateway will report an *abnormal* status to the server 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 will warn 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 will instruct 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 will know 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.

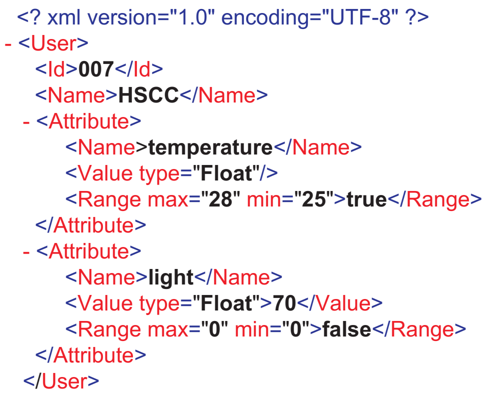
8. If there is any user in the room hearing the beeping sound or seeing blinking light, he/she can notify the server that the room is still in use by any of the following three methods:

    (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 the iPower system 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.



```
<? 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>
```

Figure 4: An example of the user profile.

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, the iPower system 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 (2006) to describe user's profiles. The current definition is illustrated in Figure 4. Specifically, the profile includes user's ID, name, and several attributes with the user's favorite temperature and brightness. For example, Figure 4 indicates that user's preference temperature is from 25 °C 28 °C and light is 70 lux.

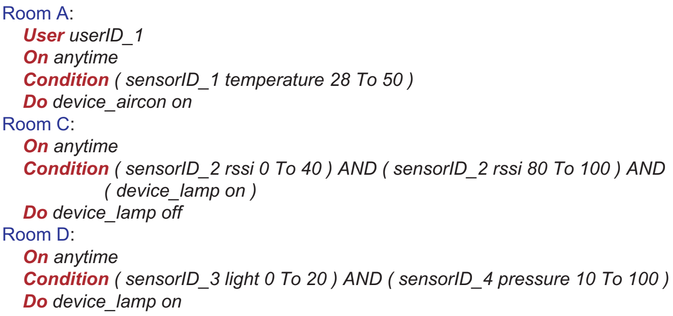
3.5 Events and Actions

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 Figure 5, we list the definition of iPower's rules, which are written in the format of *EBNF (Extended Backus-Naur Form)* (Sebesta, 1999) 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. Figure 6 shows the rules for rooms A, C, and D in Figure 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.

```
<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
```

Figure 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
```

Figure 6: Examples of the iPower's rules.

3.6 Protocol Stack

To implement the iPower system, we have designed a protocol stack in Figure 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 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 (1999), 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

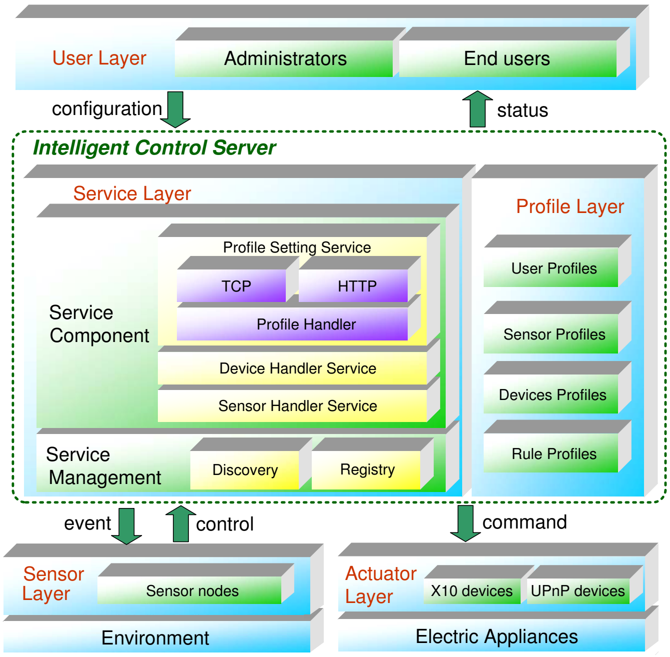


Figure 7: Protocol stack of the iPower system.

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 statues 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 in the format of XML.

- **Sensor layer:** The sensor layer controls the actions of sensor nodes. These actions 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 upper layers (i.e., service layer and profile layer). In our implementation, we choose X10 and UPnP (1999) as our device control protocols. Through these protocols, we can turn on, turn off, and adjust the electric currents of appliances.

4 Implementation Details

4.1 Hardware Specification

We use MICAz (2005) as sensor nodes. The MICAz is a 2.4 GHz, 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. Figure 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.
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 Figure 9. The user interface has an *object area*, a *monitor area*, and a *status area*. The object area provides an

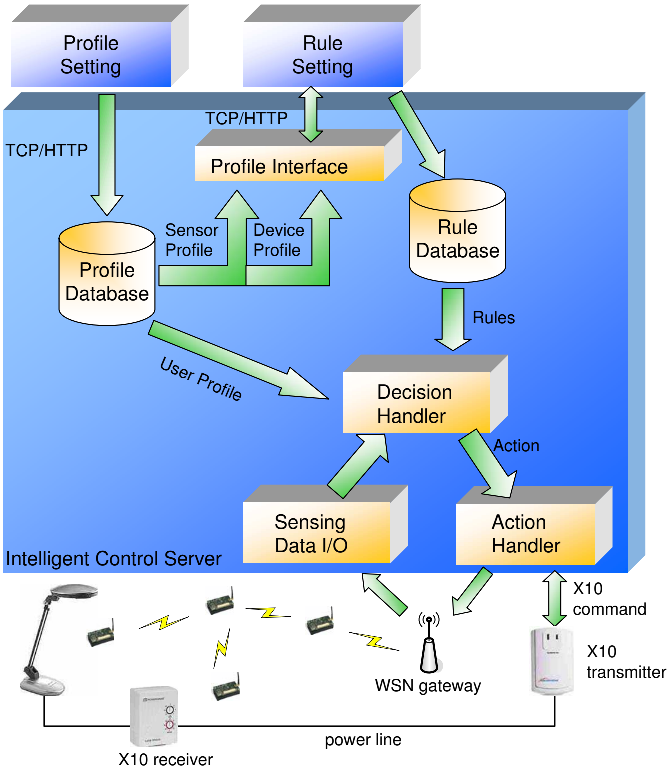
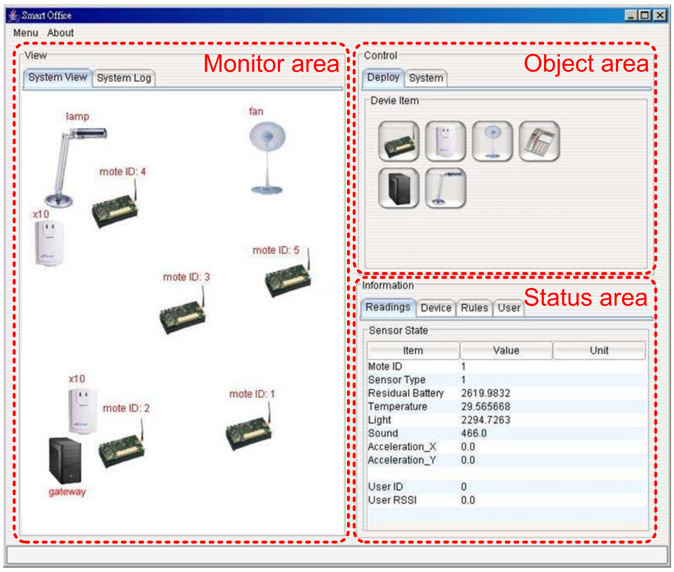


Figure 8: Design of the intelligent control server.

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 (Kleinrock, 1975) is used to model a person's behavior during every hour, as shown in Figure 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 Figure 10(b). In particular, during every twenty minutes, the person may 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.



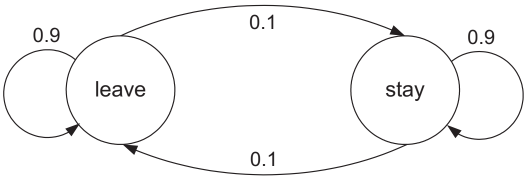
Sensor State

Item	Value	Unit
Mote ID	1	
Sensor Type	1	
Residual Battery	2619.9832	
Temperature	29.565668	
Light	2294.7263	
Sound	466.0	
Acceleration_X	0.0	
Acceleration_Y	0.0	
User ID	0	
User RSSI	0.0	

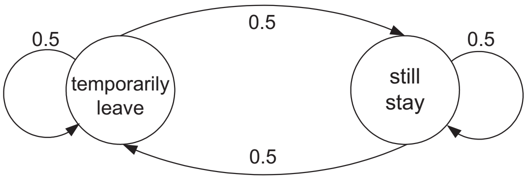
Figure 9: The user interface at the intelligent control server.

Table 1: Energy consumptions of electric appliances.

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



(a) model a person's behavior during each hour



(b) model a person's detailed behavior during each twenty minutes

Figure 10: Two-state discrete Markov models.

Figure 11 shows the energy consumption with five people during ten hours. We can observe that without iPower,

Table 2: Favorite temperatures of the five people.

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

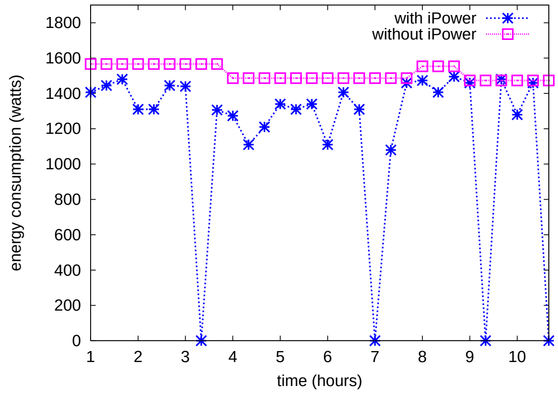


Figure 11: Energy consumptions during 10 hours.

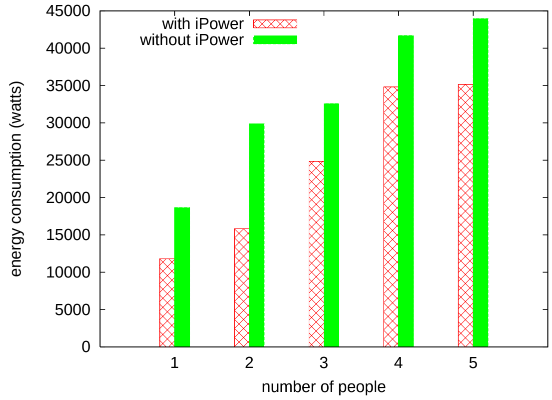


Figure 12: Total energy consumptions with different numbers of people.

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. Figure 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, we have proposed the iPower system designed for energy conservation in an intelligent building and provision of 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 and sometimes unreliable, in the future we plan to replace X10 by INSTEON (2007), 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

$$\frac{3}{3+2}\times 23+\frac{2}{3+2}=24.2\,^{\circ}\mathrm{C}.$$

  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 Figure 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 Figure 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 Branch et al. (2006); Zhuang et al. (2007); Sheng et al. (2007) 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 Singhvi (2005) suggests adjusting lamps according to users' requirements and the sunlight. Similarly, we can apply this extension in our system.

ACKNOWLEDGMENT

Y. C. Tseng's research is co-sponsored by Taiwan MoE ATU Program, by NSC grants 93-2752-E-007-001-PAE, 96-2623-7-009-002-ET, 95-2221-E-009-058-MY3, 95-2221-E-009-060-MY3, 95-2219-E-009-007, 95-2218-E-009-209, and 94-2219-E-007-009, by Realtek Semiconductor Corp., by MOEA under grant number 94-EC-17-A-04-S1-044, by ITRI, Taiwan, by Microsoft Corp., and by Intel Corp.

REFERENCES

Gassmann, O., Meixner, H., Hesse, J.W.G.J. and Gopel, W. (2001) 'Sensors in intelligent buildings: sensors applications', *Wiley-VCH.*

Schulzrinne, H., Wu, X., Sidiroglou, S. and Berger, S. (2003) 'Ubiquitous computing in home networks', *IEEE Communications Magazine*, Vol. 41, No. 11, pp.128–135.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler (2002) 'SIP: session initiation protocol', *IETF RFC 3261.*

Das, S.K., Cook, D.J., Battacharya, A., Heierman. III, E.O. and Lin, T.Y. (2002) 'The role of prediction algorithms in the mavhome smart home architecture', *IEEE Wireless Communications*, Vol. 9, No. 6, pp.77–84.

Wang, X., Dong, J.S., Chin, C.Y., Hettiarachchi, S.R. and Zhang, D. (2004) 'Semantic space: an infrastructure for smart spaces', *IEEE Pervasive Computing*, Vol. 3, No. 3, pp.32–39.

Helal, S., Zabadani, W.M.H.E., King, J., Kaddoura, Y. and Jansen, E. (2005) 'The gator tech smart house: a programmable pervasive space', *IEEE Computer*, Vol. 38, No. 3, pp.50–60.

SmartHome (2006) *X10-1132B.*
Available at: http://www.smarthome.com/.

W3C (2006) *Extensible markup language (XML).*
Available at: http://www.w3.org/TR/REC-xml/.

Sebesta, R.W. (1999) 'Concepts of programming languages, 4th ed.', *Addison-Wesley.*

OSGi Alliance (1999).
Available at: http://www.osgi.org/.

UPnP Forum (1999).
Available at: http://www.upnp.org/.

Crossbow (2005) *MOTE-KIT2400.*
Available at: http://www.xbow.com/.

Kleinrock, L. (1999) 'Queueing systems, volume 1: theory', *John Wiley & Sons, Inc..*

INSTEON (2007).
Available at: http://www.insteon.net/.

Branch, J., Giannella, C., Szymanski, B., Wolff, R. Kargupta, H. (2006) 'In-network outlier detection in wireless sensor networks', *Proceedings of IEEE International Conference on Distributed Computing Systems.*

Zhuang, Y., Chen, L., Wang, X. S. and Lian, J. (2007) 'A weighted moving average-based approach for cleaning sensor data', *Proceedings of IEEE International Conference on Distributed Computing Systems.*

Sheng, B., Li, Q., Mao, W. and Jin, W. (2007) 'Outlier detection in sensor networks', *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.219–228.

Singhvi, V., Krause, A., Guestrin, C., Garrett, J. and Matthews, H. S. (2005) 'Intelligent light control using sensor networks', *Proceedings of ACM International Conference on Embedded Networked Sensor Systems*, pp.218–229.

附件二

Y.-C. Wang and Y.-C. Tseng, "Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multi-level Coverage", IEEE Trans. on Parallel and Distributed Systems, to appear (SCI, EI).

Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multi-level Coverage

You-Chiun Wang and Yu-Chee Tseng, *Senior Member, IEEE*

***Abstract*— One of the key research issues in *wireless sensor networks* (*WSNs*) is how to efficiently deploy sensors to cover an area. In this paper, we solve the k-*coverage sensor deployment problem* to achieve multi-level (k) coverage of the area of interest $\mathcal{I}$. We consider two sub-problems: k-*coverage placement problem* and *distributed dispatch problem*. The placement problem asks how to determine the minimum number of sensors required and their locations in $\mathcal{I}$ to guarantee that $\mathcal{I}$ is k-covered and the network is connected, while the dispatch problem asks how to schedule mobile sensors to move to the designated locations according to the result computed by the placement strategy, if they are not in the current positions, such that the energy consumption due to movement is minimized. Our solutions to the placement problem consider both the binary and probabilistic sensing models, and allow an arbitrary relationship between the communication distance and sensing distance of sensors, thereby relaxing the limitations of existing results. For the dispatch problem, we propose a competition-based and a pattern-based schemes. The competition-based scheme allows mobile sensors to bid for their closest locations, while the pattern-based scheme allows sensors to derive the target locations on their own. Our proposed schemes are efficient in terms of the number of sensors required and are distributed in nature. Simulation results are presented to verify their effectiveness.**

***Index Terms*— mobile sensors, network planning, pervasive computing, sensor coverage problem, topology control, wireless sensor networks.**

I. INTRODUCTION

In the recent years, with the rapid progress in embedded micro-sensing MEMS and wireless communication technologies, *wireless sensor networks* (*WSNs*) have been studied intensively for various applications such as environment monitoring, smart home, and surveillance. A WSN usually consists of numerous wireless devices deployed in a region of interest, each able to collect and process environmental information and communicate with neighboring devices.

Sensor deployment is an essential issue in WSN because it not only determines the cost to construct the network but also affects how well a region is monitored by sensors. In this paper, we consider the sensor deployment problem for a WSN with multi-level coverage. In particular, given a region of interest, we say that the region is *k-covered* if every location in that region can be monitored by at least k sensors, where k is a given parameter. A large amount of applications may impose the requirement of $k > 1$. For instance, military or surveillance applications with a stronger monitoring requirement may impose that $k \geq 2$ to avoid leaving uncovered holes when some sensors are broken. Positioning protocols using triangulation [1] require at least three sensors (i.e., $k \geq 3$) to detect each location where an object may appear. Moreover, several strategies are based on the assumption of $k \geq 3$ to conduct data fusion [2] and to minimize the impact of sensor failure [3]. In addition, to extend a WSN's lifetime, sensors are separated into k sets, each capable of covering the whole area, to work in shifts [4]–[6].

In this paper, we address the sensor deployment problem with the following requirements:

- multiple-level coverage of the area of interest is required;
- connectivity between sensors (in terms of their communications) should be maintained;
- the area of interest may change over time;
- sensors are autonomous and mobile and thus can be dispatched to desired locations when being instructed so.

We call this the *k-coverage sensor deployment problem*, where k-level coverage of a given area of interest $\mathcal{I}$ is needed. We consider two sub-problems: k-*coverage sensor placement problem* and *distributed sensor dispatch problem*. The placement problem asks how to decide the minimum number of sensors required and their locations in $\mathcal{I}$ to ensure that $\mathcal{I}$ is k-covered and that the network is connected. Note that coverage is affected by sensors' sensing distance, while connectivity is determined by their communication distance. Considering that sensors are mobile and the area $\mathcal{I}$ may change over time, the objective of the dispatch problem is to schedule sensors to move to the designated locations (according to the result computed by the placement strategy) such that the total energy consumption of sensors due to movement can be minimized.

In the literature, one related area is the *art gallery problem* [7] in computational geometry. It intends to use the minimum number of observers to monitor a polygon area. The problem assumes that an observer can watch any point as long as line-of-sight exists and it does not address the (wireless) communication issue between observers. Another relevant issue is the *base station (BS) placement problem*. This problem discusses how to determine the optimal number and locations of BSs within an environment so as to satisfy the coverage and throughput requirements [8]. To solve this problem, many studies propose their discrete optimization models by multi-objective genetic algorithms [8], [9], parallel evolutionary algorithms [10], and simulated annealing [11] to determine the optimal placement of BSs. However, these results cannot be directly applied to our sensor placement problem.

Sensor placements for 1-coverage have been studied in several works. For example, the works in [12], [13] consider to place

---

Manuscript received Apr. 20, 2007; revised Nov. 5, 2007; accepted Nov. 14, 2007.

Y.-C. Wang is with the Department of Computer Science, National Chiao-Tung University, Hsin-Chu, 30010, Taiwan. E-mail: wangyc@cs.nctu.edu.tw

Y.-C. Tseng is with the Department of Computer Science, National Chiao-Tung University, Hsin-Chu, 30010, Taiwan. E-mail: yctseng@cs.nctu.edu.tw

sensors in a grid-like fashion to satisfy some coverage requirements, while [14] suggests to place sensors strip by strip to achieve both coverage and connectivity. In [15], a 1-coverage sensor placement method for the sensing field with obstacles is proposed. Several studies have also considered the sensor placement problem of multi-level coverage. In [3], a hexagon-like placement is proposed to guarantee the sensing field to be k-covered, under the assumption that the communication distance of sensors r_c is no smaller than twice of their sensing distance r_s. The work in [16] models the sensing field by grids and considers two kinds of sensors with different costs and sensing capabilities to be deployed in the sensing field. The objective is to make every grid point k-covered and the total cost is minimum. However, both [3] and [16] do not address the relationship between r_c and r_s. How to compute the coverage level of a given placement is addressed in [17].

Some works address the coverage and connectivity issue by assuming that there is redundancy in the initial deployment and the goal is to select a minimal set of active sensors to achieve energy conservation and maintain complete coverage of the sensing field and connectivity of the network. References [4], [18] address how to arrange some sensors to go to sleeping modes to extend the network lifetime while maintaining 1-coverage of the sensing field. On the other hand, the works [19]–[23] consider how to select these active sensors to maintain k-coverage of the sensing filed and connectivity of the network.

The use of mobile sensors has also been discussed in several works. The work [24] considers to move nodes to make the network biconnected. When events occur, [25] discusses how to move some sensors to the event locations while still maintaining complete 1-coverage of the sensing field. The works [26]–[29] study how to move sensors to enhance coverage of the sensing field by using the Voronoi diagram or attractive/repulsive forces between sensors. In [30], the sensing field is partitioned into grids, and sensors are moved from high-density grids to low-density ones to achieve more uniform coverage. The work [31] considers to add several mobile sensors into a stationary sensor network to improve the coverage and connectivity of the original network. As can be seen, the attention of prior works was mainly paid to the use of mobile sensors to improve the topology of an existing network, which is different from the sensor dispatch problem discussed in this paper. Actually, several studies [32]–[34] have proposed their design and implementation of mobile sensors. Such mobile platforms are controlled by embedded computers and mounted with sensors. These studies do motivate us to investigate the sensor dispatch problem.

In this paper, we consider more complete solutions to the k-coverage sensor deployment problem, by addressing both the placement and dispatch sub-problems. In particular, for the sensor placement problem, we allow an arbitrary relationship between sensors' communication distance r_c and their sensing distance r_s. We consider two types of sensing models: *binary* and *probabilistic*. Under the binary sensing model [14], [15], [28], a location can be either monitored or not monitored by a sensor, depending on whether the location is within the sensor's r_s range. Under the probabilistic sensing model [12], [23], [35], a location will be monitored by a sensor according to some probability function. We first consider the binary sensing model of sensors and propose two solutions to the placement problem. The first one is based on an intuitive duplication idea, while the second one is based on a more complicated interpolating scheme and thus can save more sensors. Then, we adapt these solutions to the probabilistic sensing model by properly adjusting sensing distances of sensors. For the sensor dispatch problem, we propose two distributed schemes to let sensors move to the designated locations (computed by the placement result) on their own. The first scheme assumes that sensors have the full knowledge of all target locations in the area of interest; sensors will compete with each other for moving toward their closest locations. The second scheme relaxes the above assumption in a way that sensors can derive other target locations based on several known locations, according to the patterns in our placement strategies. Therefore, we can give several locations as seeds in the beginning, and sensors will then extend their range based on the placement pattern in a distributed manner.

In this paper, we consider that the area of interest $\mathcal{I}$ may change over time (based on users' application requirements). So sensors may be dispatched in multiple rounds. Specifically, in each round when a new $\mathcal{I}$ is generated, the sink first calculates the locations to be placed with sensors in $\mathcal{I}$ by the proposed placement solutions and announces the complete or partial locations to sensors. Sensors then can automatically move to these designated locations by the proposed dispatch solutions to ensure k-coverage of $\mathcal{I}$. Because the sink does not know the current statuses and positions of mobile sensors, it cannot determine which sensor should move to which location in a centralized manner. Therefore, distributed dispatch solutions are more desirable.

Major contributions of this paper are twofold. First, our schemes allow change of the monitoring region and coverage level of the WSN in an autonomous and distributed manner. This is quite important for those applications where the region of interest may change over time. For example, one can image that a wide area is contaminated by some hazardous material such as leakage of nuclear or poisonous chemicals. By quickly providing multi-level coverage of these movable regions of pollution, the whole situation can be assessed immediately and such information can be conveniently used by the rescue team. Second, our deployment solutions are helpful in conditions where the precise initial deployment (e.g., by humans) is almost impossible because the region of interest is very dangerous or even inaccessible to people. By introducing the concept of sensor dispatch, mobile sensors can automatically move to designated locations in an efficient way and thus the region of interest can be "self-deployed" by these sensors.

The rest of this paper is organized as follows. Section II formally defines the sensor placement and dispatch problems. Sections III and IV propose our solutions to these problems. Section V presents simulation results to evaluate the proposed schemes. Section VI concludes this paper.

II. PROBLEM STATEMENT

We are given a field $\mathcal{A}$, an area of interest $\mathcal{I}$ inside $\mathcal{A}$, and a set of mobile sensors $\mathcal{S}$ resident in $\mathcal{A}$. For convenience, we assume that $\mathcal{I}$ is a rectangular region. Each sensor has a communication distance r_c and a sensing distance r_s. Sensors are homogenous, but the relationship of r_c and r_s can be arbitrary. For connectivity, we assume that two sensors can communicate with each other if their distance is no larger than r_c. For coverage, we consider both the binary and probabilistic sensing models of sensors. Under the binary sensing model, a location can be monitored by a sensor if it is within the sensor's sensing region. In this way, a location in $\mathcal{A}$ is defined as *k-covered* if it is within k sensors' sensing regions, where k is a given parameter. Under the probabilistic sensing model, the detection probability of a sensor will decay with the distance from the sensor to the monitored location. In particular, the detection probability of a location u by a sensor s_i can be evaluated as [12], [35]:

$$p(u, s_i) = \begin{cases} e^{-\varepsilon d(u,s_i)}, & \text{if } d(u, s_i) \le r_s \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where ε is a parameter indicating the physical characteristics of the sensor and $d(u, s_i)$ is the distance between u and s_i. In this way, a location in $\mathcal{A}$ is considered as k-covered if the probability that there are at least k sensors which can detect this location is no smaller than a predefined threshold p_{th}, where $0 < p_{\text{th}} < 1$. With the above definitions, an area in $\mathcal{A}$ is considered as k-covered if every location inside that area is k-covered. We assume that sensors can be aware of their own positions, which can be obtained by the *global positioning system (GPS)* [36] or other localization techniques [37], [38].

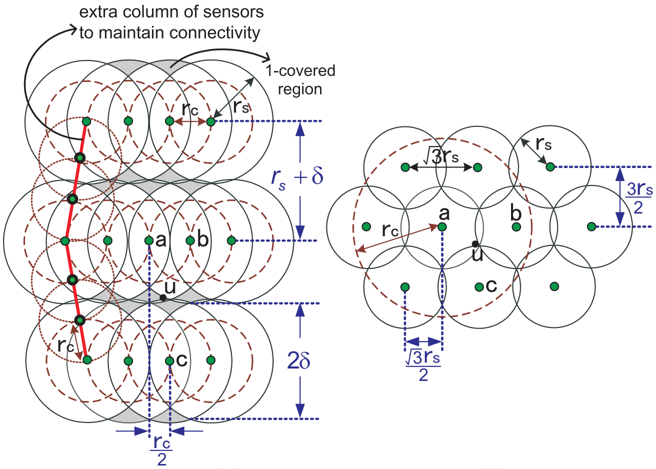
Given an integer k, the *k-coverage sensor deployment problem* can be divided into two sub-problems: *k-coverage sensor placement problem* and *distributed sensor dispatch problem*. The objective of the placement problem is to determine the minimum number of sensors required and their locations in the area of interest $\mathcal{I}$ to guarantee that $\mathcal{I}$ is k-covered and that the network is connected. Considering that mobile sensors are arbitrarily placed inside $\mathcal{A}$ and that there are sufficient sensors, the dispatch problem asks how to move sensors to designated locations (according to the result computed by the placement strategy) in a distributed manner such that the total energy consumption of sensors due to movement is minimized, i.e., $\min \sum_{i \in \mathcal{S}} e_i^{\text{move}} \times d_i$, where e_i^{move} is the energy cost for sensor i to move in one unit-distance and d_i is the total distance that sensor i has traveled. Note that here we assume that a sensor will move at a constant speed and will incur a constant rate of energy drain during its motion [30], [39]. However, the energy model may be defined in a different way from this one.

III. K-COVERAGE SENSOR PLACEMENT SCHEMES

In this section, we deal with the k-coverage sensor placement problem. We first consider the binary sensing model of sensors and propose two placement solutions. The first solution is based on a naive duplication idea, while the second solution is inspired by a more complicated interpolating concept. Then, we discuss how to adapt these placement schemes to the probabilistic sensing model.

A. The Naive Duplicate Placement Scheme

One intuitive idea to achieve a k-coverage placement is to use a good sensor placement method to determine the locations of sensors to ensure 1-coverage and connectivity in $\mathcal{I}$, and then duplicate k sensors on each designated location. For the 1-coverage placement, we adopt the method proposed in [15], which has been proved to be able to use the minimum number of sensors to achieve 1-coverage and connectivity [40]. In this 1-coverage placement, sensors are suggested to be placed row by row, where each row of sensors will guarantee continuous coverage and connectivity while adjacent rows will guarantee continuous coverage of the whole area. According to the relationship of r_c and r_s, we separate the discussion into two cases, as shown in Fig. 1. When $r_c < \sqrt{3} r_s$, sensors on each row are separated by a distance of r_c, so the connectivity of sensors in each row can be guaranteed. Since $r_c < \sqrt{3} r_s$, each row of sensors can cover a belt-like area of width 2δ, where $\delta = \sqrt{r_s^2 - \frac{1}{4} r_c^2}$. Adjacent rows will be separated vertically by a distance of $r_s + \delta$ and shifted horizontally by a distance of $\frac{r_c}{2}$. This guarantees the coverage of the whole area. When $r_c \ge \sqrt{3} r_s$, the aforementioned placement will use too many sensors, so a common regular placement of triangular lattice [41] should be adopted, where adjacent sensors will be regularly separated by a distance of $\sqrt{3} r_s$.



(a) (b)

Fig. 1. The 1-coverage sensor placement method proposed in [15]: (a) the case of $r_c < \sqrt{3} r_s$ and (b) the case of $r_c \ge \sqrt{3} r_s$.

After determining the 1-coverage placement, we can duplicate k sensors on each location to ensure k-coverage of the whole area. Note that in the case of $r_c < \sqrt{3} r_s$, since the distance between sensors on adjacent rows is larger than r_c, it is necessary to add some extra columns of sensors, where sensors on each column are separated by a distance no larger than r_c, to connect adjacent rows.

B. The Interpolating Placement Scheme

The previous duplicate scheme may result in some sub-regions in $\mathcal{I}$ that have coverage levels much higher than k. Consequently, the following interpolating placement scheme will try to balance the coverage levels of sub-regions. Observe that in Fig. 1(a), a large amount of sub-regions in a row are actually more than 1-covered. Thus, we can "reuse" these sub-regions when generating a multi-level coverage placement. Based on this observation, the interpolating placement scheme will first find out those insufficiently covered sub-regions and then place the least number of sensors to cover these regions. Note that these newly-added sensors should remain connected with the formerly placed sensors. According to the relationship of r_c and r_s, we separate the discussion into three cases.

Case of $r_c \le \frac{\sqrt{3}}{2} r_s$: From Fig. 1(a), we can observe that the insufficiently covered sub-regions (i.e., only 1-covered regions) are located between adjacent rows (marked by gray). If we add an extra row of sensors between each pair of adjacent rows in Fig. 1(a), as Fig. 2(a) shows, the coverage level of the sensing field will directly become three. Here each extra row is placed above the previous row by a distance of r_s, and neighboring sensors in each extra row are still separated by a distance of r_c. Note that in Fig. 2(a), some sensors may be placed outside the area of interest $\mathcal{I}$. This may lead to failure of the interpolating scheme to calculate a feasible solution when $\mathcal{I} = \mathcal{A}$. To solve this problem, we can place these outside sensors on the boundary of $\mathcal{I}$, as shown in Fig. 2(b). In this case, 3-coverage of $\mathcal{I}$ can still be achieved because sensors are placed more compactly.

In the case of $r_c \leq \frac{\sqrt{3}}{2} r_s$, since the distance between sensors on adjacent rows is r_s (which is larger than r_c), we have to add at least one column of sensors, each separated by a distance no larger than r_c, to connect adjacent rows.

To summarize, the previous duplicate scheme uses $3x$ rows of sensors to ensure 3-coverage of a belt-like area of width $(x-1)r_s + (x+1)\delta$, while this interpolating scheme uses only $2x+1$ rows of sensors to ensure 3-coverage of the same region. In general, for $k > 3$, we can apply $\left\lfloor \frac{k}{3} \right\rfloor$ times of the above 3-coverage placement and apply $(k \bmod 3)$ times of the 1-coverage placement to achieve k-coverage in $\mathcal{I}$. Therefore, while the duplicate placement requires kx rows of sensors to cover a region, this interpolating placement requires only $\left(\left\lfloor \frac{k}{3} \right\rfloor (2x+1) + (k \bmod 3) \cdot x\right)$ rows of sensors.

Case of $\frac{\sqrt{3}}{2} r_s < r_c \leq \frac{2+\sqrt{3}}{3} r_s$: In this case, if the desired coverage level k is two, we can directly apply the same placement in the previous case. The result is shown in Fig. 3(a). However, because the sensing distance r_s is relatively smaller (as opposed to the case of $r_c \leq \frac{\sqrt{3}}{2} r_s$), there are some sub-regions that are only 2-covered, but not 3-covered (marked by gray in Fig. 3(a)). Therefore, if the desired k is three, we need to add one extra row of sensors (marked as new' i) between each new row i and old row i, as Fig. 3(b) shows. Note that these extra rows are shifted horizontally by a distance of $\frac{r_c}{2}$ from the previous rows and neighboring sensors are separated regularly by a distance of $2r_c$. Also note that each new' row i can connect with its adjacent new row i and old row i, as shown in Fig. 3(c). In particular, because

$$\begin{aligned}&|\overline{s_n s_a}| = |\overline{s_n s_b}| = |\overline{s_n s_c}| = |\overline{s_n s_d}| \\ &= \sqrt{\left(\tfrac{1}{2} r_s\right)^2 + \left(\tfrac{1}{2} r_c\right)^2} < \tfrac{1}{2}\sqrt{\left(\tfrac{2}{\sqrt{3}} r_c\right)^2 + r_c^2} < r_c,\end{aligned}$$

the sensor s_n in a new' row i can communicate with its four neighbors s_a, s_b, s_c, and s_d in the adjacent new and old rows.

In the case of $\frac{\sqrt{3}}{2} r_s < r_c < r_s$, because the distance between sensors on adjacent rows may be larger than r_c, we have to add extra sensors between them to maintain the network connectivity. There are two cases to be discussed. When $k = 2$, we need to add at least one column of sensors between every two adjacent rows to connect them. When $k \geq 3$, because a new' row has already connected with its adjacent new and old rows, we only have to add these extra columns of sensors between each old row i and new row $i+1$, to maintain the network connectivity.

To summarize, the previous duplicate scheme uses $3x$ rows of sensors to ensure 3-coverage of a belt-like area of width $(x-1)r_s + (x+1)\delta$, while this interpolating scheme can use only $2.5x + 1$ rows of sensors to ensure 3-coverage of the same region (the third addition of rows only needs about $0.5x$ extra sensors). In general, for $k > 3$, we can also apply $\left\lfloor \frac{k}{3} \right\rfloor$ times of the above 3-coverage placement and apply $(k \bmod 3)$ times of the 1-coverage placement to achieve k-coverage in $\mathcal{I}$. Therefore, while the duplicate placement requires kx rows of sensors to cover a region, this interpolating placement only requires $\left(\left\lfloor \frac{k}{3} \right\rfloor (2.5x+1) + (k \bmod 3) \cdot x\right)$ rows of sensors.

Case of $r_c > \frac{2+\sqrt{3}}{3} r_s$: In the previous case, when r_c increases, the areas of these only 2-covered regions in Fig. 3(a) also increase. To achieve the 3-coverage placement using fewer sensors, each sensor s_n in a new' row should completely cover two only 2-covered regions (marked by gray), as shown in Fig. 3(d). In this case, we should make $|\overline{xs_n}| \leq r_s$, so we can obtain

$$\begin{aligned}&|\overline{xs_n}| = |\overline{xy}| + |\overline{ys_n}| = \left(r_c - \tfrac{\sqrt{3}}{2} r_s\right) + \tfrac{1}{2} r_c \leq r_s \\ &\Rightarrow r_c \leq \tfrac{2+\sqrt{3}}{3} r_s.\end{aligned}$$

Clearly, when $r_c > \frac{2+\sqrt{3}}{3} r_s$, sensor s_n can no longer cover the nearest two 2-covered regions. Thus, we need to add one extra sensor in the new' row to cover every 2-covered region. In this case, if the duplicate scheme uses $3x$ rows of sensors to ensure 3-coverage of a belt-like area of width $(x-1)r_s + (x+1)\delta$, this interpolating scheme should use $3x+1$ rows to achieve the same goal. Since the interpolating placement will not save sensors compared to the duplicate placement, we adopt the duplicate scheme in the case of $r_c > \frac{2+\sqrt{3}}{3} r_s$.

C. Adapting to the Probabilistic Sensing Model

In this section, we discuss how to adapt the previous two placement schemes to the probabilistic sensing model, where the detection probability of a sensor to any location follows that specified in Eq. (1). To simplify the presentation, we call the probability that a location u can be detected by at least k sensors as the *k-covered probability* of location u. To adapt our placement schemes, we first find the minimum k-covered probability $p_{\min}$ in our placement. Then, we calculate a pseudo sensing distance r_s^p according to $p_{\min}$ and p_{th}, and replace the original sensing distance r_s by r_s^p in the placement to guarantee that every location inside $\mathcal{I}$ is still k-covered under the probabilistic sensing model. In this section, we assume that $\mathcal{I} \subset \mathcal{A}$ and the desired coverage level $k \geq 3$.

1) Adaptation of the Duplicate Placement Scheme: Observing in Fig. 1, there must be a location u covered by only one sensor with a distance approximate to r_s. Such a location u is very close to the sensing boundary of the sensor placed at location a, but not inside the sensing ranges of sensors placed at locations b and c. Thus, we can derive the detection probability of location u by the sensor s_a located at a as $p(u, s_a) = e^{-\varepsilon d(u, s_a)} \approx e^{-\varepsilon r_s}$. Because the duplicate scheme places k sensors on each location specified in Fig. 1, location u will have the minimum k-covered probability $p_{\min}$. In particular, location u will be detected by a set $\mathcal{S}_a$ of k sensors placed at location a with the probability

$$p_{\min} = p(u, \mathcal{S}_a) = \prod_{s_i \in \mathcal{S}_a} p(u, s_i) \approx e^{-k\varepsilon r_s}.$$

Therefore, the duplicate scheme can guarantee a k-covered probability of at least $e^{-k\varepsilon r_s}$ in any location of the area of interest $\mathcal{I}$. On the other hand, if we want to guarantee that every location inside $\mathcal{I}$ has a k-covered probability no smaller than the given threshold p_{th}, we can calculate a pseudo sensing distance r_s^p by

$$e^{-k\varepsilon r_s^p} \geq p_{\text{th}} \Rightarrow r_s^p \leq \frac{-\ln p_{\text{th}}}{k\varepsilon}.$$

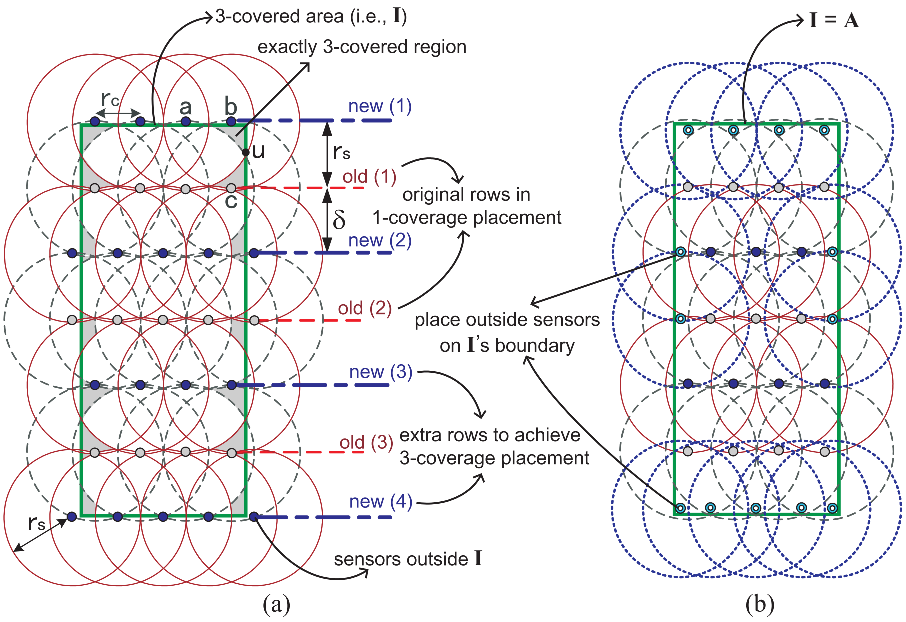


Fig. 2. The interpolating placement scheme in the case of $r_c \le \frac{\sqrt{3}}{2} r_s$: (a) the solution when $\mathcal{I} \subset \mathcal{A}$ and (b) the modified solution when $\mathcal{I} = \mathcal{A}$.

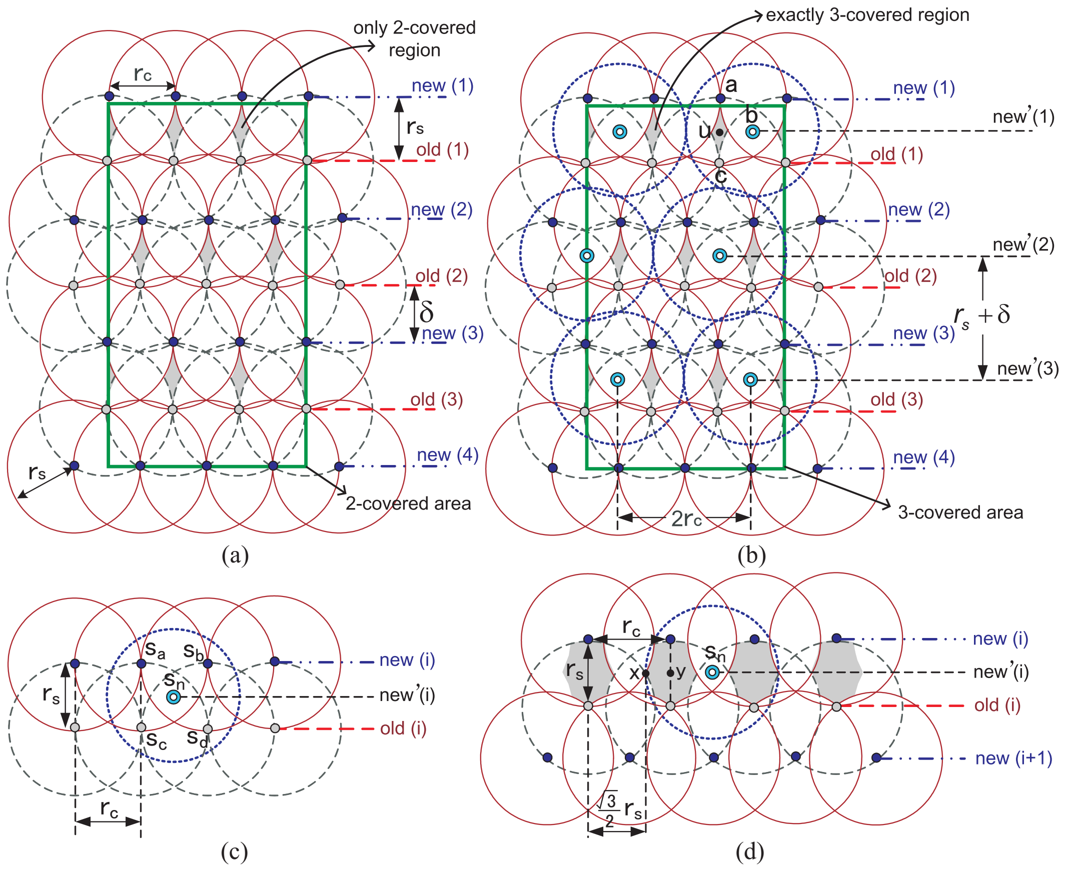


Fig. 3. The interpolating placement scheme in the case of $\frac{\sqrt{3}}{2} r_s < r_c \le \frac{2+\sqrt{3}}{3} r_s$: (a) the placement for $k = 2$, (b) the placement for $k = 3$, (c) an example to show that the connectivity between a new' row and its adjacent rows is guaranteed, and (d) the boundary case.

With the above argument, if we replace r_s by r_s^p when executing the duplicate scheme, we can guarantee that $\mathcal{I}$ is still k-covered under the probabilistic sensing model.

2) Adaptation of the Interpolating Placement Scheme: According to the relationship of r_c and r_s, we separate the discussion into three cases.

Case of $r_c \le \frac{\sqrt{3}}{2} r_s$: We first consider the case of $k = 3$. Observing in Fig. 2(a), there are some sub-regions covered by exactly three sensors (marked by gray). Among these regions, there will be a location u with the minimum 3-covered probability.

In particular, such a location u will be covered by sensors s_a, s_b, and s_c located at a, b, and c, respectively. Since $d(u, s_a) = r_s$, we have $p(u, s_a) = e^{-\varepsilon r_s}$. In addition, because

$$\begin{aligned} d(u, s_b) = d(u, s_c) &< \sqrt{\left(\tfrac{1}{2} r_c\right)^2 + \left(\tfrac{1}{2} r_s\right)^2} \\ &\leq \tfrac{1}{2}\sqrt{\left(\tfrac{\sqrt{3}}{2} r_s\right)^2 + r_s^2} = \tfrac{\sqrt{7}}{4} r_s, \end{aligned}$$

we can obtain $p(u, s_b) = p(u, s_c) > e^{-\frac{\sqrt{7}}{4}\varepsilon r_s}$. Thus, the 3-covered probability of location u is

$$p_{abc} = p(u, s_a) \cdot p(u, s_b) \cdot p(u, s_c) > e^{-\frac{2+\sqrt{7}}{2}\varepsilon r_s}.$$

In the interpolating scheme, when $k \geq 3$ is a multiple of three, we will place $\frac{k}{3}$ sensors on each location specified in Fig. 2(a). Therefore, we can obtain

$$p_{\min} = (p_{abc})^{\frac{k}{3}} > e^{-\frac{2+\sqrt{7}}{6} k \varepsilon r_s}. \quad (2)$$

When k is not a multiple of three, we will add extra ($k \bmod 3$) sensors on each location in old rows in Fig. 2(a). Thus, we have

$$p_{\min} = (p_{abc})^{\frac{k}{3}} \cdot (p(u, s_c))^{(k \bmod 3)}. \quad (3)$$

By combining Eqs. (2) and (3), we can derive that

$$p_{\min} > e^{-(\frac{2+\sqrt{7}}{6} k + \frac{\sqrt{7}}{4}(k \bmod 3))\varepsilon r_s}. \quad (4)$$

To calculate the pseudo sensing distance r_s^p, we can make

$$\begin{aligned} & e^{-(\frac{2+\sqrt{7}}{6} k + \frac{\sqrt{7}}{4}(k \bmod 3))\varepsilon r_s^p} \geq p_{\text{th}} \\ \Rightarrow & r_s^p \leq \frac{-\ln p_{\text{th}}}{\left(\frac{2+\sqrt{7}}{6} k + \frac{\sqrt{7}}{4}(k \bmod 3)\right)\varepsilon}. \end{aligned}$$

Case of $\frac{\sqrt{3}}{2} r_s < r_c \leq \frac{2+\sqrt{3}}{3} r_s$: Again, we first consider the case of $k = 3$. Observing in Fig. 3(b), there are some sub-regions covered by exactly three sensors (marked by gray). Among these regions, there will be a location u that has the minimum 3-covered probability. In particular, such a location u is covered by sensors s_a, s_b, and s_c at locations a, b, and c, respectively. Because $d(u, s_a) = d(u, s_c) = \frac{1}{2} r_s$, we have $p(u, s_a) = p(u, s_c) = e^{-\frac{1}{2}\varepsilon r_s}$. Moreover, since $d(u, s_b) = \frac{1}{2} r_c \leq \frac{2+\sqrt{3}}{6} r_s$, we can obtain $p(u, s_b) \geq e^{-\frac{2+\sqrt{3}}{6}\varepsilon r_s}$. Thus, the 3-covered probability of location u will be

$$p_{abc} = p(u, s_a) \cdot p(u, s_b) \cdot p(u, s_c) \geq e^{-\frac{8+\sqrt{3}}{6}\varepsilon r_s}.$$

Similar to Eq. (4), when $k \geq 3$, we can derive the minimum k-covered probability $p_{\min}$ as

$$(p_{abc})^{\frac{k}{3}} \cdot (p(u, s_c))^{(k \bmod 3)} \geq e^{-(\frac{8+\sqrt{3}}{18} k + \frac{1}{2}(k \bmod 3))\varepsilon r_s}.$$

Again, the pseudo sensing distance r_s^p can be derived as

$$\begin{aligned} & e^{-(\frac{8+\sqrt{3}}{18} k + \frac{1}{2}(k \bmod 3))\varepsilon r_s^p} \geq p_{\text{th}} \\ \Rightarrow & r_s^p \leq \frac{-\ln p_{\text{th}}}{\left(\frac{8+\sqrt{3}}{18} k + \frac{1}{2}(k \bmod 3)\right)\varepsilon}. \end{aligned}$$

Case of $r_c > \frac{2+\sqrt{3}}{3} r_s$: In this case, since the duplicate scheme is adopted, we can obtain $p_{\min} \geq e^{-k\varepsilon r_s}$ and $r_s^p \leq \frac{-\ln p_{\text{th}}}{k\varepsilon}$.

Table I summarizes the approximate threshold values of the minimum k-covered probability $p_{\min}$ and the pseudo sensing distance r_s^p in the interpolating scheme.

IV. DISTRIBUTED SENSOR DISPATCH SCHEMES

After determining the locations to be placed with sensors, the next issue is how to move existing sensors in the field $\mathcal{A}$ to the designated locations in $\mathcal{I}$ such that the energy consumption of sensors due to movement can be minimized. Since we cannot obtain the current statuses and positions of sensors, it is impossible to compute an optimal solution to dispatch sensors in a centralized manner. Thus, we propose two distributed dispatch schemes in this section.

A. The Competition-based Dispatch Scheme

In this scheme, when an area of interest $\mathcal{I}$ is determined, the sink will first calculate a set of locations $\mathcal{L} = \{(x_1, y_1, n_1), (x_2, y_2, n_2), \cdots, (x_m, y_m, n_m)\}$ to be placed with sensors in $\mathcal{I}$, according to our placement schemes in Section III. Here each element (x_j, y_j, n_j), $j = 1..m$, indicates that n_j sensors need to be placed on location (x_j, y_j). The sink then broadcasts $\mathcal{L}$ to all sensors.

On receiving $\mathcal{L}$ from the sink, sensors will compete with each other to move toward these locations. In particular, each sensor s_i will construct a table $OCC[1..m]$ such that every entry $OCC[j] = \{(s_{j_1}, d_{j_1}), (s_{j_2}, d_{j_2}), \ldots, (s_{j_\alpha}, d_{j_\alpha})\}$, $\alpha \leq n_j$, contains the set of sensors that have already moved into, or are still on their ways moving toward, location (x_j, y_j) and their corresponding distances to (x_j, y_j). Specifically, each record $(s_{j_\beta}, d_{j_\beta})$, $\beta = 1..\alpha$, indicates that sensor s_{j_β} has chosen to cover location (x_j, y_j) and its current estimated distance to (x_j, y_j) is d_{j_β}. When $d_{j_\beta} = 0$, it means that sensor s_{j_β} has already arrived at (x_j, y_j). Initially, $OCC[j] = \emptyset$ for all $j = 1..m$. To simplify the presentation, we say that a location (x_j, y_j) is *covered* if a sufficient number n_j of sensors have committed to move toward (x_j, y_j) (i.e., $|OCC[j]| = n_j$); otherwise, (x_j, y_j) is *uncovered.* A sensor s_i is *engaged* if it has chosen to move to, or already moved into, any location in $\mathcal{L}$; otherwise, it is *free* or *terminated.* The initial state of each sensor is free. A free sensor will try to become engaged and move toward a destination. When the free sensor finds that there is no location that it can cover, it will enter the terminated state. Fig. 4 illustrates the state transition diagram of a sensor.

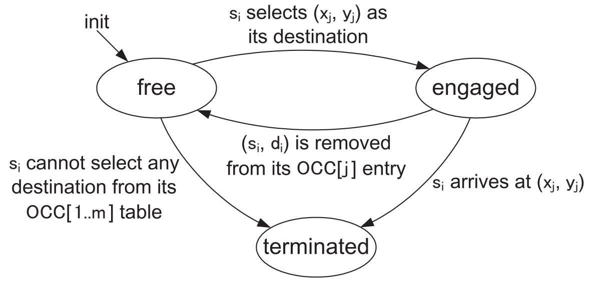


Fig. 4. The state transition diagram of each sensor s_i in the competition-based dispatch scheme.

When the state of a sensor s_i is free, it will check its $OCC[1..m]$ table to select a location in $\mathcal{L}$ as its destination. The selection is as follows:

- The first priority is to consider uncovered locations. Specifically, if there is a location (x_j, y_j) such that $|OCC[j]| < n_j$, (x_j, y_j) will be considered first. If multiple locations are qualified, the location (x_j, y_j) such that $d(s_i, (x_j, y_j))$

TABLE I

APPROXIMATE THRESHOLD VALUES OF $p_{\min}$ AND r_s^p IN THE INTERPOLATING PLACEMENT SCHEME.

case	$p_{\min}$	r_s^p
$r_c \leq \frac{\sqrt{3}}{2} r_s$	$e^{-(0.77k+0.66(k \bmod 3))\varepsilon r_s}$	$\frac{-\ln p_{\text{th}}}{(0.77k+0.66(k \bmod 3))\varepsilon}$
$\frac{\sqrt{3}}{2} r_s < r_c \leq \frac{2+\sqrt{3}}{3} r_s$	$e^{-(0.54k+0.5(k \bmod 3))\varepsilon r_s}$	$\frac{-\ln p_{\text{th}}}{(0.54k+0.5(k \bmod 3))\varepsilon}$
$r_c > \frac{2+\sqrt{3}}{3} r_s$	$e^{-k\varepsilon r_s}$	$\frac{-\ln p_{\text{th}}}{k\varepsilon}$

is minimized will be selected, where $d(s_i, (x_j, y_j))$ is the distance between s_i's current position to (x_j, y_j). In this case, s_i will add a record $(s_i, d(s_i, (x_j, y_j)))$ in its $OCC[j]$ entry and enter the engaged state.

- If all locations in $\mathcal{L}$ are already covered (i.e., $|OCC[j]| = n_j$, $\forall j = 1..m$), s_i selects a location (x_j, y_j) such that there is a record $(s_k, d_k) \in OCC[j]$ and $e_i^{\text{move}} \times d(s_i, (x_j, y_j)) < e_k^{\text{move}} \times d_k$. If multiple locations are qualified, the location (x_j, y_j) such that $e_k^{\text{move}} \times d_k - e_i^{\text{move}} \times d(s_i, (x_j, y_j))$ is maximized will be selected. In this case, s_i will replace the original record $(s_k, d_k) \in OCC[j]$ by the new record $(s_i, d(s_i, (x_j, y_j)))$ in its $OCC[j]$ entry and enter the engaged state. Here both sensors s_i and s_k are competing for the same location (x_j, y_j). Because s_i can consume less energy to move to (x_j, y_j), we should replace s_k's mission by s_i to reduce the total moving energy. Note that sensor s_k will realize that it loses the competition when it receives an update message originated from s_i later.

When sensor s_i becomes engaged, it begins moving toward its destination. Otherwise, s_i will enter the terminated state because it does not need to cover any location.

For the maintenance purpose, each sensor s_i will periodically perform the following two actions:

- Update the content of its $OCC[1..m]$ table. Specifically, for each $(s_{j_\beta}, d_{j_\beta}) \in OCC[j]$, $j = 1..m$, we decrease d_{j_β} by the expected moving distance of s_{j_β} during the past period of time, unless $d_{j_\beta} = 0$.
- Broadcast s_i's current status to its one-hop neighbors, including its ID, its moving energy cost e_i^{move}, its $OCC[1..m]$ table, and its current position and state.

The above actions can be controlled by setting two timers $T_{\text{update\_OCC}}$ and $T_{\text{broadcast}}$. Note that the update of $OCC[1..m]$ table is based on the assumption that sensors all move in the same constant speed. If this assumption is not valid, d_{j_β} is only an estimated distance for sensor s_{j_β} to location (x_j, y_j). In this case, we can make an extension by including each sensor's moving speed in its broadcast message.

When a sensor s_i receives an update message from another sensor s_k, two actions will be taken:

- First, s_i has to update its $OCC[1..m]$ table as follows. Let us denote by $OCC_i[1..m]$ and $OCC_k[1..m]$ the tables of s_i and s_k, respectively. For each $j = 1..m$, we calculate the union $U_j = OCC_i[j] \cup OCC_k[j]$. If $|U_j| \leq n_j$, we will replace $OCC_i[j]$ by U_j. Otherwise, it means that there are too many sensors scheduled to cover (x_j, y_j), in which case we will truncate those records (s_k, d_k) in U_j that have more moving energy (i.e., large value of $e_k^{\text{move}} \times d_k$), until the size $|U_j| = n_j$. Then we replace $OCC_i[j]$ by the truncated U_j. Note that the above merge of two sets may lead to a special case that s_i was in the original $OCC_i[j]$ entry, but is not in the new $OCC_i[j]$ entry. In this case, it means that s_i has been replaced by some other sensors with a lower moving energy to (x_j, y_j). If so, sensor s_i should change its state from engaged to free and then reselect another destination.
- After the above merge, if s_i remains engaged, say, with (x_j, y_j) as its destination, we will conduct the following optimization. We will check if
$$e_i^{\text{move}} \times d(s_i, (x_l, y_l)) + e_k^{\text{move}} \times d(s_k, (x_j, y_j)) < e_i^{\text{move}} \times d(s_i, (x_j, y_j)) + e_k^{\text{move}} \times d(s_k, (x_l, y_l)),$$
where (x_l, y_l) is the current destination of s_k. If so, it means that the total moving energy of s_i and s_k can be reduced if we exchange their destinations. In this case, s_i will communicate with s_k for this trade. Once the trade is confirmed, s_i will replace the records (s_i, d_i) and (s_k, d_k) in $OCC_i[j]$ and $OCC_i[l]$ by the new records $(s_k, d(s_k, (x_j, y_j)))$ and $(s_i, d(s_i, (x_l, y_l)))$, respectively. Note that s_k will also update its $OCC_k[j]$ and $OCC_k[l]$ entries with the same records.

In the above steps, if any entry in $OCC_i[1..m]$ table has been changed, s_i will broadcast the modified content to its direct neighbors.

When a sensor s_i is in the engaged state, it will keep moving toward its destination (x_j, y_j). When s_i arrives at (x_j, y_j), it will change its state to terminated and begin its monitoring job at the designated location. Meanwhile, it still executes the maintenance actions until the sink commands it to stop. Since the sink will eventually observe that $\mathcal{I}$ is k-covered (by receiving the sensing reports from sensors), it can notify all sensors to exit from the dispatch algorithm. Fig. 5 summarizes the main steps of the competition-based scheme. Theorem 1 shows that the competition-based scheme can guarantee $\mathcal{I}$ to be k-covered if there are sufficient sensors.

Step 1:	Sink broadcasts a set of location $\mathcal{L}$ to all sensors.
Step 2:	Each sensor constructs an OCC table from $\mathcal{L}$ and sets its state as free.
Step 3:	A free sensor s_i selects an uncovered (or replaceable) location (x_j, y_j) as its destination. If s_i cannot find such (x_j, y_j), it enters the terminated state and stops. Otherwise, s_i becomes engaged and moves to (x_j, y_j).
Step 4:	Each sensor periodically broadcasts and updates its OCC table. If the record (s_i, d_i) is discarded during the updating operation, sensor s_i changes its state to free and goes back to step 3.
Step 5:	When sensor s_i arrives at (x_j, y_j), it changes to terminated and begins the monitoring job. s_i still conducts step 4 until the sink commands to stop.

Fig. 5. The main steps of the competition-based dispatch scheme.

Theorem 1: Given an area $\mathcal{I} \subseteq \mathcal{A}$, the competition-based dispatch scheme guarantees that $\mathcal{I}$ will be eventually k-covered if there are sufficient mobile sensors inside $\mathcal{A}$.

Proof: Since the proposed placement schemes in Section III can compute a set of locations $\mathcal{L}$ inside $\mathcal{I}$ to be placed with sensors to ensure that $\mathcal{I}$ is k-covered, we only have to show that every location $(x_j, y_j) \in \mathcal{L}$ will eventually be covered by n_j sensors. Observe that in the competition-based scheme, it is guaranteed that an engaged sensor s_i will eventually arrive at location (x_j, y_j) if the record (s_i, d_i) remains in s_i's $OCC[j]$ entry. However, if the record (s_i, d_i) is removed during s_i's movement toward (x_j, y_j), it means that either another sensor s_k trades its current destination (x_l, y_l) with s_i or s_i loses the competition. In the former case, the locations (x_j, y_j) and (x_l, y_l) will be covered by s_k and s_i, respectively. In the latter case, it means that (x_j, y_j) has already been committed by more than n_j sensors, so it is safe for sensor s_i to give up the location (x_j, y_j). In this case, s_i has to reselect another destination. If s_i finds that $|OCC[j]| = n_j$ for all $j = 1..m$, then every location in $\mathcal{L}$ has been committed by sufficient sensors. So all locations will be eventually covered by n_j sensors. Therefore, the competition-based dispatch scheme guarantees that $\mathcal{I}$ will be eventually k-covered if there are sufficient mobile sensors. ■

Remark 1: Theorem 1 also shows that the competition-based scheme can converge when there are sufficient sensors. However, when the number of sensors is not sufficient to cover $\mathcal{I}$, the competition-based scheme still guarantees that each sensor can eventually find a location to cover. In this case, if the sink knows in advance the total number of mobile sensors, it can also notify all sensors to exit from the dispatch algorithm earlier. If this assumption is not valid, a time-out mechanism should be applied to guarantee the convergence of this dispatch scheme. In this case, the sink can maintain a timer to decide when to terminate the dispatch algorithm.

Remark 2: There is a hidden assumption that the initial deployment of the network is connected, so sensors can receive the target locations $\mathcal{L}$ from the sink safely. For those sensors isolated from the initial network, they can only receive $\mathcal{L}$ when other sensors with $\mathcal{L}$ move close to them (by step 4). However, to alleviate the worst situation that some sensors may be always isolated from other sensors, we can enforce sensors to roam around randomly from time to time to increase the probability of information exchange.

Remark 3: Most message exchanges in the competition-based scheme rely on broadcast mechanism (steps 1 and 4). Since sensors will periodically broadcast their statuses and OCC tables, this scheme can tolerate the slight loss of messages. Thus, no extra acknowledgement mechanism is required to ensure proper operations of the competition-based scheme. In addition, to ensure sensors correctly update their OCC tables, a timestamp or a sequence number is needed in each message to distinguish new from old messages.

Remark 4: In the competition-based scheme, sensors will find out and move to their destinations on their own, without any interaction with the sink. The sink only announces available target locations in the beginning. Thus, the competition-based scheme is essentially distributed.

B. The Pattern-based Dispatch Scheme

The previous competition-based scheme assumes that every sensor has the full knowledge of all target locations inside $\mathcal{I}$. This requires the sink to execute the placement scheme for $\mathcal{I}$ and then to broadcast all target locations to every sensor. Consequently, in this section, we propose a pattern-based dispatch scheme which allows sensors to derive the target locations on their own, thus relaxing the above limitation.

TABLE II

COORDINATES OF THE SIX NEIGHBORS OF A SENSOR LOCATED AT (x, y) IN THE DUPLICATE PLACEMENT SCHEME.

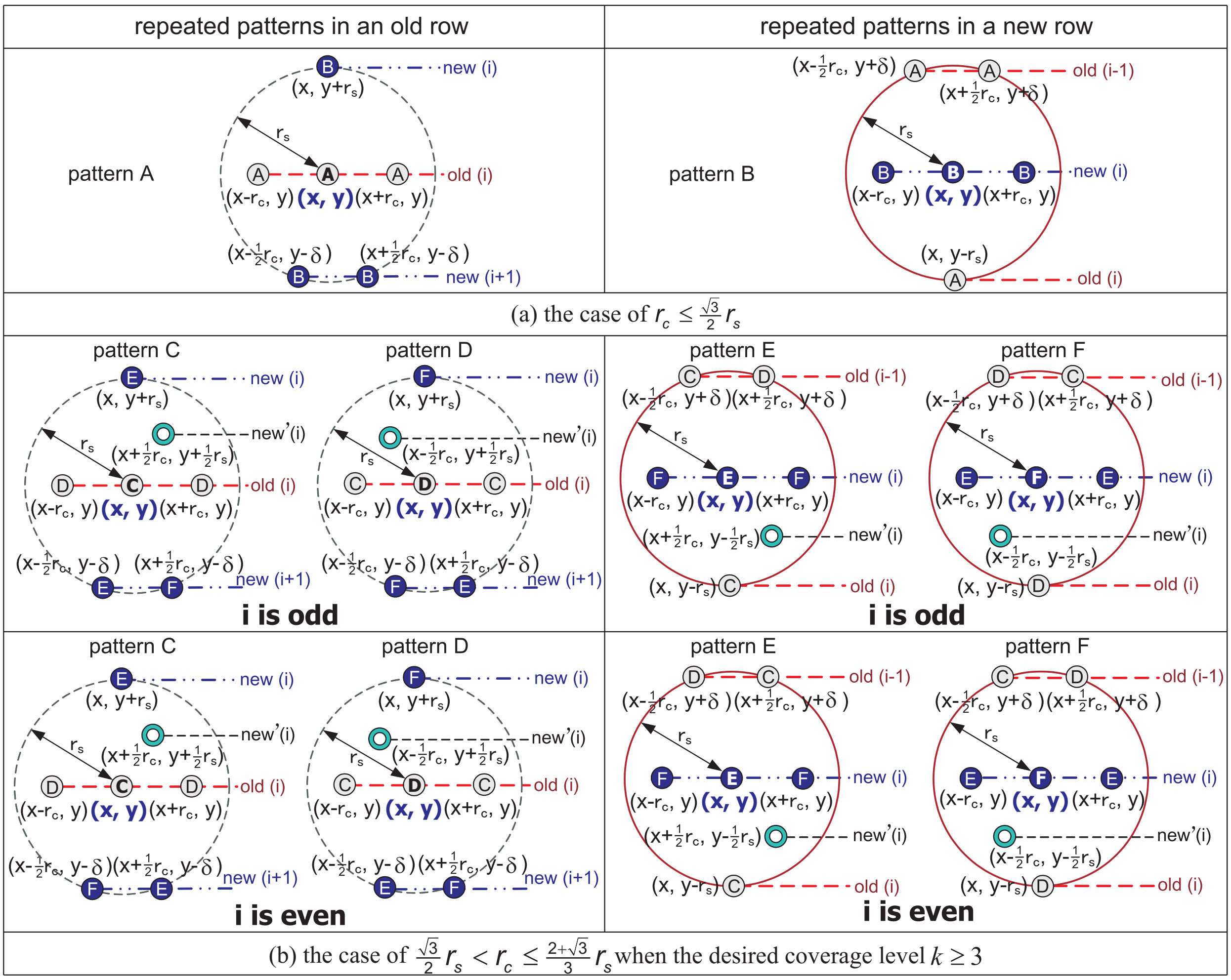
neighbor	$r_c < \sqrt{3}r_s$	$r_c \geq \sqrt{3}r_s$
n_1	$(x + r_c, y)$	$(x + \sqrt{3}r_s, y)$
n_2	$(x + \frac{1}{2}r_c, y - r_s - \delta)$	$(x + \frac{\sqrt{3}}{2}r_s, y - \frac{3}{2}r_s)$
n_3	$(x - \frac{1}{2}r_c, y - r_s - \delta)$	$(x - \frac{\sqrt{3}}{2}r_s, y - \frac{3}{2}r_s)$
n_4	$(x - r_c, y)$	$(x - \sqrt{3}r_s, y)$
n_5	$(x - \frac{1}{2}r_c, y + r_s + \delta)$	$(x - \frac{\sqrt{3}}{2}r_s, y + \frac{3}{2}r_s)$
n_6	$(x + \frac{1}{2}r_c, y + r_s + \delta)$	$(x + \frac{\sqrt{3}}{2}r_s, y + \frac{3}{2}r_s)$

Observe that our placement schemes in Section III actually place sensors with some regular patterns. Specifically, in the duplicate placement scheme, sensors will be placed in a hexagon-like fashion. Thus, each sensor at the location (x, y) can derive its potential six neighbors' positions according to Table II. When the interpolating placement scheme is adopted, the pattern will be changed according to the relationship of r_c and r_s:

- $r_c \leq \frac{\sqrt{3}}{2}r_s$. Recall the placement in Fig. 2(a). There are two patterns A and B, which will be repeated in each old row and new row, as shown in Fig. 6(a). Therefore, a sensor s_i located at (x, y) can derive its five neighbors' positions according to its pattern. Moreover, s_i can also derive the patterns of its neighbors depending on its own pattern (indicated by the letters inside circles in Fig. 6(a)).
- $\frac{\sqrt{3}}{2}r_s < r_c \leq \frac{2+\sqrt{3}}{3}r_s$. In this case, if the desired coverage level k is two, we can directly apply the patterns A and B in the previous case. However, when $k \geq 3$, there is an extra row (marked as new') between each new and old rows in Fig. 3(b). This will result in four placement patterns C, D, E, and F, as Fig. 6(b) shows, depending on a sensor's position and its row number. Thus, a sensor s_i located at (x, y) can derive its six neighbors' positions based on its pattern. In addition, s_i can also derive the patterns of its neighbors according to its own pattern (indicated by the letters inside circles in Fig. 6(b)). Note that we do not derive the patterns for sensors at the extra new' rows (although this is feasible, deriving these patterns will complicate the problem a lot). That's why sensors marked by double circles are not assigned with any pattern letter.
- $r_c > \frac{2+\sqrt{3}}{3}r_s$. In this case, since the duplicate placement scheme is adopted, a sensor can compute its neighbors' positions according to Table II.

To summarize, the above observations allow a sensor to derive its direct neighbors (within the r_s range) as well as the patterns to be used by them. This property allows us to expand from a partial deployment to a full deployment of sensors in $\mathcal{I}$. Note that since the values of r_c and r_s are known, each sensor can maintain a small table to record the related positions of its neighbors in each pattern. Thus, the calculation of neighbors' positions can be translated to a simple table lookup procedure.

With the above property, the pattern-based dispatch scheme works as follows. Each sensor initially keeps a set of seed locations $\mathcal{L}' = \{(x_1, y_1, n_1, \rho_1), (x_2, y_2, n_2, \rho_2), \cdots, (x_\alpha, y_\alpha, n_\alpha, \rho_\alpha)\}$, which is a partial list of locations to be



(a) the case of $r_c \le \frac{\sqrt{3}}{2} r_s$

(b) the case of $\frac{\sqrt{3}}{2} r_s < r_c \le \frac{2+\sqrt{3}}{3} r_s$ when the desired coverage level $k \ge 3$

Fig. 6. The repeated patterns in the interpolating placement scheme.

placed with sensors in $\mathcal{I}$, where ρ_j is the pattern used by the sensor at location (x_j, y_j). Clearly, $\mathcal{L}'$ can be considered as a subset of $\mathcal{L}$. Note that these seed locations should be sparsely distributed over $\mathcal{I}$, so that sensors may not crowd into only few locations in the beginning. Each sensor then executes the competition-based scheme to contend for their closest locations in $\mathcal{L}'$. However, the original steps 3 and 5 in the competition-based scheme should be modified as follows:

- **Step 3':** A free sensor s_i will try to select a location in $\mathcal{L}'$ as its destination. If s_i cannot find any available location from its current $OCC[\cdot]$ table, it will calculate some new locations based on the known locations and their patterns in the $OCC[\cdot]$ table. Then, s_i will try to select a destination among these newly-derived locations. However, if s_i cannot calculate any new location from its current $\mathcal{L}'$ (which means that $\mathcal{L}' = \mathcal{L}$), s_i will enter the terminated state since it does not need to cover any location.
- **Step 5':** When an engaged sensor s_i arrives at its destination, it will derive some new locations from its current $\mathcal{L}'$ and add the corresponding new entries in its $OCC[\cdot]$ table.

Corollary 1: Given an area $\mathcal{I} \subseteq \mathcal{A}$, the pattern-based dispatch scheme guarantees that $\mathcal{I}$ can be k-covered if there are sufficient mobile sensors inside $\mathcal{A}$.

Proof: From Theorem 1, we know that the competition-based dispatch scheme can ensure that $\mathcal{I}$ is k-covered if there are sufficient sensors. Since the pattern-based scheme works similarly to the competition-based one, we only need to show that the complete information of $\mathcal{L}$ can be eventually known by all sensors. Observing in the pattern-based scheme, since a sensor can either derive new locations by itself (according to steps 3' and 5') or learn new locations from other sensors (by step 4 in the competition-based scheme), the complete information of $\mathcal{L}$ can be propagated throughout the whole network. Therefore, the pattern-based scheme also guarantees $\mathcal{I}$ to be k-covered when there are sufficient mobile sensors. ■

V. EXPERIMENTAL RESULTS

In this section, we present some experimental results to evaluate the performances of the proposed schemes. The evaluation includes three parts. First, we measure the numbers of sensors required by different placement schemes discussed in Section III. Second, we verify the effectiveness of our sensor dispatch schemes proposed in Section IV. Finally, we study the effect of seed locations on the pattern-based dispatch scheme.

A. Evaluations of the Sensor Placement Schemes

The first experiment measures the numbers of sensors required by different placement schemes. We design an area of interest

$\mathcal{I}$ as a 1000 m (meter) × 1000 m square region to be placed with sensors. The communicate distance r_c is set to 10 m, which is approximate to that specified in IEEE 802.15.4 standard [42] in an indoor environment. To reflect the relationships of $r_c < \frac{\sqrt{3}}{2}r_s$, $r_c \approx \frac{\sqrt{3}}{2}r_s$ (boundary case), $\frac{\sqrt{3}}{2}r_s < r_c < \frac{2+\sqrt{3}}{3}r_s$, $r_c \approx \frac{2+\sqrt{3}}{3}r_s$ (boundary case), and $r_c > \frac{2+\sqrt{3}}{3}r_s$, we set the sensing distance r_s to 15 m, 11.55 m, 10 m, 8.04 m, and 6 m, respectively. We mainly compare the results of the duplicate and interpolating placement schemes discussed in Section III. For baseline reference, we also calculate the theoretical lower bound of the number of sensors required by $\left\lceil \frac{|\mathcal{I}|}{\pi r_s^2} \right\rceil \times k$, where $|\mathcal{I}|$ is the area of $\mathcal{I}$ (i.e., $10^6\,\text{m}^2$ in this experiment). Note that the above lower bound can never be achieved because it does not consider the connectivity and coverage overlapping between sensors.

Fig. 7 illustrates the numbers of sensors required when the desired coverage level k increases from two to seven. When $k = 2$, the interpolating scheme requires slightly more sensors compared with the duplicate scheme, because the former needs an extra row of sensors to ensure 2-coverage of $\mathcal{I}$'s boundary. However, when $k \geq 3$, the interpolating scheme can save approximately 19.4% $\sim$ 32.5% and 10.1% $\sim$ 16.8% sensors as opposed to the duplicate scheme in the case of $r_c \leq \frac{\sqrt{3}}{2}r_s$ and $\frac{\sqrt{3}}{2}r_s < r_c \leq \frac{2+\sqrt{3}}{3}r_s$, respectively. When $r_c > \frac{2+\sqrt{3}}{3}r_s$, the interpolating scheme works the same as the duplicate scheme, so they require the same number of sensors. Note that when r_c becomes larger, our placement schemes will be dominated by the value of r_s. So the numbers of sensors required by the duplicate and interpolating schemes are closer to the theoretical lower bound as r_c increases.

B. Performances of the Sensor Dispatch Schemes

In the second experiment, we estimate the total moving energy and average moving distance of sensors when different dispatch schemes are adopted. We design a field $\mathcal{A}$ as a 600 m × 600 m square region. The area of interest $\mathcal{I}$ is a 300 m × 300 m square region located at the center of $\mathcal{A}$. Three scenarios, namely *hollow*, *right*, and *central*, are considered. In the hollow scenario, sensors are randomly distributed inside the region of $\mathcal{A} - \mathcal{I}$. In the right scenario, sensors are arbitrarily placed inside a 150 m × 600 m rectangle region located at the right side of $\mathcal{A} - \mathcal{I}$. In the central scenario, sensors are initially concentrated inside a 100 m × 100 m square region located at the center of $\mathcal{I}$. With the setting of (r_c, r_s) = (34.7 m, 20.0 m), (24.1 m, 13.9 m), (19.3 m, 11.1 m), (16.7 m, 9.62 m), (14.9 m, 8.6 m), (13.4 m, 7.71 m), and (12.5 m, 7.16 m), we can obtain 100, 200, 300, 400, 500, 600, and 700 locations to be placed with sensors inside $\mathcal{I}$, respectively, according to the interpolating placement scheme (in the case of $r_c > \frac{2+\sqrt{3}}{3}r_s$). We set the desired coverage level $k = 3$, so that there will be 300, 600, 900, 1200, 1500, 1800, and 2100 sensors needed to be dispatched to $\mathcal{I}$. The moving speed of each sensor is set to one meter per second. The moving energy cost e_i^{move} of a sensor i is randomly selected from [0.8 J (joule), 1.2 J] per meter. For our sensor dispatch schemes, the two timers $T_{\text{update\_OCC}}$ and $T_{\text{broadcast}}$ are set to five seconds. In the pattern-based dispatch scheme, we randomly select 10%, 20%, and 30% target locations inside $\mathcal{I}$ as the seed locations. For comparison purpose, we design a *greedy* dispatch scheme, where sensors are assumed to know all target locations inside $\mathcal{I}$ and they will simply move toward their closest locations without exchanging any information with other sensors. In this case, a sensor can realize that its destination has been occupied by sufficient sensors only when the sensor moves close to its destination (i.e., no larger than the communication distance r_c). For baseline reference, we also design a *centralized* dispatch scheme. In this scheme, we assume that the sink knows the positions and statuses of all sensors and thus can calculate an optimal dispatch.

Fig. 8 shows the total moving energy and average moving distance of sensors under the greedy, competition-based, and centralized dispatch schemes. As can be seen, when the number of sensors increases, the average moving distances of the greedy and competition-based schemes also increase. This is because each sensor has to compete with more other sensors and thus increases its moving distance. Nevertheless, the greedy scheme will lead sensors to move much longer distances (and thus consumes more energy) compared with the competition-based scheme. This is because sensors just blindly move toward their nearest locations without exchanging necessary information to avoid moving to the same locations. In the hollow and right scenarios, the situation becomes worse as the number of sensors increases since the number of unnecessary contests also increases in the greedy scheme. In the central scenario, the average moving distance of the greedy scheme always keeps very high (as compared with the other two dispatch schemes) because sensors are initially concentrated in a small region. Thus, from Fig. 8, we can observe that simply taking a greedy strategy to dispatch sensors will make them exhaust much energy, thereby greatly shortening the network lifetime. On the other hand, by properly exchanging and maintaining necessary information of sensors, our competition-based scheme can consume slightly more energy compared with the centralized scheme (especially in the hollow and right scenarios). Note that in the central scenario, since sensors have similar initial positions, there will be more sensors that compete for the same destinations. Thus, the competition-based scheme will cause sensors to move longer distances compared with the centralized scheme.

Fig. 9 illustrates the total moving energy and average moving distance of sensors under the pattern-based and competition-based dispatch schemes. The competition-based scheme outperforms the pattern-based scheme because sensors have the full knowledge of target locations. In the pattern-based scheme, the average moving distance will arise as the number of sensors increases. This is because sensors have to compete for those few known locations in the beginning, thus increasing their moving distances. However, the average moving distance (and total moving energy) of the pattern-based scheme can decrease when there are more target locations selected as seeds.

C. Effect of Seed Locations on the Pattern-based Dispatch Scheme

The third experiment evaluates the effect of seed locations on the average moving energy of sensors in the pattern-based dispatch scheme. In this experiment, we set the number of sensors as 600 and 1500, and randomly select 5% to 70% target locations inside $\mathcal{I}$ as the seed locations.

Fig. 10 shows the effect of seed locations. As can be seen, the average moving energy of sensors can be reduced when the number of seed locations increases. When the percentage of seed locations arrives at 100%, the pattern-based scheme will work the same as the competition-based scheme. From Fig. 10(a),

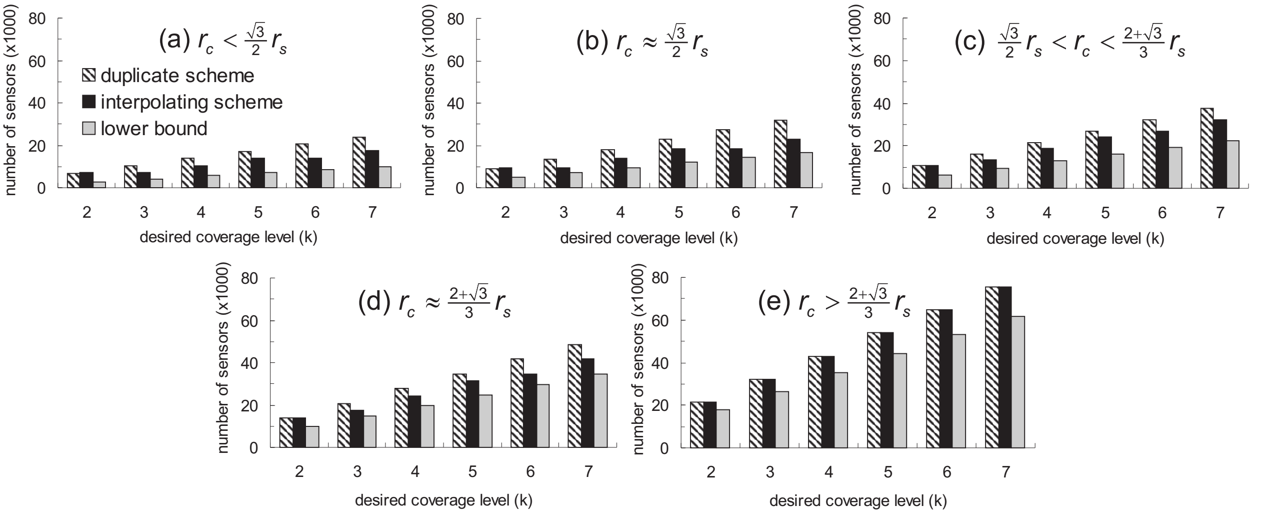
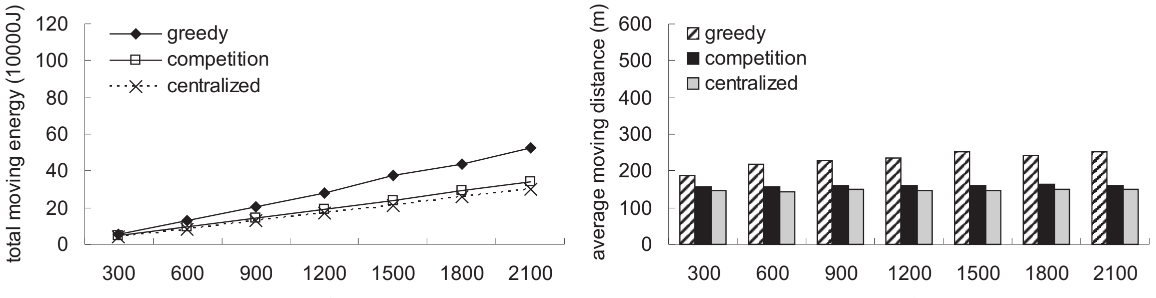
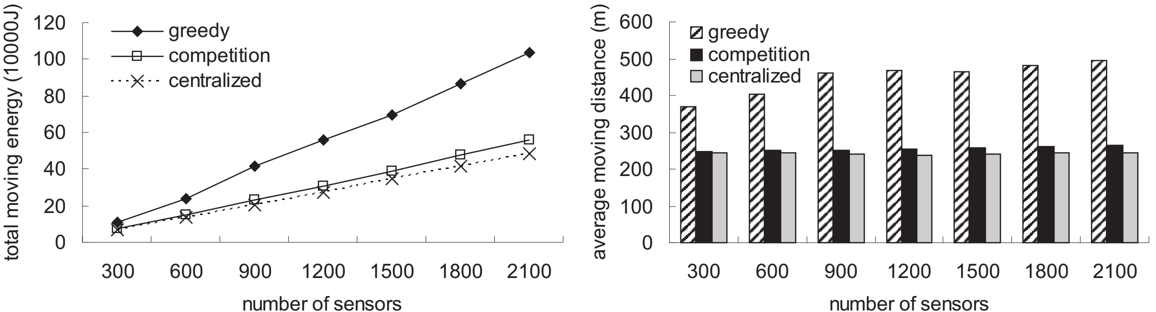


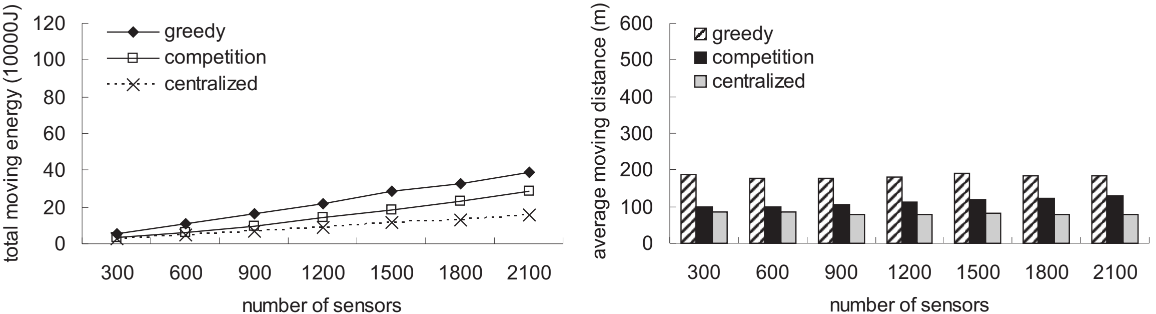
Fig. 7. Comparison on numbers of sensors required under different coverage level k.



(a) hollow scenario



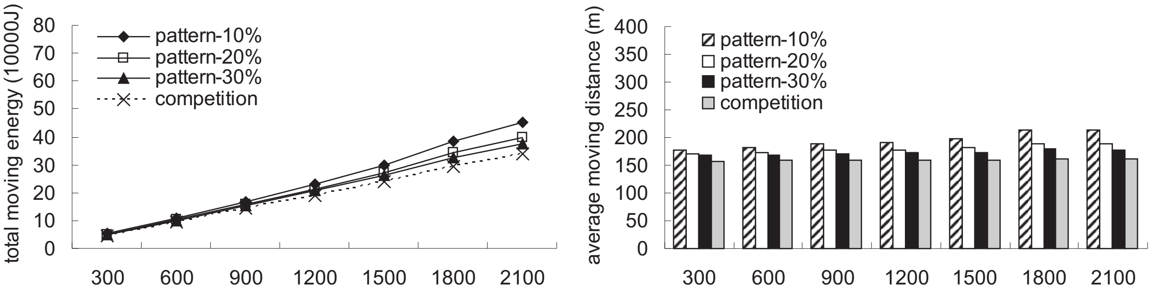
(b) right scenario



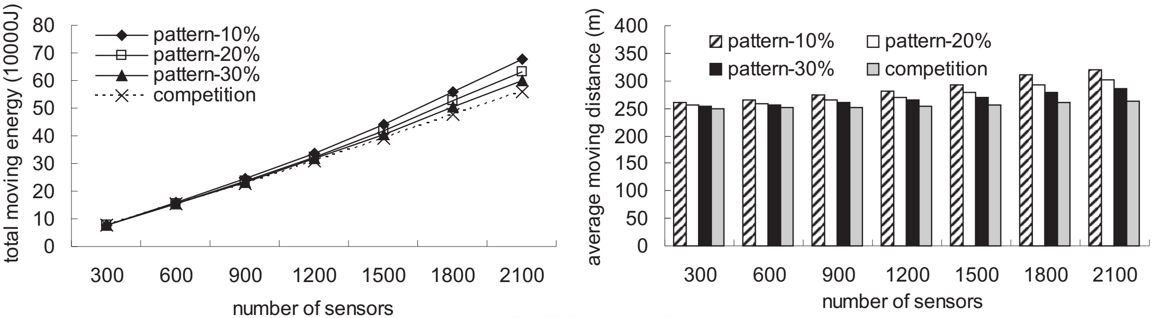
(c) central scenario

Fig. 8. Comparison on total moving energy and average moving distance of sensors under the greedy, competition-based, and centralized dispatch schemes.

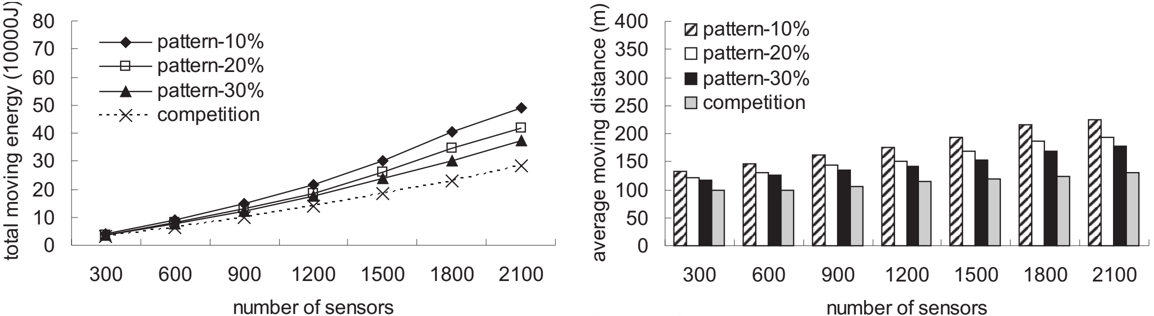
we can observe that in the hollow scenario, the difference between average moving energies of the pattern-based scheme and competition-based scheme can be smaller than 10 J when there are more than 40% $\sim$ 45% target locations selected as seeds. In the right scenario (i.e., Fig. 10(b)), when the number of sensors is 600 (respectively, 1500), such difference can be achieved if there are more than 15% $\sim$ 20% (respectively, 35% $\sim$ 40%) target locations selected as seeds. On the other hand, in the central scenario (i.e.,



(a) hollow scenario



(b) right scenario



(c) central scenario

Fig. 9. Comparison on total moving energy and average moving distance of sensors under the pattern-based and competition-based dispatch schemes.

Fig. 10(c)), such difference can be achieved if we select at least 45% ~ 50% seed locations. To summarize, from Fig. 10, we can observe that the performance of the pattern-based scheme can be significantly improved by selecting 40% ~ 50% target locations as seeds.

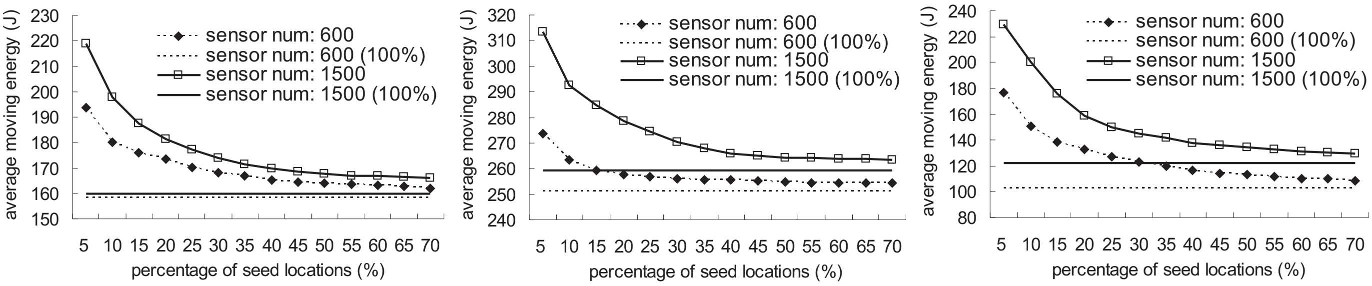
VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed systematical solutions to the k-coverage sensor placement problem and distributed sensor dispatch problem. Our placement solutions allow an arbitrary relationship of sensors' communication distance and their sensing distance, and can work properly under both binary and probabilistic sensing models. It is verified that the interpolating placement scheme requires fewer sensors to ensure k-coverage of the sensing field and connectivity of the network as compared with the duplicate placement scheme. Our dispatch solutions are based on the competitive nature of a distributed network. Simulation results have shown that the competition-based dispatch scheme performs better than the greedy and pattern-based dispatch schemes. However, by selecting sufficient seed locations, the pattern-based scheme can work as efficient as the competition-based scheme.

As to future work, sensor deployment in arbitrary-shaped regions for multi-level coverage deserves further study. When the area of interest $\mathcal{I}$ is of an arbitrary shape, one potential approach is to form a rectangle region that can fully cover $\mathcal{I}$. Then we can apply our solution to this rectangle and then remove those sensors that are outside the area of interest. Another way is to approximate $\mathcal{I}$ by multiple smaller rectangles. In our model, sensors' energy drain is at a constant speed when moving around. More sophisticated energy consumption models of mobile sensors can be defined and this deserves further investigation. For example, a startup energy cost may be incurred when first moving a sensor, and a cost may be incurred to enforce a sensor to turn around. In addition, variable moving speeds can be considered too.

ACKNOWLEDGMENT

Y. C. Tseng's research is co-sponsored by Taiwan MoE ATU Program, by NSC grants 93-2752-E-007-001-PAE, 96-2623-7-009-002-ET, 95-2221-E-009-058-MY3, 95-2221-E-009-060-MY3, 95-2219-E-009-007, 95-2218-E-009-209, and 94-2219-E-007-009, by Realtek Semiconductor Corp., by MOEA under grant number 94-EC-17-A-04-S1-044, by ITRI, Taiwan, by Microsoft Corp., and by Intel Corp.



(a) hollow scenario (b) right scenario (c) central scenario

Fig. 10. Effect of seed locations on the average moving energy of sensors in the pattern-based dispatch scheme.

REFERENCES

[1] D. Nicules and B. Nath, “Ad-hoc positioning system (APS) using AoA,” *Proc. IEEE INFOCOM*, pp. 1734–1743, 2003.

[2] L. A. Klein, “A boolean algebra approach to multiple sensor voting fusion,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 317–327, Apr. 1993.

[3] T. Sun, L. J. Chen, C. C. Han, and M. Gerla, “Reliable sensor networks for planet exploration,” *Proc. IEEE Int’l Conf. Networking, Sensing and Control*, pp. 816–821, 2005.

[4] D. Tian and N. D. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks,” *Proc. ACM First Int’l Workshop on Wireless Sensor Networks and Applications (WSNA ’02)*, pp. 32–41, 2002.

[5] Z. Abrams, A. Goel, and S. Plotkin, “Set k-cover algorithms for energy efficient monitoring in wireless sensor networks,” *Proc. IEEE Third Int’l Symp. Information Processing in Sensor Networks (IPSN ’04)*, pp. 424–432, 2004.

[6] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” *Proc. IEEE INFOCOM*, pp. 1976–1984, 2005.

[7] T. C. Shermer, “Recent results in art galleries,” *Proc. IEEE*, vol. 80, no. 9, pp. 1384–1399, Sep. 1992.

[8] M. D. Adickes, R. E. Billo, B. A. Norman, S. Banerjee, B. O. Nnaji, and J. Rajgopal, “Optimization of indoor wireless communication network layouts,” *IIE Trans.*, vol. 34, no. 9, pp. 823–836, Sep. 2002.

[9] K. S. Tang, K. F. Man, and K. T. Ko, “Wireless LAN design using hierarchical genetic algorithm,” *Proc. Seventh Int’l Conf. Genetic Algorithm*, pp. 629–635, 1997.

[10] L. Raisanen, R. M. Whitaker, and S. Hurley, “A comparison of randomized and evolutionary approaches for optimizing base station site selection,” *Proc. ACM Symp. Applied Computing (SAC ’04)*, pp. 1159–1165, 2004.

[11] H. R. Anderson and J. P. McGeehan, “Optimizing microcell base station locations using simulated annealing techniques,” *Proc. IEEE 44th Vehicular Technology Conf. (VTC ’94)*, pp. 858–862, 1994.

[12] S. S. Dhillon and K. Chakrabarty, “Sensor placement for effective coverage and surveillance in distributed sensor networks,” *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC ’03)*, pp. 1609–1614, 2003.

[13] F. Y. S. Lin and P. L. Chiu, “A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks,” *IEEE Comm. Letters*, vol. 9, no. 1, pp. 43–45, Jan. 2005.

[14] K. Kar and S. Banerjee, “Node placement for connected coverage in sensor networks,” *Proc. Int’l Symp. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt ’03)*, 2003.

[15] Y. C. Wang, C. C. Hu, and Y. C. Tseng, “Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks,” *Proc. IEEE First Int’l Conf. Wireless Internet (WICON ’05)*, pp. 114–121, 2005.

[16] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE Trans. Computers*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.

[17] C. F. Huang and Y. C. Tseng, “The coverage problem in a wireless sensor network,” *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, Aug. 2005.

[18] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, “PEAS: a robust energy conserving protocol for long-lived sensor networks,” *Proc. IEEE 23rd Int’l Conf. Distributed Computing Systems (ICDCS ’03)*, pp. 28–37, 2003.

[19] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” *Proc. ACM First Int’l Conf. Embedded Networked Sensor Systems (SenSys ’03)*, pp. 28–39, 2003.

[20] Z. Zhou, S. Das, and H. Gupta, “Connected k-coverage problem in sensor networks,” *Proc. IEEE 13th Int’l Conf. on Computer Commu. and Networks (ICCCN ’04)*, pp. 373–378, 2004.

[21] S. Yang, F. Dai, M. Cardei, and J. Wu, “On multiple point coverage in wireless sensor networks,” *Proc. IEEE Int’l Conf. Mobile Adhoc and Sensor Systems*, 2005.

[22] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, “Ensuring area k-coverage in wireless sensor networks with realistic physical layers,” *Proc. IEEE Fifth Conf. Sensors*, pp. 880–883, 2006.

[23] J. P. Sheu and H. F. Lin, “Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks,” *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC ’07)*, pp. 2631–2636, 2007.

[24] P. Basu and J. Redi, “Movement control algorithms for realization of fault-tolerant ad hoc robot networks,” *IEEE Network*, vol. 18, no. 4, pp. 36–44, Jul.-Aug. 2004.

[25] Z. Butler and D. Rus, “Event-based motion control for mobile-sensor networks,” *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 34–42, Oct.-Dec. 2003.

[26] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” *Proc. IEEE INFOCOM*, pp. 1293–1303, 2003.

[27] G. Wang, G. Cao, and T. L. Porta, “Movement-assisted sensor deployment,” *Proc. IEEE INFOCOM*, pp. 2469–2479, 2004.

[28] N. Heo and P. K. Varshney, “Energy-efficient deployment of intelligent mobile sensor networks,” *IEEE Trans. Systems, Man and Cybernetics, Part A*, vol. 35, no. 1, pp. 78–92, Jan. 2005.

[29] K. Akkaya and M. Younis, “C^2AP: coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks,” *Proc. IEEE Int’l Performance, Computing, and Comm. Conf. (IPCCC ’07)*, pp. 281–288, 2007.

[30] G. Wang, G. Cao, T. L. Porta, and W. Zhang, “Sensor relocation in mobile sensor networks,” *Proc. IEEE INFOCOM*, pp. 2302–2312, 2005.

[31] X. Du and F. Lin, “Improving sensor network performance by deploying mobile sensors,” *Proc. IEEE 24th Int’l Performance, Computing, and Comm. Conf. (IPCCC ’05)*, pp. 67–71, 2005.

[32] T. A. Dahlberg, A. Nasipuri, and C. Taylor, “Explorebots: a mobile network experimentation testbed,” *Proc. ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND ’05)*, pp. 76–81, 2005.

[33] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, “Mobile Emulab: a robotic wireless and sensor network testbed,” *Proc. IEEE INFOCOM*, 2006.

[34] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y. Y. Hsieh, “iMouse: an integrated mobile surveillance and wireless sensor system,” *IEEE Computer*, vol. 40, no. 6, pp. 60–66, Jun. 2007.

[35] Y. Zou and K. Chakrabarty, “A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks,” *IEEE Trans. Computers*, vol. 54, pp. 978–991, Aug. 2005.

[36] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*. 4th ed., Springer Verlag, 1997.

[37] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *IEEE Personal Commu.*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[38] L. Hu and D. Evans, “Localization for mobile sensor networks,” *Proc. ACM MobiCom*, pp. 45–57, 2004.

[39] R. Rao and G. Kesidis, "Purposeful mobility for relaying and surveillance in mobile ad hoc sensor networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 3, pp. 225–231, Jul.-Aug. 2004.

[40] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," *Proc. ACM 7th Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pp. 131–142, 2006.

[41] R. Kershner, "The number of circles covering a set," *American J. Math.*, vol. 61, pp. 665–671, Jul. 1939.

[42] "IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks specific requirements part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)," 2003.

You-Chiun Wang received his BEng and MEng degrees in Computer Science and Information Engineering from the National Chung-Cheng University and the National Chiao-Tung University, Taiwan, in 2001 and 2003, respectively. He obtained his Ph.D. degree in Computer Science from the National Chiao-Tung University, Taiwan, in October of 2006. Currently, he is a postdoctoral research associate at the Department of Computer Science, National Chiao-Tung University, Taiwan. His research interests include wireless network and mobile computing, communication protocols, and wireless sensor networks.

Yu-Chee Tseng obtained his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. He is Professor (2000–preset) and Chairman (2005–present) at the Department of Computer Science, National Chiao-Tung University, Taiwan. From 2006 to present, he serves as Adjunct Chair Professor at the Chung Yuan Christian University. Dr. Tseng received the Outstanding Research Award, by National Science Council, ROC, in both 2001–2002 and 2003–2005, the Best Paper Award, by Int'l Conf. on Parallel Processing, in 2003, the Elite I. T. Award in 2004, and the Distinguished Alumnus Award, by the Ohio State University, in 2005. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Dr. Tseng served as an Associate Editor for Telecommunication Systems (2005–present), as an Associate Editor for IEEE Trans. Vehicular Technology (2005–present), and as an Associate Editor for IEEE Trans. Mobile Computing (2006–present).