19th International Computer Science and Engineering Conference (ICSEC)
Chiang Mai, Thailand, 23-26 November, 2015

Remote Air Conditioning Control System Based on ZigBee Wireless Sensor Network for Building

Wibhada Naruephiphat
NECTEC
National Science and Technology
Development Agency (NSTDA)
Bangkok, Thailand
wibhada.naruephiphat@nectec.or.th

Ridnarong Promya
NECTEC
National Science and Technology
Development Agency (NSTDA)
Bangkok, Thailand
ridnarong.promya@nectec.or.th

Aimaschana Niruntasukrat*
NECTEC
National Science and Technology
Development Agency (NSTDA)
Bangkok, Thailand
aimaschana.nir@nectec.or.th

***Abstract*—We design and develop an intelligent remote air conditioning control system for building based on a Wireless Sensor/Actuator Network (WSN) that automatically adjusts to the comfort level of the room occupants. The system connectivity is based on ZigBee technology to do away with the conventional wiring and intrusive installation. The architecture is designed with the scalability and flexibility in mind by employing message oriented middleware. Consequently, all the components are decoupled from each other such that it is easy for any future adjustment, whether it is to replace, to remove, or to add components. Finally, we set up a testbed and the experiment results show that the system responds to the user's commands with little to almost no delay and with satisfactory performance.**

Keywords—Wireless Sensor network; ZigBee; air conditioning; Heat Index, asynchronous messages

I. INTRODUCTION

In 1991, Mark Weiser published a seminal article [1], coining the term "Ubiquitous Computing", and described it as the age when technology faded into the background of our environment. He envisioned the future in which technology seamlessly blends into everything in our lives and becomes invisible, but at the same time it transforms our living to be smoother and more efficient. He also pointed out three things that would help realize this vision: 1) cheap and low-power displays, 2) application software, and 3) networking.

Two decades later, his idea of "tomorrow's world" has not only taken shape, but made a profound impact across various fields. For a few examples, nowadays wireless networks play a prominent role in data communications, having rapidly evolved from those days they were first invented and used purely for voice communication. Wireless technologies such as Wireless Local Area Network (WLAN) and Wireless Personal Area Network (WPAN), both standardized in the 1990s, are less invasive than their wired counterparts, hence making the machines they connect better immersed into the environment. The next decade sees the emergence of ever smaller, cheaper, and more powerful single-board computers such as Raspberry Pi, Beaglebone, Arduino that can add intelligence and connectivity to any everyday devices and appliances. We see significant advances in sensor technology, an enabler of context awareness in machines and applications. The size of sensors has become smaller, the cost cheaper, and the performance more powerful [2]. This, along with the development of robust routing paradigms such as Mobile Ad Hoc Network (MANET), Delay Tolerant Network (DTN), and Mesh Network, helps drive the proliferation of Wireless Sensor Network (WSN) which is commonly found these days applied in a variety of fields, e.g. healthcare [3], military [4], agriculture [5],[6], and industry [7].

With everything in place – abundance of smart, energy-efficient devices, scores of connectivity options, several effective software platforms for applications, the Ubiquitous Computing age seems to be within reach. However, one of the key jigsaws missing here that will help complete the picture is automation. Without it, we are left with impossibly enormous task of manual configuring, decision making, system managing, etc.

Commercially and academically, most proposed applications of automation fall into two categories – residential (or home) automation and building automation. For both categories, we find the majority of work focusing on two intertwining areas: energy saving and climate sensing/control.

In home automation, interoperability seems to be the common theme as the lack of standard continues to hinder its full adoption. Mineno et al. [8] presents an energy management system for home or building by employing overlay sensor networks over two media, Power-Line Communication (PLC) and ZigBee. The main goal is to improve the system coverage. Another home automation for device control is proposed in [9]. A home gateway is installed to provide interoperability between heterogeneous networks (WiFi and ZigBee) that coexist.

For building automation, Kintner-Meyer et al. [10] place temperature sensors throughout an office building as well as additional sensors on a rooftop HVAC unit to monitor its performance. Shah et al. [11] presents a power management system consisting of a central controller and several end nodes, each of which connects to a device in the building. The controller sets the power threshold. The end nodes monitor the power being consumed by their connecting devices and cut off the supply whenever it exceeds the threshold. Another recent work in the area can be found in [12] by Peng and Qian. They propose an energy monitoring and control system based on

978-1-4673-7825-3/15/$31.00 ©2015 IEEE

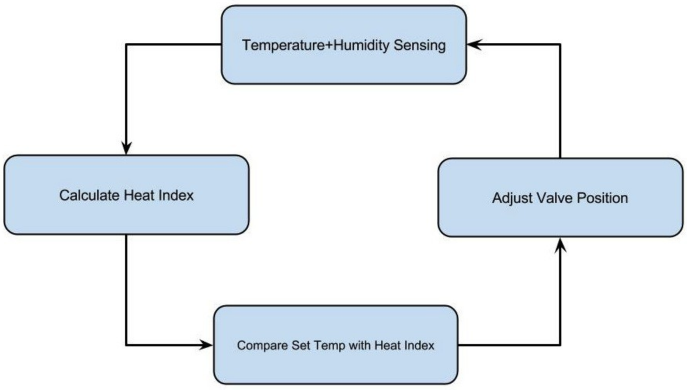


Fig. 1. System control loop.

ZigBee. The current and voltage are periodically measured by a measuring circuit and sent to a microcontroller to check for abnormality. The information is passed on to a building server where a database is maintained. In case of emergency, the user can send command to cut off the power supply through a GUI software program.

Although, there are a number of articles discussing the use of WSN for building monitoring and control. Rarely have we found scalability mentioned or raised as a design criteria. Considering that sensor nodes usually are limited in resource, the burden of carrying out most functions is shifted to sink node and central server. One server may suffice for now, serving only one application, whether it is temperature control or light control or energy consumption management. However, in the future when we combine these applications together to build a complete building automation, especially in large-scale building, we might have to deal with a much larger number of sensors of diverse capabilities. Therefore, in [13] we had designed a scalable service platform for sensor network applications, aiming specifically to apply in building automation. It can serve a large volume of data.

In this paper, we build on top of our service platform in [13] a remote control system for Air Conditioning (AC) in building. In Thailand where the weather is hot and humid all year round, AC accounts for the vast majority of the building energy consumption [14]. Therefore, it is no surprise that when it comes to energy saving guidelines, AC management/maintenance is usually on the top of the list. General suggestion is to set the temperature at 25ºC [15]. Our own office has adopted this as policy. Unfortunately, during the time of day that the humidity is relatively high, which could be extremely high during summer in Thailand, it does not feel like 25ºC but much higher. Without knowing better, we basically trade off our comfort for cost saving. Even worse when the condition becomes intolerable and people start turning on supplement AC units in their rooms, the cause is practically lost. In this paper, we seek a compromising solution by designing a remote AC control system that automatically adjusts to the comfort level of the room occupants while keeping the felt temperature at the setting point. After all, in his last paper [16], Weiser concluded that the "calm" state of mind of the user is the ultimate goal of Ubiquitous Computing, and we cannot achieve that without the user's comfort.

Employing ZigBee-based WSN, the system constantly measures the ambient temperature and humidity. Then, it calculates Heat Index (HI) (see [17] for the formulation), which is a quantity that combined the effects of the temperature and the humidity together and makes a better indication of our bodies' perception to the weather. Finally, the system adjusts the chilled water tube's valve position in the AC to maintain HI at the set point. The architecture of the system is highly scalable and flexible as it is easy to add, remove, or make change to any component without affecting other components.

The outline of this paper is as follows. Section II proposes the remote AC control system. Section III describes our experimental setup. Section IV presents experimental results to verify system's minimal latency and its effectiveness, and Section V concludes the paper.

II. PROPOSED REMOTE AIR CONDITIONING CONTROL SYSTEM

Our system consists of a network of wireless sensor and actuator nodes which are installed throughout the building in order to sense humidity and ambient temperature and to control the air conditioners. The goal is to achieve the desired condition for users/occupants through two modes of operations: manual mode and automatic mode.

In the manual mode, user can set the preferred temperature in a room via our web application which subsequently sends the command to the actuator controlling the air conditioning in that room. The actuator then adjusts the valve position that allows the flow of chilled water into the room. In the automatic mode, the system persistently maintains the desired condition through the feedback loop depicted in Fig. 1. The sensors in each room periodically measure ambient temperature and humidity and sent this data to the room actuator. Average values of temperature and humidity gathered from all the sensor nodes in the room are used to calculate Heat Index. If the Heat Index differs from the set temperature, the actuator will adjust the chilled water pipe's valve position accordingly.

In addition, our web application also provides real-time monitoring report for authorized users and/or administrator. Since the system incurs only small traffic, its communication can be implemented by sharing the same resource on the existing network infrastructure, i.e., LAN, WiFi.

A. System Architecture

As shown in Fig. 2, the remote AC control system consists of the following components:

1. Wireless Sensor Network (WSN) of sensor (S) and actuator (A) nodes, each of which is equipped with a ZigBee communication chip. The sensor nodes have temperature and humidity sensing capabilities. Every sensor node periodically sends its data to a ZigBee coordinator node which forms the network and acts as a gateway of the WSN.

2. ZigBee gateway. We employ a Raspberry Pi, a single-board microcomputer, as a ZigBee gateway. The ZigBee gateway is a bridge between the WSN and the server, performing protocol translation between ZigBee used in the WSN and Advanced Message Queuing Protocol (AMQP) used in our server.

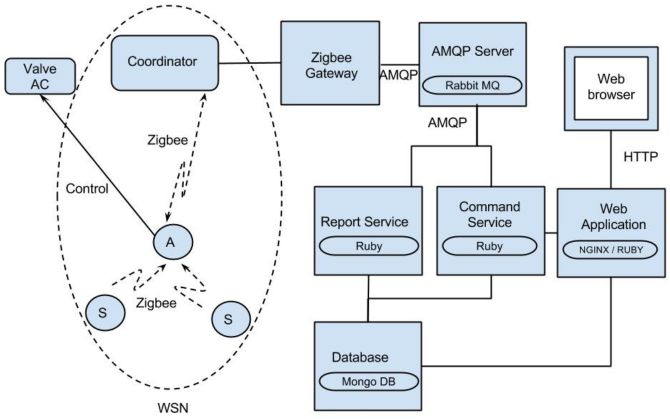


Fig. 2. Remote AC control system architecture

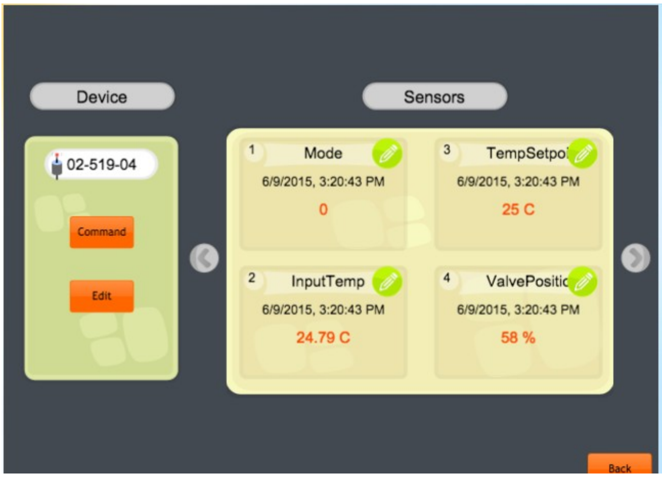
All gathered sensing data from the WSN will be sent to the server via the ZigBee gateway.

3. AMQP messaging server. We choose the Advanced Message Queuing Protocol (AMQP) for communication among all service modules in order to decouple components from each other. As a result, we can easily control or change flow of control in software component. In particular, we use RabbitMQ to implement AMQP. RabbitMQ is an open source software that provides rich and high performance AMQP server. It also has several useful features such as acknowledgment, message exchange, message routing. The server reliably routes messages and puts them into their respective queues.

4. Command service is the consumer of the responses sent from the nodes in the WSN and the ZigBee gateway. Implemented with Ruby programing language, the command service performs two main functions. First, it sends push notification to the web application so that user can view the data in real time. Second, it waits for the response and pairs it with its corresponding command before writing them in the database because of the asynchronous flow of messaging communication.

5. Report service. When the ZigBee gateway receives incoming data from the sensor nodes, it calls report service to preprocess and store sensor data to the database. Additionally, the report service is responsible for sending push notification to the web application. Also implemented in Ruby, this service is designed to work in a “fire and forget” manner, therefore no response is needed after the service has been called by the ZigBee gateway.

6. Web Application. We develop Graphical User Interface (GUI) on web browser for general users to be able to use our system via HTTP protocol. It also provides application program interface (API) for third party applications to retrieve the data in JSON format. The GUI itself actually pulls data from API and renders it by using Javascript. Through our web application, the user can manage sensors, retrieve sensor data and send control command to actuators.



02-519-04

1 Mode
6/9/2015, 3:20:43 PM
0

3 TempSetpoi
6/9/2015, 3:20:43 PM
25 C

2 InputTemp
6/9/2015, 3:20:43 PM
24.79 C

4 ValvePositic
6/9/2015, 3:20:43 PM
58 %

Fig. 3. Graphical user interface (GUI)

7. Database. We store all collected data, commands, and responses in the NoSQL database namely MongoDB for its easy scalability such that past information can be retrieved for further analyzation.

B. *Graphical User Interface*

For general users, our web application provides arrays of basic functions for system management ranging from managing user account, finding device, monitoring room environment, plotting graph to controlling the temperature. For an example, the web application retrieves sensor data from the database and then renders to the web browser via GUI shown in Fig. 3. Username and password are used for authorization. The application supports both Mac OS and Windows OS and runs on all major modern browsers including Internet Explorer, Chrome, Mozilla Firefox and Safari.

C. *Command Process*

When the user sends a command to the system through the GUI, the software management converts command from HTTP protocol to AMQP protocol. However, how it goes from this point forward depends on the type of the command which we categorize by its intended receiver into two types.

1) Local Command. This type of command is originated from the user and intended for a ZigBee gateway (unicast) or more (multicast). Local commands generally deal with ZigBee network status and date/time. The activity diagram in Fig. 4 depicts the command flow of the local command.

First, the client, in this case the web application, initiates a command which is sent to the message queue in the AMQP server. The command is routed to the destination, in this case a ZigBee gateway. Both ends communicate by calling the command service running on both sides. On the receiving end, the command process handles receiving and processing of user command. If the command is correct, it is then executed and a response is generated and sent back to the client.2) Remote Command. This type of command is originated from the user and intended for a particular node in the WSN (unicast) or more (multicast). Remote commands involves controlling the

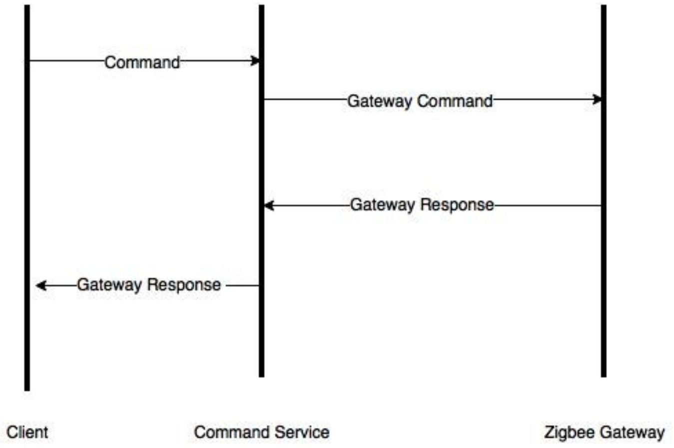


Fig. 4. Local command process diagram

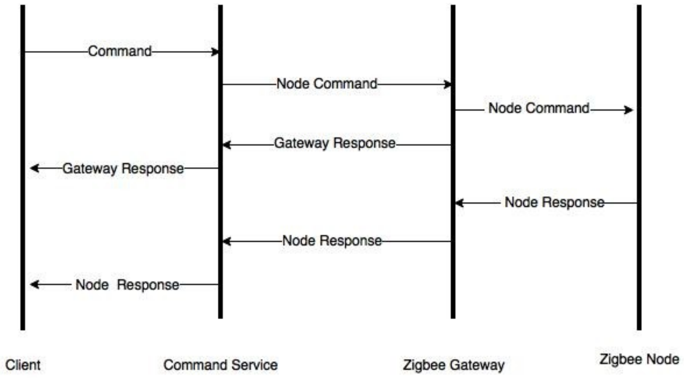


Fig. 5. Remote command process diagram

nodes and requesting for node status report. The remote command flow is shown in Fig . 5.

When the client, in this case the web application, initiates a remote command, the command is routed to the ZigBee gateway(s) connecting to the intended node(s). Like in the local command flow, the entity responsible for processing the remote command is the command process running on the ZigBee gateway. If the command is correct, the process extracts the node address and the command payload, then generates a ZigBee command. The ZigBee gateway sends the ZigBee command to the serial port destined for the intended ZigBee node(s). After the command is successfully sent, the gateway replies to the client with a default gateway response.

Because of the asynchronous flow of messaging communication, we assign a transaction number to every remote command in order for us to be able to keep track of what has been successfully executed and what has not. After the ZigBee node receives the command and finishes processing it, it sends a response along with the transaction number back to the ZigBee gateway which forwards it to the client. The client can identify by the transaction number and the timestamp the originating command of such a response.

III. EXPERIMENTAL SETUP

We set up a testbed in our 170 m^2 office room, of which the floor plan is shown in Fig. 6. Two sensor nodes (S1 and S2)

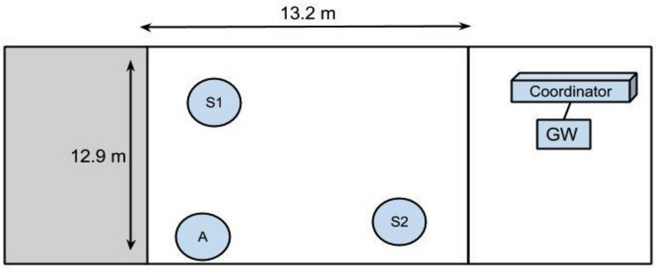


Fig. 6. Floor plan of the testbed location

and one actuator node (A) are installed in the room, while a coordinator node and a Zigbee gateway (GW) are in the adjacent room. three offsite servers are used, one for AMQP, one for database, and one for web.

Depicted in Fig. 7, the AC used in the experiment is a central system based on chilled water and heat exchanger. Chiller, which is responsible for generating chilled water, removes heat from the chilled water loop by air cooled fan. The chilled water is distributed through insulated tube. In our office building, each room has its own Air Handling Unit (AHU) with control valve that delivers chilled water to cool the air flowing into the room space. By adjusting the valve position, we can control the flow rate of chilled water delivered into the room. Therefore, we connect the actuator node to the valve so that the temperature in the room can be remotely controlled by this mean.

IV. EXPERIMENTAL RESULTS

We collect raw data consisting of measured temperature and relative humidity from the sensors every 10 minutes from 08.00 AM to 5:00 PM.

First, we test the system in automatic mode by setting the temperature constant at 25ºC and see how effective the system is in maintaining the HI at the set point.

In Fig. 8, the graph shows the comparison between the temperature set by the user (SetPoint), the actual temperature (InputTemp), the HI, as well as the outside temperature (OutdoorTemp). The result shows that our system is able to keep the HI in the room almost constant and very close to the set point, even though the outside temperature considerably

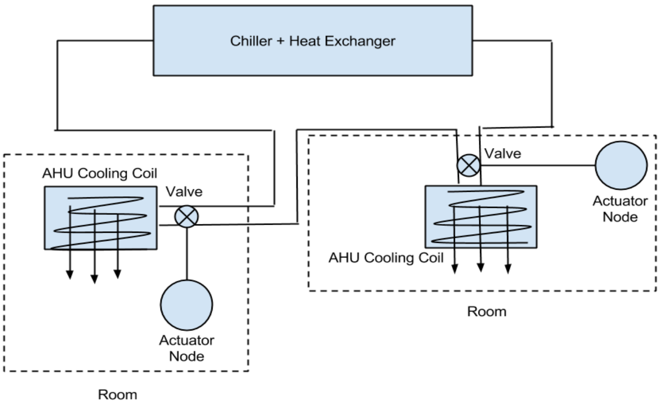


Fig. 7. Central air conditioning system connecting to actuator nodes

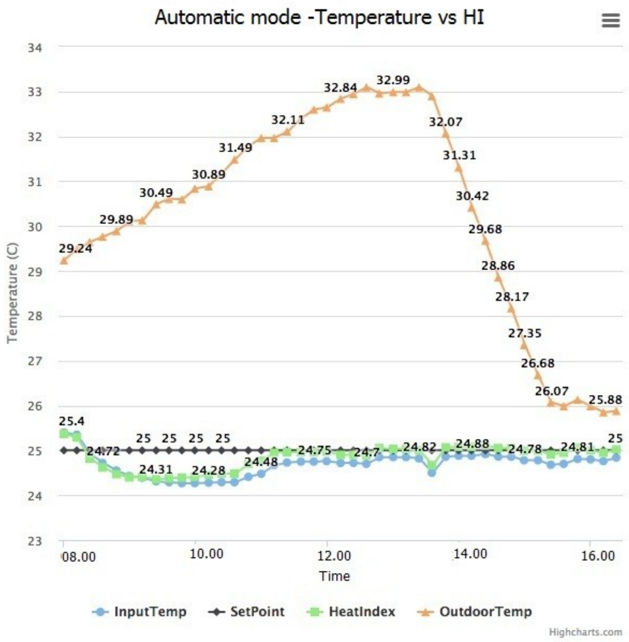


Fig. 8. Temperature control performance in automatic mode test.

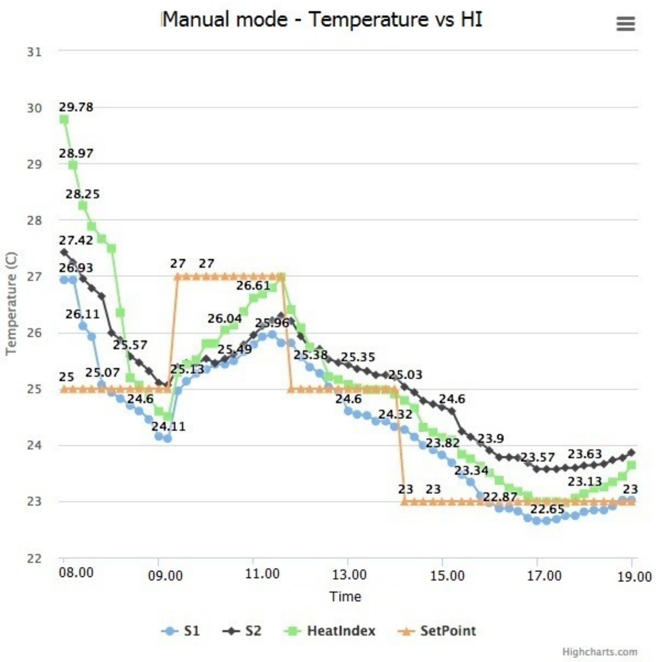


Fig. 9. Temperature control performance in manual mode test

changes during the day. The error ranges from 0% to 2.6% when we compare the accuracy between set point and HI. One factor that might contribute to the error is that the HI is calculated from the average values of data collected from two sensor nodes at two different locations. One is installed in the direct flow of the AC outlet while the other is at the point with little to no direct flow of cool air. Therefore, there is quite a bit of difference between the temperature values measured at these two points. Large range between sample points, coupled with nonlinear relationship between two sets of measurements, makes it difficult to achieve close to 0% accuracy at all time.

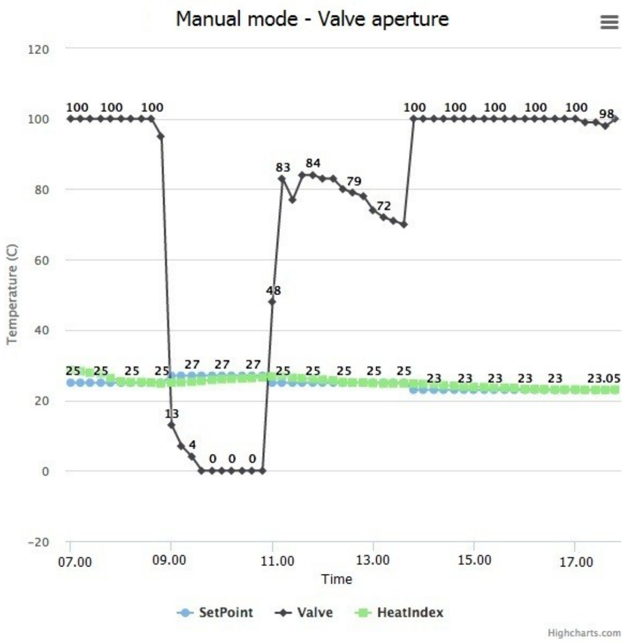


Fig. 10. Chilled water tube's valve aperture in manual mode test

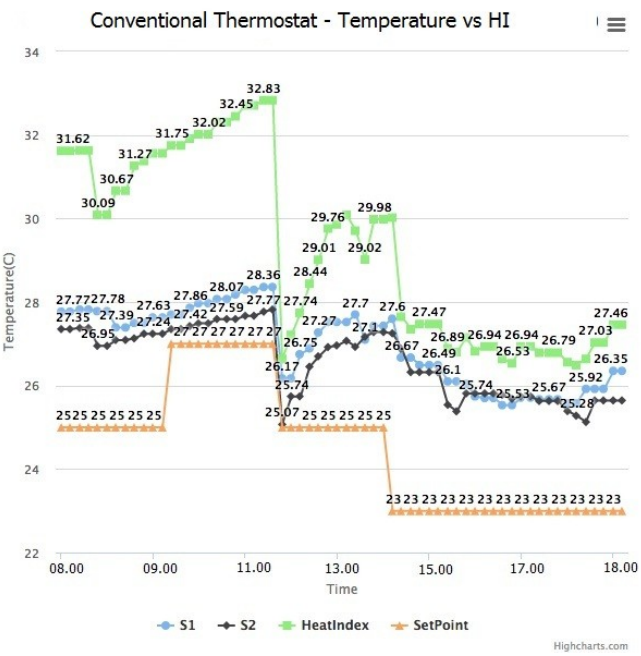


Fig. 11. Temperature control performance when using conventional thermostat

Furthermore, both nodes send data asynchronously to the controller. Thus, the average value is not exactly calculated from the data at the same instance. This also leads to some of the errors.

In the second experiment, we test the system in manual mode by having the user change the set temperature up and down during the day in order to see how well the system responds to the change. Fig. 9 shows the result comparing set temperature, HI, and two measurements from sensor nodes (S1 and S2). To better demonstrate the action behind the scenes, we also plot the chilled water tube's valve aperture in degrees over the same period in Fig. 10. Large aperture means high flow rate and vice versa. We can see that despite what seems to be a slow response in terms of temperature/HI (20-70 minutes/1ºC change), the valve control mechanism reacts immediately to the commands. However, since there is a limit to how much we can adjust the valve in either direction, it takes some time for the temperature/HI to catch up. Also, there are other obvious factors affecting the control performance: initial conditions (temperature, humidity), temperature in the chilled water tube, outdoor temperature, etc. However, despite various variables and large-area testbed, our system is still able to fulfill the objective. For comparison, we turn off the system and switch to use a conventional thermostat instead. The temperature is manually set in the same exact manner as how we did in the manual test. The results comparing set temperature, HI, and two measurements from sensor nodes (S1 and S2) are shown in Fig. 11. Evidently, using measured temperature as the only input data to adjust the AC results in the felt temperature (HI) being much higher than the set point. At some point the difference is almost 6ºC.

Finally, we measure the full-loop response time of the command process, starting from the instance the user issues the command until the web application receives the acknowledging response. The result in Table 1 shows that the system responds within 1 second for both local and remote processes. It goes to show that the loose-coupling, asynchronous architecture incurs little to no latency to the system performance.

TABLE I. TIME RESPONSE OF COMMAND PROCESS

Command Type	Response Time (ms)
Local – Get date/time	777.8
Local – Read network info	771.4
Remote – Set temperatures	964.8
Remote – Get set temperature	872.8

V. CONCLUSION

In this paper, we present the design and implementation of an intelligent remote air conditioning control system for building. Using ZigBee-based WSN technology, the system automatically controls the AC unit in order for the perceived temperature in the room to be as close to the set point as possible. The system is built on top of a sensor application platform we designed such that the system is easily scalable for adding more future applications. The experiment shows satisfactory results in terms of both performance and delay.

REFERENCES

[1] Wieser, Mark, The computer for the 21st century. Scientific american 265.3, 1991, pp. 94-104.

[2] Jian Lu, H. Okada, T. Itoh, T. Harada and R. Maeda, 'Toward the World Smallest Wireless Sensor Nodes With Ultralow Power Consumption', IEEE Sensors Journal, vol. 14, no. 6, pp. 2035-2041, 2014.

[3] Alemda, Hande and Cem Ersoy, Wireless sensor networks for healthcare: A survey. Computer Networks 54.15 (2010), 2010, pp. 2688-2710.

[4] Lee and S. Hyuk, 'Wireless sensor network design for tactical military applications: remote large-scale environments.', in Military Communications Conference, 2009. MILCOM 2009., 2009.

[5] Abbasi, A. Zafar, N. Islam and Z. Ahmed Shaikh, A review of wireless sensors and networks' applications in agriculture. Computer Standard & Interfaces 36.2, 2014, pp. 263-270.

[6] N. Watthanawisuth, A. Tuantranont and T. Kerdcharoen, '"Microclimate real-time monitoring based on ZigBee sensor network', in Sensor,2009 IEEE, 2009.

[7] J. Valverde, V. Rosello, G. Mujica, J. Portilla, A. Uriarte and T. Riesgo, 'Wireless sensor network for environmental monitoring: application in a coffee factory', International Journal of Distributed Sensor Networks 2012, 2012.

[8] Mineno, H., Kato, Y., Obata, K., Kuriyama, H., Abe, K., Ishikawa, N. and Mizuno, T., 'Adaptive home/building energy management system using heterogeneous sensor/actuator networks', in Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, 2010.

[9] K. Gill, S. Yang, F. Yao and X. Lu, 'A zigbee-based home automation system', IEEE Transactions on Consumer Electronics, vol. 55, no. 2, pp. 422-430, 2009.

[10] M. Kintner-Meyer, M.R. Brambley, T.A. Carlon and N.N. Bauman, 'Wireless sensors: technology and cost-savings for commercial buildings, Teaming for Efficiency', 2002, pp. 7.121–7.134.

[11] P. Shah, T. Shaikh, K. Ghan and S. Shilaskar, 'Power Management Using ZigBee Wireless Sensor Network', 2008 First International Conference on Emerging Trends in Engineering and Technology, 2008.

[12] C. Peng and K. Qian, 'Development and Application of a ZigBee-Based Building Energy Monitoring and Control System', The Scientific World Journal, vol. 2014, pp. 1-13, 2014.

[13] W. Naruephiphat, R. Prom-Ya and C. Charnsripinyo, 'A design of scalable service platform for sensor network applications', 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014.

[14] Region3.prd.go.th, 'การอนุรักษ์พลังงาน ประหยัดพลังงานในที่ทำงาน', 2015. [Online]. Available: http://region3.prd.go.th/energy/energysave_work.php. [Accessed: 30- Jun- 2015].

[15] Nus.edu.sg, 'Office of Environmental Sustainability (OES), National University of Singapore', 2015. [Online]. Available: http://www.nus.edu.sg/oes/about/policy_circular/aircon_25c.html. [Accessed: 30- Jun- 2015].

[16] Weiser, Mark, Rich Gold, and John Seely Brown. "The origins of ubiquitous computing research at PARC in the late1980s." IBM systems journal 38.4 (1999): 693696.

[17] Rothfusz, Lans P., and NWS Southern Region Headquarters. "The heat index equation (or, more than you ever wanted to know about heat index)." Fort Worth, Texas: National Oceanic and Atmospheric Administration, National Weather Service, Office of Meteorology (1990): 9023.