Energy Efficient HVAC System with Distributed Sensing and Control

Cheng Li\textsuperscript{1,5}, Zhenjiang Li\textsuperscript{1,5}, Mo Li\textsuperscript{1,5}, Forrest Meggers\textsuperscript{2,5},
Arno Schlueter\textsuperscript{3,5}, Lim Hock Beng\textsuperscript{4,5}

\textsuperscript{1}School of Computer Engineering, Nanyang Technological University
\textsuperscript{2}School of Architecture, Princeton University
\textsuperscript{3}Department of Architecture, ETH Zurich
\textsuperscript{4}IntelliSys Laboratory, Nanyang Technological University
\textsuperscript{5}Future Cities Laboratory, Singapore ETH Center, ETH Zurich

Abstract—This paper presents our implementation experience in building an energy efficient HVAC system for cooling and air conditioning. The system exercises the "low exergy" theory and leverages high temperature water (18°C) cooling for better energy efficiency. In order to achieve this, the system decomposes the cooling and dehumidification functionalities, and employs decentralized air control for on-demand dehumidification and ventilation. The system comprises two control modules, namely, radiant cooling module and distributed ventilation module, cooperating with each other to provide the HVAC control. Abundant sensors and embedded control devices are customized and instrumented, and we develop a wireless sensor network to support control data exchange among those devices. Our experimental evaluation demonstrates that the system achieves accurate control targets and promptly responds to environment dynamics. The wireless sensor network effectively supports the system needs with long system lifespan. Compared with traditional HVAC systems, our system is of much higher energy efficiency, as measured by the standard COP metric.

I. INTRODUCTION

Heating, ventilation, and air conditioning (HVAC) systems provide occupants thermal comfort (cooling or heating), air dryness (dehumidification), and good air quality (ventilation). Towards green buildings, energy efficiency of HVAC systems continuously draws people's attention. Until 2012, buildings solely contribute up to half of the world electricity consumption and 47% of the building energy is attributed to HVAC systems [28], while it was recently reported by U.S. Energy Information Administration (EIA) that at least 30% of the HVAC energy usage is wasted [27].

This paper presents our implementation experience of BubbleZERO, an energy efficient HVAC system, deployed in a tropical country, Singapore, for cooling and air conditioning. BubbleZERO exercises the "low exergy" theory in building designs and leverages high temperature water (18°C) cooling to reduce the temperature gradient in the heat circulation process, and thus achieves higher energy efficiency.

The novelties of BubbleZERO are two-folds. First, BubbleZERO features a distributed HVAC framework that separates the cooling and ventilation functionalities in two control modules. Control decomposition allows each individual module to operate with the lowest exergy and achieve the best energy efficiency in the heat exchange. The control management is, however, challenged in such a framework. For instance, due to the control decomposition, condensation occurs when the air temperature and humidity mismatch. The control management needs to ensure not only the proper execution of individual modules but also their collaboration. We develop and deploy abundant sensing and control devices and build a wireless network to address the challenges. Second, the unique nature of BubbleZERO and the application requirements ask for tailored techniques for efficient wireless communication. In particular, we differently treat AC and battery powered devices so as to enable efficient message exchange and prolong the network lifespan. We duty cycle the battery powered devices and control their transmission periods according to sensory data dynamics. By so doing, we reduce the data volume to be transmitted for saving energy of batter powered devices yet still meet control needs. In addition, we let the AC powered devices adapt their transmission schedules to alleviate channel contentions. In light of this, we reduce the packet loss and delay, and further improve the energy efficiency of battery powered devices.

We implement and comprehensively exercise a prototype system to evaluate the performance of BubbleZERO. Our experimental study demonstrates that the system accurately achieves control targets and promptly responds to environment dynamics. With the outdoor temperature of 28.9°C and dew point of 27.4°C, the system approaches the target condition, 25°C temperature and 18°C dew point, in 30 minutes and maintains on the equilibrium. The distributed control in BubbleZERO well accommodates the disturbance from human activities and reacts in low convergence delay. Compared with traditional HVAC systems, BubbleZERO increases the energy efficiency by up to 45.5%, measured by the standard COP metric. The wireless network effectively supports the control message exchange in BubbleZERO. Our proposed adaptive transmission schedule satisfies the control timeliness requirement and also enables low energy consumption for battery powered devices, ensuring a long system lifespan.

The rest of this paper is organized as follows. §II introduces the background of low exergy HVAC. The HVAC
logic and wireless networking in BubbleZERO are detailed in §III and §IV, respectively. In §V we evaluate BubbleZERO. §VI is the literature review and §VII concludes this work.

II. BACKGROUND

Low exergy HVAC. Modern HVAC systems are designed based on the first law of thermodynamics, i.e., the conservation of energy. To approach a desired/target indoor temperature, some amount of heat must be absorbed (supplied) for cooling (heating), while the same amount of heat has to be exhausted to (taken from) the environment. The second law of thermodynamics indicates that an isolated system spontaneously evolves towards thermodynamic equilibrium. Extra power thus has to be consumed to stimulate the thermodynamic cycle for heat movement, as Carnot and Kelvin have proven in [16]. This part of energy consumption is directly related to the temperature gradient in the Kelvin have proven in [16]. This part of energy consumption is directly related to the temperature gradient in the thermodynamic cycle, which is quantified by the concept of “exergy” in building designs [23]. The exergy, $$E_x$$, of a heat flux, $$Q$$, moved from a room at reference temperature, $$T_0$$, compared to its working temperature, $$T$$, is defined as $$E_x = Q(1 - T/T_0)$$. It has been shown that lower exergy leads to less energy consumption to move the same amount of heat [23]. The temperature gradient determines the exergy, i.e., a higher difference between $$T_0$$ and $$T$$ will lead to the dramatically increased energy consumption to accomplish the heat exchange.

To exercise a low exergy design, we develop the HVAC system using the 18°C water as the thermal media to cool the room. Benefiting from the high heat capacity of water, we can achieve as much cooling power as the low temperature air used in traditional HVAC systems, but with a much lower exergy because of its relatively higher working temperature. Due to the humid weather in the tropical area, we need strong dehumidification capacity in addition to prevent condensation, which in our case, however, requires below 10°C dry air supply and has to be decomposed from the cooling module. Our system thus comprises two modules of separate functionalities, namely radiation cooling module (cooling) and distributed ventilation module (dehumidification & ventilation), each of which with its own control logic. The decomposition allows each module to best reduce its own temperature gradient for the lowest exergy operation. Compared with traditional systems that use as low as 8°C air for both cooling and dehumidification, we can separately use 18°C water for radiation cooling and 8°C air for dehumidification.

The need for wireless sensors. However, the decomposition of HVAC functional modules requires distributed control decision making, which relies on extensive interactions and communications among sensing and control devices over the space. Connecting devices by wire is straightforward whereas undesirable as it would add tens or even hundreds of endpoints and miles of cables to the system. The cable deployment is constrained by existing indoor layout, which significantly challenges endpoints’ installment. Moreover, a wired deployment is relatively fixed. If additional hardware needs to be integrated for new control demands or existing devices need to be repaired, it will lead to high system upgrade and maintenance cost and complexity.

For those reasons, wireless networks serve as the feasible option. In addition, networks with IEEE 802.15.4 radios are more attractive than Wi-Fi radios or bluetooth. 802.15.4 network has a simpler protocol stack and devices only need low-end MCUs, e.g., MSP430, which reduces the development and implementation complexity. On the other hand, 802.15.4 network stack is completely open-source, which facilitates customizing drivers or interfaces to enable 802.15.4 based devices to communicate with commercial HVAC units adopted. Finally, the combination of low-power 802.15.4 radio and low-power MCU leads to lower overall power consumption of the system. The need for low-power energy consumption is apparent as HVAC systems require sustainable deployment.

Deploying an efficient 802.15.4 network for the distributed HVAC system is, however, non-trivial. First, available interfaces of 802.15.4 devices are limited usually such that devices may have no common interfaces with commercial HVAC units. The network design needs extensive hardware and driver development efforts for interfacing all networking endpoints. Second, the data rate of 802.15.4 radio is limited, i.e., with the 250 Kbps peak rate while the effective rate is much lower due to the MAC-layer overhead. It is non-trivial to achieve an effective message exchange for satisfying control requirements and promptly adapting.
panels mismatch, i.e., given condensation. Power while strictly satisfies the humidity constraint to avoid the room. Hence, this module provides sufficient cooling open the door or window, the outdoor humid air enters the panel surface in the cooling. Condensation happens if the preferred temperature through thermal radiation. The major design challenge is to avoid condensation on the ceiling panel and concrete buildings, including special plastic cover filled up with flowing air that wraps the containers to reduce heat exchange, double glazing that selectively allows most visible daylight but rejects most heat from infrared, and insulated high performance facades that have high thermal resistance. With the configuration made by building engineering experts, the laboratory has similar thermal attributes and heat exchange performance as normal buildings.

III. DISTRIBUTED HVAC SYSTEM OF BUBBLEZERO

A. BubbleZERO HVAC overview

Figure 2 depicts the architecture of BubbleZERO, which comprises radiant cooling (blue) and distributed ventilation (yellow and red) two modules. The indoor space is organized into four equal subspaces labelled from subspace-1 to subspace-4 for ease of performance evaluation in Section V.

Radiant cooling module: radiant cooling module includes one water tank, one chiller, and two metal ceiling panels. Each ceiling panel is associated with two pumps that supply cold water from the tank and cools down the room to a preferred temperature through thermal radiation. The major design challenge is to avoid condensation on the ceiling panel surface in the cooling. Condensation happens if the temperature $T$ and the humidity $H$ of the air beneath ceiling panels mismatch, i.e., given $H$, condensation happens when $T$ drops below a threshold. It can happen when people open the door or window, the outdoor humid air enters the room. Hence, this module provides sufficient cooling power while strictly satisfies the humidity constraint to avoid condensation.

Distributed ventilation module: the distributed ventilation module controls the indoor humidity to the target level configured by the user. In addition, it ensures a proper humidity threshold such that the radiant cooling module can provide sufficient cooling power without causing condensation. This module also controls the indoor air freshness based on the indoor $\text{CO}_2$ concentration.

Above two modules work together to achieve the HVAC control objective. A total number of 38 sensors of different types (wired and wireless, general and customized, AC and battery powered ones) are deployed to coordinate and control the two modules. In the rest of this section, we elaborate the detailed design and implementation of each module.

B. Radiant cooling module

1) Radiant cooling control logic: A chiller is installed to cool down the water in a tank and the cold water is supplied to ceiling panels via supply pumps. The temperature of the supplied water $T_{supp}$ is 18°C, which is calculated based on the cooling need of BubbleZERO. Two radiant panels are deployed on the ceiling and controlled separately. For the ease of description, we depict one ceiling panel and the associated hydraulic loop in Figure 3. Through the thermal radiation and heat exchange, panels absorb heat from the room to lower the indoor temperature.

Directly supplying cold water to ceiling panels may lead to condensation on the surface and water drops. Condensation occurs when the air temperature is below a threshold, called dew point. Given the air temperature $T$ and humidity $H$, the dew point ($T_{dew}$) is calculated by:

$$T_{dew}(T, H) = a \cdot \frac{\ln(H/100) + (b \cdot T)/(a + T)}{b - \ln(H/100) - (b \cdot T)/(a + T)},$$

where $a = 243.12$ and $b = 17.62$. The radiant cooling module avoids condensation by a feedback design as shown in Figure 3. With a recycle pipe bridging the supply pipe and the return pipe, the module can redirect certain warm water from the return pipe and mix it with the cold water supplied from the tank. By controlling the speeds of the two pumps, we can adjust the percentage of the cold and warm water in the final mixture and thus control its temperature. Figure 4(a) depicts the connections of those pipes.
In Figure 3, $T_{\text{mix}}$ and $F_{\text{mix}}$ represent the temperature and the flow rate of the mixed water, respectively. They are the control parameters that together determine the final cooling performance. A lower $T_{\text{mix}}$ and a larger $F_{\text{mix}}$ provide higher cooling capacity. $T_{\text{mix}}$ should be sufficiently low to guarantee adequate cooling capability; Meanwhile, it should also be sufficiently high (above the ceiling panel surface dew point $T_{\text{dew}}$) to prevent condensation. In our design, $T_{\text{mix}}$ is controlled to achieve the target $T_{\text{mix}}^t = \max\{T_{\text{sup}}, T_{\text{dew}}\}$. $T_{\text{dew}}$ is calculated based on the readings collected from 6 temperature and humidity sensors deployed below the ceiling panel as shown in Figure 4(b).

- If $T_{\text{sup}} > T_{\text{dew}}$, the cold water from tank can be supplied to the ceiling panel directly, i.e., $T_{\text{mix}} = T_{\text{sup}}$.
- If $T_{\text{sup}} < T_{\text{dew}}$, the recycle pump is enabled to increase $T_{\text{mix}}$ to $T_{\text{dew}}$ for preventing condensation.

Given $T_{\text{mix}}$, we further control $F_{\text{mix}}$ to provide suitable cooling capacity according to the preferred temperature $T_{\text{pref}}$ (set by the occupant) and the current room temperature $T_{\text{room}}$. We use the temperature difference $\Delta T = T_{\text{room}} - T_{\text{pref}}$ to control $F_{\text{mix}}$. $T_{\text{room}}$ is computed by averaging temperature readings from a set of sensors deployed in the room. If $\Delta T$ is positive, $F_{\text{mix}}$ is increased to provide more cooling power; Otherwise, $F_{\text{mix}}$ is decreased. To achieve a rapid and robust control of $F_{\text{mix}}$, we adopt the Proportional-Integral-Derivative (PID) algorithm in the control. The PID controller takes the preferred temperature $T_{\text{pref}}$ as the input, transforms it to a flow rate value $F_{\text{mix}}^t$ as output. $F_{\text{mix}}$ acts as the control target of $F_{\text{mix}}$. $F_{\text{mix}}$ is adapted by properly controlling voltages $V_{\text{sup}}$ and $V_{\text{rcyc}}$ of the supply and return pumps. The adaptation of $F_{\text{mix}}$ impacts the indoor temperature, which deductively feedbacks to refine the $F_{\text{mix}}$. The PID controller finally outputs a stable flow rate value when $\Delta T$ approaches to zero.

2) Sensor development and deployment: We embed 8 ADT7410 digital temperature sensors (“temperature sensor” in Figure 5) in the water pipes to measure $T_{\text{mix}}, T_{\text{sup}}, T_{\text{rcyc}}$ for both ceiling panels. The sensing accuracy is $\pm 0.5^\circ\text{C}$. We develop an interface board, named Control-C-1, as shown in Figure 5(a), which embeds an ARM micro-controller LPC1114 to connect ADT7410 temperature sensors via I^2C interfaces. The board is then integrated with a TelosB mote for computation and communication. We calculate $T_{\text{mix}}^t$ on this board.

We embed 6 VISION-2000 flow sensors (“flow sensor” in Figure 5) in the water pipes to measure $F_{\text{mix}}, F_{\text{sup}}, F_{\text{rcyc}}$. A flow sensor outputs a series of pulses and the pulse frequency is proportional to its measured flow rate. We use two DC pumps (“water pump” in Figure 5) to circulate water in the cooling system, which takes a voltage signal ranging from 0V to 5V as the input to control its speed. We design another control board, named Control-C-2, as shown in Figure 5 to collect the flow rate data and control the DC pumps. The attached TelosB mote receives temperature measurement from Control-C-1 as well. The PID controller for $F_{\text{mix}}$ is implemented on this board. Based on the PID output, the micro-controller LPC1114 instructs the DAC to generate appropriate voltage values to drive the two DC pumps. Both boards are deployed behind the water pipes and wired to the pipe sensors (see Figure 4(a)).

C. Distributed ventilation module

1) Distributed ventilation control logic: Distributed ventilation module controls indoor humidity and CO$_2$ concentration levels. It contains a tank supplying cold water (8$^\circ$C), and four pairs of airbox and CO$_2$ flap, dividing BubbleZERO into four equal subspaces. Each subspace contains one airbox and CO$_2$ flap pair as shown in Figure 6. Airboxes and CO$_2$ flaps cooperate to ventilate the room and four pairs are controlled separately. An airbox consists of four DC fans (inhale air), one damper (prevent the air leakage when fans are not working), one filter (remove dusts), and 3 copper pipes (dehumidify) circulated with cold water. A CO$_2$ flap is integrated with an exhaust channel. By driving the flap, CO$_2$ flap exhausts the indoor air to the outside.

The ventilation module blows dry and fresh air into the room to neutralize both the indoor humidity and air CO$_2$ concentration to the target levels. As the outdoor air is usually humid, it needs to be dehumidified before blown into the room. The outdoor air with high humidity decreases. Two parameters are controlled to achieve effective ventilation, i.e., the dryness of the airbox output air...
output air from airboxes (measured by its dew point \(T_{\text{dew}}^a\)) and the ventilation speed \(F_{\text{vent}}\).

We first introduce how to decide the dew point of the output air from airboxes \(T_{\text{dew}}^a\). We denote its control target as \(T_{\text{dew}}^r\), which is determined by other three values \(T_{\text{dew}}^a\), \(T_{\text{supp}}\), and \(T_{\text{dew}}^p\). \(T_{\text{dew}}\) is the dew point calculated by the preferred temperature \(T_{\text{pref}}\) and humidity \(H_{\text{pref}}\) set by the occupant. \(T_{\text{supp}}\) is the water temperature in the tank for radiant cooling. \(T_{\text{dew}}^r\) is the current room dew point calculated using the average room temperature \(T_{\text{room}}\) and humidity \(H_{\text{room}}\) that can be obtained from temperature and humidity sensors deployed in the room. We set the target dew point of the room air \(T_{\text{dew}}^a\), to \(\min\{T_{\text{dew}}^p, T_{\text{supp}}\}\). To ensure the humidity requirement from the occupant and avoid condensation, \(T_{\text{dew}}^a\) is determined by:

- If \(T_{\text{dew}}^a < T_{\text{dew}}^r\), \(T_{\text{dew}}^a\) is set to \(T_{\text{dew}}^r - 2^\circ\text{C}\) to quickly pull down the room air dew point to the preferred level.
- If \(T_{\text{dew}}^a > T_{\text{dew}}^r\), \(T_{\text{dew}}^a\) is set to \(T_{\text{dew}}^r\) to maintain the room air dew point.

The flow rate of the circulated water inside the copper array in airboxes is linearly proportional to the dew point of the air, i.e., a higher flow rate leads to a lower output air dew point. This property suggests to control the water pump speed for approaching the target output dew point \(T_{\text{dew}}^a\). We design a similar PID controller as that in the radiant cooling module to ensure the control accuracy and convergence speed.

DC fan speeds determine the ventilation volume. The exact amount of air needed individually for dehumidification \(V_{\text{humd}}\) and \(CO_2\) concentration deduction \(V_{\text{CO}_2}\) can be calculated based on the volume of the room, the difference between the current and target humidity levels, and the difference between the current and target \(CO_2\) concentration levels. To promptly approach to the control targets in \(T\) seconds (e.g., 60 seconds), the final ventilation speed \(F_{\text{vent}}\) is calculated by \(\max\{F_{\text{humd}}, F_{\text{CO}_2}\}\), where \(F_{\text{humd}} = V_{\text{humd}}/T\) and \(F_{\text{CO}_2} = V_{\text{CO}_2}/T\). According to the airbox hardware specification, we can lookup the best matched DC fan speed for the given \(F_{\text{vent}}\). When DC fans are working, \(CO_2\) flaps are open, driven by a stepper motor, for exhaust.

2) Sensor development and deployment: We embed one DC pump and one VISION-2000 flow sensor in the supply pipe (for each airbox) from the tank. All sensors and pumps (of four airboxes) are connected to another control board the same as Control-C-2 depicted in Figure 5(b), named Control-V-1. Control-V-1 locally calculates \(T_{\text{dew}}^a\). It also communicates with Control-C-1 to obtain \(T_{\text{supp}}\), and collects readings from temperature and humidity sensors deployed in the room to calculate \(T_{\text{dew}}^r\). Based on the difference between the measured \(T_{\text{dew}}^a\) and its target, the PID controller is implemented on Control-V-1 to adjust the flow rate.

The commercial airbox fans used in BubbleZERO only have RS232 ports for interfacing. To control their speeds, we integrate with a TelosB mote through an RS232 adapter and collect readings from Control-V-1 (for \(T_{\text{dew}}^a\) and \(H_{\text{room}}\)) and \(CO_2\) sensors deployed in the room. The TelosB driver thus drives the speeds of DC fans with RS232 read and write. We denote the TelosB based driver as Control-V-2. In addition, SHT75 sensors are also connected to the customized TelosB motes through the extended pins on the motes. Such temperature and humidity information is needed by Control-V-1 for the \(T_{\text{dew}}^a\) calculation. Figure 7(b) depicts our developed RS232 adapter and Control-V-2.

Similar to airboxes, we integrate the \(CO_2\) flap with TelosB via the RS232 adapter. We upgrade the TelosB-based driver to control the stepper motor and also communicate with the \(CO_2\) sensor of \(CO_2\) flaps, as shown in Figure 7(c). We denote the TelosB based control board as Control-V-3. Each \(CO_2\) flap is integrated with one Control-V-3 board which is attached to the ceiling panel as shown by Figure 7(d).

IV. WIRELESS NETWORKING

As the previous section introduces, the BubbleZERO system involves rich control and sensing devices distributed over the space. Each of the control boards and special purpose sensors is integrated with a TelosB mote such that all devices can compute and communicate through TelosB motes. Figure 8 depicts the data supply and consumption relationship among those devices. Each arrow in the figure indicates one pair of supplier and consumer. Sensors serve as data suppliers, while most control boards serve as data consumers and some of them also provide processed data to others. One data packet is usually needed by multiple destinations. On the other hand, the system also features heterogeneous power supplies. Some devices get AC power (e.g., \textit{ac-device}), while others only get batteries due to practical limits (e.g., \textit{bt-device}). This section describes our effort in developing a suitable wireless networking mechanism that address these challenges.

A. Message exchange

Unlike most traditional sensor network solutions that first aggregate the sensor data at some “sink” node [14], we let the data suppliers directly feed the data to the consumers in a peer-to-peer style, which complies with the distributed
var min = 0, var max = 10
C1 C2 C3 C4 C5
U1 = 5
U2 = 10
U3 = 3
U4 = 7
U5 = 5

![Diagram of BubbleZERO network](image)

Figure 8. Data supply and consumption relationship among different devices.

decision making nature of our system. As Figure 8 shows, the same supplied data may be consumed by multiple control modules, so we do not explicitly address the receivers of the data. Instead, we let the suppliers categorize and address its data messages to certain “types”, e.g., temperature, humidity, CO2 concentration, etc, and broadcast data to the wireless channel. All potential consumers fetch data messages from the wireless channel and filter out messages with undesired types. Since TelosB motes can reliably communicate up to 50 m in the indoor environment, all potential consumers can receive messages with high success rates in BubbleZERO. Devices pick up the useful ones for calculation and control according to their application needs, which makes the best use of the wireless broadcast effect and thus saves unnecessary transmissions. When multi-hop communication must be concerned in large-scale environments, we can potentially extend our design by forming “type” based multicast groups and routing messages with existing ad-hoc multicast approaches (e.g. [22]). We leave it as an important future work of this paper.

B. Adaptive sensory data transmission by bt-devices

A half of devices in BubbleZERO are powered by batteries. It is prohibitive to configure bt-devices in an always-on mode; Otherwise, batteries last less than one week. In fact, controllers can preserve control accuracy as long as the granularity of the input data captures environmental dynamics. To prolong the network lifespan meanwhile meet control needs, we thus duty cycle data sampling and transmitting for bt-devices.

Sampling period. The period length of data sampling can be empirically investigated. We first analyze sensory data stability in stable indoor environments. We further trigger different events in BubbleZERO, e.g., opening door, opening window, occupant density varying, occupant transition between different rooms, which influence the indoor environments, e.g., temperature, humidity, CO2 concentration, and analyze the resulting sensory reading dynamics. Through experiments, we observe that indoor environmental factors demonstrate high stability while external events could cause readings to diverge from the stable state for several minutes. To ensure a high sampling granularity, we set the data sampling rate two orders higher than the dynamic duration. In particular, the sampling period $T_{spl}$ for temperature, humidity, CO2 concentration sensors in BubbleZERO is set to be 3 s, 2 s, and 4 s, respectively.

Sending period. As packets in BubbleZERO are addressed by data types, e.g., temperature, humidity, CO2 concentration, different types of sensor data are transmitted using separate packets with individual transmission periods. After transmitting a data packet, a bt-device sets a timer for the next packet of the same data type using the corresponding transmission period $T_{snd}$. When timer becomes expired, it sends out the next packet and resets the timer using $T_{snd}$ (the value of $T_{snd}$ may change which will be detailed soon). Due to the fact that the power consumption of reading on-board sensors is much lower than transmitting packets, e.g., 0.3 mW for sampling and 54 mW for transmitting, we duty cycle sampling and transmitting with different periods. When sensory data are stable, a long $T_{snd}$ is used to save bt-devices’ energy. $T_{snd}$ will adjust to be short if the environmental dynamics occur. In particular, $T_{snd}$ is set to be $w$ times of the sampling period $T_{spl}$, i.e., $T_{snd} = w \times T_{spl}$, where $w$ is a positive integer. Parameter $w$ is adaptively determined based on the data variance within a sliding time window.

With $m$ samples in the time window, $X = \{ x_1, x_2, ..., x_m \}$, the data variance is calculated as $\text{var}(X) = E[(X - E(X))^2] = E(X^2) - (E(X))^2$. Sliding window advances whenever a new sample enters the window and the new variance is calculated. If the variance is greater than a threshold $\lambda$, the sensory reading is considered not stable, i.e., in a transition state. The device adjusts $T_{snd}$ the same as $T_{spl}$ and immediately resets the timer using the updated $T_{snd}$ ($= T_{spl}$) to provide a prompt and fine-grained sensory data updating. If the variance is smaller than $\lambda$, the device regards the reading to be stable and remains to follow the current transmission period $T_{snd}$, whereas $T_{snd}$ is doubled if the variance does not exceed the threshold after 10 successive $T_{spl}$s, until an upper bound is reached. We set the maximum $w$ to be 32 in our current implementation.

Threshold $\lambda$. Threshold $\lambda$ aims to divide historical vari-
ances into two clusters. The cluster with smaller variances represents the stable state and another cluster represents the transition state. How $T_{end}$ is adapted is determined by which cluster the new variance value falls into. We note that a fixed $\lambda$ configuration limits the feasibility and applicability of the system as the environment dynamics levels are highly diverse in different time and areas. Threshold $\lambda$ should be determined automatically as the system executes. The optimal threshold is the one that minimizes the total intra-cluster distance, i.e., summation of the distances from each variance value to its cluster center, which can be learned from historical variance values. However, it is not practical to locally store all specific variance values as it requires increasing amount of memory space and computation complexity in the clustering. To address this issue, we propose a histogram based mechanism to approximately record variances and compute $\lambda$, trading certain classification accuracy for constant memory occupancy and computation complexity. In our method, devices record the maximum and minimum variances, $\text{var}_{\text{max}}$ and $\text{var}_{\text{min}}$, observed so far, and divide its difference into $N$ slots. Each slot is of $\Delta \text{var} = (\text{var}_{\text{max}} - \text{var}_{\text{min}})/N$ step length and any slot $i$ is represented by its slot center $c_i = \text{var}_{\text{min}} + (i - 0.5) \cdot \Delta \text{var}$, where $i = 1, 2, \cdots, N$. Instead of storing all historical variance values, devices round each variance value to the closest slot center and maintain a counter $U_i$ to count the number of variance values falling into each slot $i$. Devices use such a histogram to approximate all variance values within each slot. In Figure 9 for cluster 1, $U_1 = 5$ means five variance values falling into slot 1. For any slot $i$, devices only maintain a counter $U_i$. The memory occupancy of our method is thus constant for any given $N$.

Given $\text{var}_{\text{max}}, \text{var}_{\text{min}}, N$, and $U_i$, where $i = 1, 2, \cdots, N$, we numerate all possible $N-1$ clusters, i.e., slots $j$ to $j$ form the first cluster, denoted as fcls, and slots $j + 1$ to $N$ form the second cluster denoted as scls, where $j = 1, 2, \cdots, N - 1$. Centers of clusters fcls and scls are $c_{cc} = \frac{1}{2} \sum_{k=1}^{2} \text{var}_{\text{min}} + (k - 0.5) \cdot \Delta \text{var}$ and $c_{cc} = \frac{1}{N-2} \sum_{k=j}^{N} \text{var}_{\text{min}} + (k - 0.5) \cdot \Delta \text{var}$, respectively. We find the index $j^*$ which minimizes the total intra-cluster distance as the optimal classification position and set the threshold $\lambda = \text{var}_{\text{min}} + j^* \cdot \Delta \text{var}$. The threshold $\lambda$ selection is formalized in Algorithm 1. In principle, the optimal $\lambda$ can be calculated whenever a new variance value is observed. Instead of storing all historical variances, histogram values will be rounded to $N$ new slot centers. In Section V, we evaluate the impact of the $\text{var}_{\text{max}}$ and $\text{var}_{\text{min}}$ dynamics on the clustering performance. We observe that after observing certain events, $\text{var}_{\text{max}}$ and $\text{var}_{\text{min}}$ become stable and our histogram based method achieves comparable performance with the method using each original variance value. On the other hand, after Algorithm 1 runs for a long time, e.g., one week, each $U_i$ can be reset to be zero to eliminate approximation errors cumulated in the past week.

V. PERFORMANCE EVALUATION

The full BubbleZERO system has been implemented and stably operated in several real experimental trials. We install TelosB based sniffer nodes to collect all network packets and log all control data with time stamps, based on which we conduct full analysis on the system performance. To examine the energy efficiency, we also install power meters at major energy consuming devices, including chillers and pumps. The power consumption of remaining devices can be calculated based on their working statuses (e.g., voltage and workload) and durations from the data logs.

A. HVAC performance

This section evaluates the HVAC performance of BubbleZERO. For a fine-grained demonstration, we look at the four equal subspaces, each containing one airbox and CO$_2$ flap pair, and examine the performance for each subspace. Figure 10 summarizes the experimental results from our latest trial.

Overall performance: Figure 10 depicts the result logged from the experiment conducted from 13:00 to 14:45 in one afternoon. The experiment includes two phases. Initially, the indoor condition is similar to the outdoor, with 28.9°C temperature, 27.4°C dew point of humidity. The control targets of temperature and humidity dew point are set to 25°C and 18°C respectively. In the first 60 minutes, the system boots up, approaches, and maintains at the target state. From Figure 10(a), we see that the system cools down all four subspaces from 28.9°C to 25°C in the first 30 minutes. Afterwards, the room gradually approaches and maintains at the target temperature value. The humidity

<table>
<thead>
<tr>
<th>Algorithm 1 Threshold $\lambda$ selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Sum}_{\text{min}} = +\infty$</td>
</tr>
<tr>
<td>for $(j = 1; j &lt; N; j + 1)$ do</td>
</tr>
<tr>
<td>$\text{Sum}_{1} = \text{sum of intra – cluster distance in fcls}$</td>
</tr>
<tr>
<td>$\text{Sum}_{2} = \text{sum of intra – cluster distance in scls}$</td>
</tr>
<tr>
<td>if $\text{Sum}<em>{1} + \text{Sum}</em>{2} &lt; \text{Sum}_{\text{min}}$ then</td>
</tr>
<tr>
<td>$\text{Sum}<em>{\text{min}} = \text{Sum}</em>{1} + \text{Sum}_{2}$</td>
</tr>
<tr>
<td>$\lambda = \text{var}_{\text{min}} + j \cdot \Delta \text{Var}$</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>Return $\lambda$</td>
</tr>
</tbody>
</table>

It is clear that the computation complexity of our method is also a constant for any given $N$. During the system execution, if either $\text{var}_{\text{max}}$ or $\text{var}_{\text{min}}$ is changed, histogram values will be rounded to $N$ new slot centers. In Section V, we evaluate the impact of the $\text{var}_{\text{max}}$ and $\text{var}_{\text{min}}$ dynamics on the clustering performance. We observe that after observing certain events, $\text{var}_{\text{max}}$ and $\text{var}_{\text{min}}$ become stable and our histogram based method achieves comparable performance with the method using each original variance value. On the other hand, after Algorithm 1 runs for a long time, e.g., one week, each $U_i$ can be reset to be zero to eliminate approximation errors cumulated in the past week.
reduction has a similar trend. Dew points of four subspaces drop from 27.4°C to 18°C in 30 min and stay around this value as depicted in Figure 10(b).

We introduce disturbance in phase two to examine the control stability. The second phase starts from 14:00 and lasts 45 minutes. At 14:05, we open the door for 15 seconds, but do not enter the room, during which the hot and humid outdoor air enters the room. As the door is in subspace-1 and close to subspace-2, the humidities of the two subspaces immediately increase but with a slight amount (0.6°C at its dew point). The airboxes in subspace-1 and subspace-2 promptly react and the humidity change is quickly controlled. Figure 10(b) shows such a process (between 14:05 and 14:15). Meanwhile, the temperatures of subspace-1 and subspace-2 also slightly increase but are quickly pulled down to the target value. At time 14:25, we leave the door open for a relatively long period of 2 minutes. Both the temperatures and the humidities of all four subspaces are significantly increased. As Figure 10(a) and (b) show, the system reacts and adapts back to the target temperature in 15 minutes.

In summary, Figure 10 demonstrates the effectiveness of BubbleZERO in accurately achieving the control targets and promptly responding to environment dynamics.

B. Energy efficiency

We use the standard Coefficient of Performance (COP) metric to examine the energy efficiency of BubbleZERO. For a cooling system, \( \text{COP} = \frac{\text{Removed heat from the room}}{\text{Consumed power}} \). A larger COP value indicates higher energy efficiency. We install energy meters to measure the consumed power of the chillers. We calculate the removed heat from the room of the cooling and ventilation systems based on the water supply measurements: with the suppling water of supply temperature \( T_{\text{supp}} \), return temperature \( T_{\text{ret}} \), and flow rate \( F \), the removed heat is \( P_{\text{remove}} = c \cdot F \cdot (T_{\text{ret}} - T_{\text{supp}}) \), where \( c \) is a constant related to the water thermal capacity and density.

We compare the energy efficiency of BubbleZERO with prevalent Aircon HVAC systems in Figure 11. Traditional air conditioning systems (AirCon) achieve the COP of about 2.8 \([23][26]\). For BubbleZERO, the power meter results show that the radiant cooling module absorbs 964.8 W of heat from room, while its chiller consumes 213.4 W of electrical power. Ventilation module absorbs 213.2 W of heat from inhaled air and consumes 75.6 W. We thus can calculate the COP of BubbleZERO according to the measurement results. In particular, the COPs of the radiant cooling module (Bubble-C) and the ventilation module (Bubble-V) are 4.52 (= 964.8/213.4) and 2.82 (213.2/75.6), respectively, which translate to an overall COP value of 4.07 (= 964.8+213.4)/(213.4+75.6), denoted by BubbleZERO in Figure 11. Compared with “AirCon”, BubbleZERO improves the energy efficiency by up to 45.5%.

C. Networking performance

Overall BubbleZERO meets the control requirements with high energy efficiency. For a fine-grained understanding, we detail the performance of our adaptive transmission methods for both bt-devices and ac-devices in this subsection. To this end, we re-launch BubbleZERO for 5 hours in one afternoon and trigger external events, e.g., door opening and window opening, about every 30 minutes. In addition, we install flash memory to each device and record the ground truth of all received and generated data by the device with time stamps for the performance analysis.

Choosing the right \( N \). We first investigate parameter \( N \) in the histogram based mechanism, which divides the variance difference \( \text{var}_{\max} - \text{var}_{\min} \) into \( N \) slots to cluster the newly observed variance and decide how to adapt transmission period \( T_{\text{snd}} \). In general, a larger \( N \) leads to higher clustering accuracy while incurs more computation overhead and buffer occupancy. By referring to the data log, we can get the adaptation decisions made by each bt-device. As we also logged the ground truth of all variance values, we can further use exact variance values to conduct clustering and obtain the optimal adaptation decisions. For each bt-device, we define accuracy as the ratio between the number of adaptation decisions made by our histogram mechanism which are the same as the corresponding optimal decisions, and the number of adaptation decisions made in total. The accuracy depicted in Figure 12(a) is the average accuracy among all bt-devices. Figure 12(a) shows that when \( N \) is large enough, inconsistent adaptations due to the histogram approximation are minimal and the accuracy reaches around 98%. However, a large histogram size could incur extensive
computational and storage overhead to bt-devices, e.g., when \( N = 60 \), it takes 130 bytes (out of 10K bytes RAM) to store the entire histogram (Figure 12(b)) and 1600ms to complete clustering (Figure 12(c)). To balance accuracy, storage, and computation, we select \( N = 40 \) as the default setting.

**Accuracy as time elapses.** Figure 13 further depicts the average adaptation accuracy of bt-devices as time elapses given \( N = 40 \). Initially, the accuracy is relatively low. It is because \( var_{\text{max}} \) and \( var_{\text{min}} \) are not stable before sufficient external events are encountered. When either \( var_{\text{max}} \) or \( var_{\text{min}} \) varies, the histogram is reformed and the approximation error might lead to clustering errors. Improper adaptations may either unnecessarily lead to a short transmitting period to waste energy or harm the timeliness of the updating. From the experiment results, we find that the latency for \( var_{\text{max}} \) and \( var_{\text{min}} \) becoming stable is short in practice. In particular, \( var_{\text{max}} \) stabilizes after 1.5 hour and \( var_{\text{min}} \) stabilizes after 140 seconds. Afterwards, the accuracy remains to be high, ranging between 97% to 99%.

The direct consequence of the transmission adaptation method is that \( T_{\text{snd}} \) of bt-devices varies to save energy for bt-devices. In Figure 14, we plot a snapshot covering five door opening events for one bt-device. When the indoor environments are stable (measured by the indoor dew point), \( T_{\text{snd}} \) for temperature data equals to 64 seconds which is the multiply of the sampling period (2 seconds) and the maximum \( w (= 32) \). After the door is open, the indoor dew point increases rapidly, and \( T_{\text{snd}} \) adjusts to 2 seconds promptly for a fine-grained updating against such dynamics. However, due to clustering errors, \( T_{\text{snd}} \) adaptation might be delayed as shown by the zoomed-in sub-figure in Figure 14. According to the statistics, the delay is slightly, where the maximum delay in this experiment trail is 4s and the average delay is 2.7s.

We further investigate the \( T_{\text{snd}} \) distribution in Figure 15. Compared with the Fixed scheme which conservatively sets \( T_{\text{snd}} \) to be the same as \( T_{\text{apl}} (= 2 \) seconds), our BT-ADPT scheme adaptively adjusts \( T_{\text{snd}} \) from 2 seconds till 64 seconds, and achieves an average transmission period of 48 seconds. According to the energy profile of bt-devices, if external events on average occur every 30 minutes, the battery powered nodes can sustain longer than 3.2 years with 2 common AA batteries benefitting from our transmission adaptation method. On the contrary, if the transmission period is fixed, the lifespan of bt-devices is reduced to 0.7 years merely.

**VI. RELATED WORK**

HVAC acts as a basic component in today’s buildings. Modern HVAC systems, however, still suffer from very low efficiency, wasting tremendous energy [3]. BubbleZERO boosts the efficiency of the HVAC system for cooling and air conditioning by leveraging high temperature water at \( 18^\circ C \) as the cooling medium. Similar ideas of the water-based radiation and distributed ventilation have been previously explored in [23] and [5]. Water based radiation has been explored for heating purpose [23]. Google also employs sea water for cooling the hot servers at its data centers [1]. Those two applications, however, work at relatively high temperature or with dry air, and never face the condensation challenge as BubbleZERO does. BubbleZERO employs distributed and wireless cooperated control modules to overcome such challenges.

Many sensing based approaches have been explored to facilitate the HVAC control. The Smart Thermostat [21] instruments buildings with passive infrared sensors to learn the occupant behavior patterns and adaptively control the programmable thermostat to save energy. [2] further investigates the deployment of cheap occupancy sensors in buildings for aggressively duty-cycling the HVAC system without compromising users’ comfort. Thermvote [10] leverages participatory sensing and takes human feeling of comfort for efficient HVAC operation to achieve both human comfort and energy saving. There are abundant other works that study optimal HVAC control strategy in response to occupancy activities [11], [9], [24], [8], [25]. A most recent study looks at the central HVAC system of a building and provides a unified way of building service abstraction and HVAC control [7]. Sentinel [4] further leverages existing Wi-Fi infrastructure to save HVAC energy in commercial buildings. None of existing works, however, aim at deeply instrumenting and coordinating the HVAC system itself to improve its energy efficiency as BubbleZERO does.

Many efforts have been put into wireless sensor based networking and monitoring. Gu et al. [15] propose a dynamic switch based forwarding scheme DSF to optimize data delivery ratio, delay, and energy consumption. Ganti et al. in [13] design a datalink streaming transmission paradigm for
high network throughput. [6] introduces a utility-based asynchronous flow control algorithm for congestion avoidance. Wireless bus [12] and Chaos [18] leverage the concurrent transmissions for higher throughput. A reliability-oriented transmission service for sensor networks is proposed in [20]. On the other hand, Zhu et al. [30] introduce a probabilistic approach to provisioning guaranteed QoS for distributed event detection. Lin et al. [19] introduce an adaptive transmission power control scheme called ATPC for wireless sensor networks. TriopusNet is proposed in [17] for the pipeline monitoring using wireless sensor networks. However, those works are not directly related to HVAC control and do not take control timeliness and power heterogeneity into consideration as targeted in this paper.

VII. CONCLUSION

This paper presents our implementation and experimenting experience of BubbleZERO, an energy efficient HVAC system with distributed cooling, dehumidification and ventilation. Leveraging a network of abundant sensors and customized control devices, we fully instrument the HVAC loop and provide coordinated control in a decentralized way. Our experiment results demonstrate that BubbleZERO meets the stringent control requirements with higher energy efficiency compared to traditional HVAC systems. In our future work, we will mainly focus on improving the scalability of BubbleZERO, including the extension to multihop networking conditions, improvement on its real time reactions, as well as optimized aggregation of sensing and control information, so as to support building level deployment and integration.

VIII. ACKNOWLEDGMENT

This work was established at the Singapore-ETH Centre for Global Environmental Sustainability (SEC), co-founded by the Singapore National Research Foundation (NRF) and ETH Zurich. We acknowledge the support from Singapore MOE AcRF Tier 2 grant MOE2012-T2-1-070, and NTU Nanyang Assistant Professorship (NAP) grant M4080738.020.

REFERENCES