Internet of Things for Buildings Monitoring: Experiences and Challenges

Adam Coates
School of Mathematics, Computing and Digital Technology
Manchester Metropolitan University
Chester Street
Manchester, UK. M1 5GD
adam.coates@stu.mmu.ac

Mohammad Hammoudeh
School of Mathematics, Computing and Digital Technology
Manchester Metropolitan University
Chester Street
Manchester, UK. M1 5GD

Kieran Gerard Holmes
School of Mathematics, Computing and Digital Technology
Manchester Metropolitan University
Chester Street
Manchester, UK. M1 5GD

ABSTRACT

In recent years, smart buildings have proliferated around the world, offering new solutions to the problems of smart living. This paper describes the authors' experience and lessons learned in deploying Internet of Things (IoT) for smart building monitoring and management. It addresses critical implementation issues related to the communication architecture to build reliable and versatile system to monitor small and medium sized building. The paper explains how to create services on top of the data gathering architecture. It then focuses on the deployment approach and gives the results from the test site.

CCS CONCEPTS

•Hardware → Sensor applications and deployments; Sensor devices and platforms; Wireless integrated network sensors;

KEYWORDS

Internet of Things, Smart buildings, Temperature monitoring

ACM Reference format:
Adam Coates, Mohammad Hammoudeh, and Kieran Gerard Holmes. 2017. Internet of Things for Buildings Monitoring: Experiences and Challenges. In *Proceedings of ICFNDS '17, Cambridge, United Kingdom, July 19-20, 2017,* 6 pages.
DOI: 10.1145/3102304.3102342

1 INTRODUCTION

Ever since the Neolithic time, buildings had the purpose of providing comfortable space to its occupants. Early structures were basic shelters made from wood, stones, animal skins and other material found in nature. Buildings evolved over time driven by numerous trends to increase the durability of the construction, level of control over the interior environment, entertainment, reduce running cost, etc. While they hardly compare to modern skyscrapers, the first buildings ever built had the same goal - to offer a comfortable accommodation for its occupants. Today's buildings are sophisticated mix of structures and technologies. Over time, every element inside a building has been enhanced and developed, offering building occupants the ability to control heating, ventilation, air conditioning, lighting, entertainment and security systems at the touch of a button, or even configure their surroundings to adapt autonomously to their lifestyle and activity.

The Internet of Things (IoT) has become a key innovation driver that is transforming every aspect of our lives. IoT exists nearly anywhere, from cars, homes, cities to wearable devices. Connecting objects to the IoT ecosystem is offering new data insights, which is shaping the digital economic era. The consistently growing range of connected devices available in the market offer new opportunities for smart building deployments in both the common pluggable and integrated forms.

The implementation of smart buildings has been held back by many issues, such as cost, scalability, integration and interoperability. As shown in Section 2, there is a wealth of literature and studies to deliver various smart building services. While simulation studies of smart building protocols, services and applications provide good insight into their performance and feasibility, real-world implementations provide a more accurate, realistic, and replicable validation mechanism for algorithms and protocols. This paper describes the authors' experience and lessons learned in deploying IoT for smart building light, temperature and occupancy monitoring. It describes implementation issues related to the communication architecture that would have been difficult to identify in simulation or analytical studies. The paper describes the deployment approach and gives the results from the test site.

The remainder of the paper is organized as follows: Section 2 gives the state-of-the art in architectural concepts for the smart building using the IoT Technology. Section 3 explains the deployment environment and the challenging characteristics of the studied building. Section 4 starts by presenting the IoT monitoring system deployment environment, then, it presents the details of the exchanged packets structure and the proposed routing protocol. Section 5 concludes the paper.

2 RELATED WORK

Smart building management systems utilize embedded sensor devices in order to operate. Sensors provide information on the current occupants, lighting, heating, ventilation, electrical and other machinery systems in a building. Current research in IoT smart buildings systems focuses on the indoor communication and sensor technologies. There is also a wealth of literature on applications and

ICFNDS '17, Cambridge, United Kingdom

DOI: 10.1145/3102304.3102342

impact of smart buildings on the environment, occupants comfort and running costs. The authors of [5] presented a comprehensive review of a number of deployment challenges that impact the scope of IoT utilization in smart buildings. This review covers the requirements of smart buildings and how IoT technology can improve building surveillance and reduce building costs. In this section, we focus on the state-of-the art in architectural concepts for the smart buildings using IoT.

Recently, Putra et al. [7] presented an investigation into the utilization of Bluetooth Low Energy (BLE) and WiFi for monitoring building occupancy. They further propose size-constrained BLE packets over HTTP for transmitting occupancy data with a suitable data structure to overcome the limited throughput of BLE packets. BLE was proven to be 30% more energy efficient than WiFi when used for building occupancy applications. This empirical study was conducted using a small number of mobile phone devices, which does not expose problems related into BLE bottlenecks and interference between Bluetooth and WiFi.

In [6], a conceptual framework to achieve simulation-based smart Building Energy Management Systems (BEMS) is proposed. The proposed control system is limited to lighting and temperature energy control, which are generated from combinations of weather data changes. In this study, the occupants' uncertainty is also limited to the desire of opening windows to create a new boundary condition, which is used for analyzing thermal performance of selected room.

The authors of [9] summarize their experience of using IoT to build a cost effective and multipurpose BEMS for small- and medium-sized commercial buildings. The paper discusses design and implementation issues learned from the author's experience in the deployment of IoT in buildings. This work focuses on IoT device integration using an open source platform designed specifically to operate buildings efficiently and save energy. While these research findings are platform-specific and based on testing a small number of IoT objects, they serve as reference for future research and real-world implementations of smart buildings.

In [8], the authors address the trade-off between energy consumption and occupant comfort in smart spaces. They propose a smart building manager that is able to detect contextual changes and adapt the building's energy consumption behavior accordingly. To avoid overconsumption situations, this system implements a basic set of energy saving tactics based on unverified user experience assumptions. The system was evaluated using a simple scenario of a simulated smart home, which does not contain enough complex smart objects and does not capture the energy consumption adaptation implications on occupant comfort.

The problem of integrating multi-vendor IoT devices into a single system has been investigated in the context of smart buildings in [1]. An alternative approach for API standardization that can achieve open services in smart buildings is presented. To support IoT application developers, functionality descriptions are provided via inspecting the required information during the application development and utilizing it in attribute endpoint. The authors implemented a smart building API in to evaluate the efficiency of the proposed design. The API was applied to a smart room as a use case. While the new API reduces the development time, it increases the complexity significantly. It requires application developers to accommodate all use cases and configure them into the application logic.

The work in [2] compares the performance of two different approaches to solve the room temperature modeling problem in a smart building. The first approach is based on a black-box identification process, which is suitable for real-time control. The second approach is based on first principles, hence, it requires advanced calibration procedures. However, the second approach was found to accurately describe the physical state of the monitored system. Similar to the work in [2], the authors in [4] investigate in-building temperature modeling and measurement. They propose adopting an orthogonal matching pursuit algorithm to improve the modeling cost and accuracy compared to the conventional least-squares fitting method. By combining temperature modeling with a small number of sensor readings using the Bayesian model fusion algorithm, the spatial temperature distribution can be accurately estimated. The studies presented in [2, 4] provide some insight into the utilization of sensor readings in optimizing control systems in a smart building. However, restricting the study to the temperature readings while ignoring other sensing modalities, sensor placement and application context, limits its usefulness.

In [3], the authors present an indoor temperature control of smart buildings that decouples the cost minimization problem into sub-problems. Every sub-problem is optimally solved only using the next-hour electricity price to take advantage of preheating/cooling. The proposed optimization system was demonstrated with real data and results show significant economic savings compared with the intuitive strategies. This system does not consider occupant comfort and other factors that could affect the building operation.

As shown in the reviewed studies, buildings have many requirements in terms of energy management, comfort, usability and security. IoT-based systems can provision these requirements at a very low cost. New IoT communication technologies, e.g., wireless and power line communication, as part of an IoT-based solution, offer unprecedented opportunities in revolutionizing the in-building connectivity of a large number of devices. In general, the majority of the available smart building solutions that can be found in the literature focus on the evaluation of the energy consumption of buildings using temperature sensors, and on the development of suitable control strategies at building level. Moreover, in literature there are studies that utilize various data collection techniques such as artificial neural networks. Consequently, different types of building monitoring models were developed under different perspectives and with various final objectives. In this paper, we present the authors' experience and lessons learned in deploying IoT for smart building monitoring and management. We focus on the vital implementation issues related to the communication architecture to build reliable and versatile systems to monitor small-and medium-sized buildings.

3 DEPLOYMENT ENVIRONMENT

The selected building for the IoT-system deployment offered a range of issues and hurdles, making it the ideal place to test such a system. The building offered a wide range of conditions to both provide useful data, i.e., some rooms significantly warmer than others, and to inhibit the collection of said data.

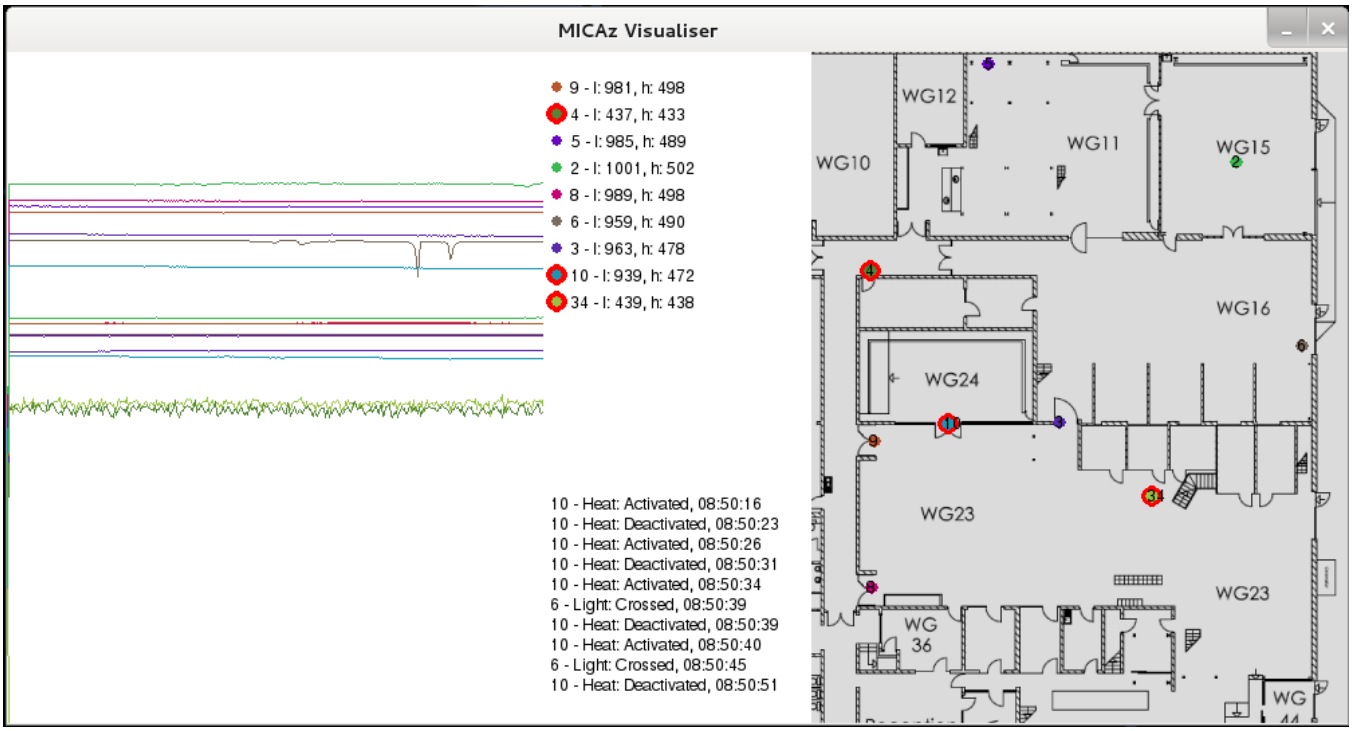

Figure 1: Building monitoring visualizer showing the floor plan.

The deployment occurred in a converted industrial building, characterized by high ceilings, thick walls, and the presence of a large amount of metallic obstacles. When combined with the already short range and low signal strength of the IoT devices used, this presented a significant obstacle to achieving full network coverage, which did not appear during simulation. Indeed, the final configuration required almost twice the number of nodes as the simulation suggested.

Due to the building's use as a technology center, there was a high amount of near-constant background traffic over a wide variety of bandwidths, protocols and systems. To contribute additional difficulty and thus aid testing, other researchers were not discouraged from having their systems interfere with the authors' system. While we expect an environment such as this to be uncommon in actual deployments this situation is one that can not be effectively simulated, further showing the need for physical as well as virtual testing.

The occupancy of the building remained roughly constant throughout the working day, although on several occasions the laboratory played host to large events, discussions, and gatherings on a wide variety of topics. This served as an invaluable opportunity to gather data on the movements of people through the building while it was significantly busier than usual, without the risk of influencing the results by staging an event.

Figure 1 shows a screenshot of the visualiser in our laboratory. This example deployment consisted of 9 devices, 2 of which were deployed without sensors to act as relays. We also experimented with 30 devices - more than enough to cover every potential entrance to the building.

The left hand side of the display shows the graphs of the various sensors. They can be visually divided into three bands, corresponding to the purpose of the node. From top to bottom, these bands are:

(1) Light sensors
(2) Heat sensors
(3) Relays

As a person walks past a light sensor, they obstruct enough of the ambient light to produce a characteristic dip in the readings. This can be seen on two occasions in the graphs. There is a major drawback to this, prohibiting its use as the sole security system in an area - it cannot detect a person moving in darkness. However, it is able to differentiate between a person walking towards the sensor, and one walking past it. The authors believe this could have applications, for instance in energy-saving automatic doors.

The heat sensors barely changed throughout the duration of the test. There are some limitations to the heat sensors used - even when placed beside one another, they rarely agreed on the temperature. It is likely some calibration process is required. Placing a finger on the heat sensor produces a very distinctive "fin" shaped curve, perhaps allowing a heat sensor to serve as a button that requires body heat to activate - it could not be triggered from afar using a pole.

The final band is used by network relays. As they are not equipped with sensors, they send their current voltage. This is

not particularly useful for building monitoring, but does allow the system to detect if a relay goes down and adjust itself accordingly.

The central section contains a numeric readout of the current light and heat levels for each sensor. Hovering over the coloured circle highlights the lines belonging to them, and also the location of that node on the map. Clicking the circles allows the locations to be reassigned by clicking the map. A red outline indicates that a node is currently experiencing an "event" - either a power outage, network collision, or the sensors registering an anomaly. Below the readout is a log of the last 10 events to occur. The rapid reconnecting from node 10 is an indicator of the node being almost out of transmission range.

The right hand side of the display shows the floor plan of the area, with the location of nodes indicated. Both the floor plan and the default node locations are loaded from a simple configuration file, to aid in setting up the system in a new location.

Initially, a simulation in Python was constructed to inform the placements of nodes. Nodes were placed at various "chokepoints" throughout the building, such as doorways and the tops of staircases. Where these points were too far apart, additional nodes were placed to bridge signals. However, due perhaps to the materials used in construction of the building, the simulation repeatedly overestimated the signal range of the nodes. Instead of requiring 12 nodes, the final layout required 20 to ensure reliable network connectivity. Using more nodes also allowed for the failure of some without compromising the entire network.

4 NETWORK FORMATION AND TOPOLOGY

The IoT devices used adhered to the IEEE 802.15.4 wireless standard, which is designed for low-cost, low-speed communication over short distances. As it does not define routing protocols - or indeed anything above the physical and MAC layers - a simple protocol was constructed for use during testing. We define two components: a packet structure, and rules for handling valid incoming packets. Invalid packets are silently dropped, as they are deemed to be interference.

4.1 Packet Structure

The default packet maximum size is 28 bytes on the hardware platform used in the deployment: MICAz[1]. While this was sufficient for our current purposes, a later revision may increase this limit to allow for encryption or other security features as well as increasing the number of readings transferred per message. While increasing the time between readings, this may be an acceptable trade-off for extended battery life. All packets have the following fields:

Offset	Size	Component
0	2	Version
2	1	Network ID
3	1	TTL
4	1	Type
5	2	Node ID
7	1	Count
8	20	Payload

[1]http:www.memsic.comuserfilesfilesDatasheetsWSNmicaz_datasheet-t.pdf

The first field, "version", declares the version number of the source code that the node is operating on. This allows a receiver to react differently to out-of-date nodes - for instance, by notifying a system operator that a node requires updating.

"Network ID" allows multiple networks to coexist in the same location without significant performance penalties from forwarding packets from any other network. Alternatively, forwarding the packets of other networks can be permitted and receivers tuned to a subset of available networks. This would allow, for instance, one operator to manage all nodes (i.e., those in networks *a* or *b*), while another operator can only take readings from network *a*. If an update alters the packet structure, such as by increasing the size of the payload, extra care must be taken to ensure all nodes are updated.

"TTL", Time To Live, serves the same purpose as in a typical routing protocol. It is decremented at each hop, and the packet dropped when the TTL reaches zero. While endlessly cycling packets are prevented using another mechanism (described in Subsection 4.2), that mechanism can fail in sufficiently densely packed deployments.

"Type" is an indicator of the content of the message, and describes how the payload should be interpreted. We declared two types - a control message, and a data message.

"Node ID" contains the ID of the sender. It is initially set during the flashing/building process, but could later be changed upon receipt of an appropriate control message.

"Count" is a monotonically increasing counter, resetting from 255 to 0. It is used to allow receivers to order packets should they arrive out of order, and serves a role in the routing protocol. Since the data throughput is low, a situation where two different packets with the same node ID and count are in transit at the same time will not occur.

Finally, the contents of the payload depends on the type of the message. Control messages specify the ID of the intended recipient, the action to perform, and any arguments required by that action. A data message simply contains the value of the last 10 readings from the sensors.

4.2 Routing

As no routing protocol is defined by the IEEE 802.15.4 wireless standard, we implemented a simple protocol with an emphasis on survivability in the case of node failure. We selected a peer-to-peer topology, where the peers of each node are simply defined as those within radio range - approximately 5 meters. All nodes are able to relay messages from their peers, and so they are required to be Fully-Functioning Devices (FFDs). At least one member of the network must be a base station, which bridges the signals to a serial port and from there, the monitoring station. In addition to bridging, base stations act as network coordinators, and may act upon control messages intended for any node.

Upon receipt of a packet, a node performs three checks on the validity of the data. It checks that it is not the original sender of the packet, that the TTL of the packet is greater than zero, and that it has not recently received that packet from another source. If these conditions are met, it decrements the TTL and then broadcasts the packet to its own peers. These conditions serve to eliminate loops, though the latter one does place an upper bound on the size of the

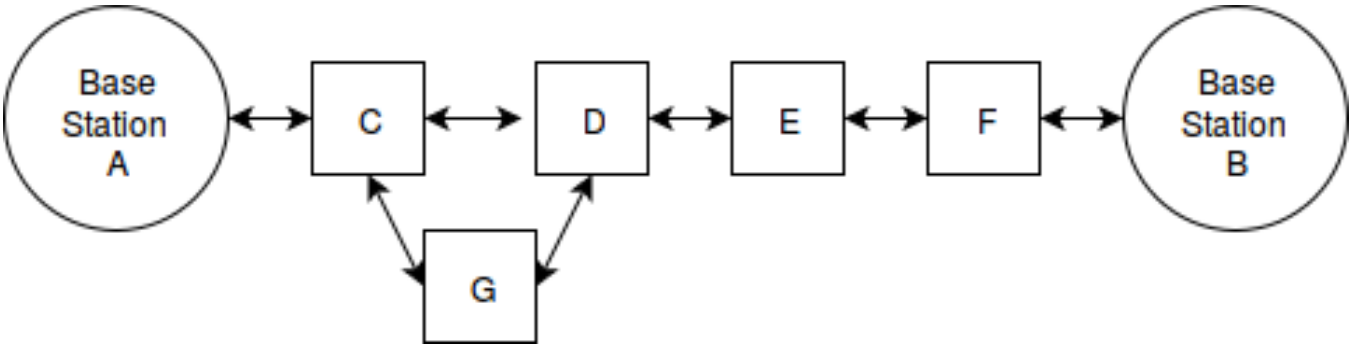

Figure 2: A simplified network with 2 base stations and 5 motes.

network. To remedy this, control messages can increase the TTL of packets sent by a specific node or cluster, and thus extend the range. Alternatively, base stations can respond to packets with a non-zero TTL by reducing the TTL of the originating node by that many, helping to reduce network congestion.

The effectiveness of this protocol depends to some extent on the network topology. Optimal performance is approached if the network resembles a tree - loops of longer than length 2 are impossible by topological constraints, and loops of length 2 are prohibited by the first of the two validity checks (that the recipient of a packet is not the original sender). However, as the network begins to approach a complete graph, performance degrades quickly without the appropriate corrective action taken with control messages. We expect such a topology to be unlikely in real-world deployments.

The base station is configured to expect a packet from each node every 2 seconds. Altering this signaling rate is a careful optimization task between battery life and resolution of data as the default packet size can not transfer more than 10 readings. Sampling the sensors less than once every 0.2 seconds caused important information to be missed, thereby causing the network to lose the ability to differentiate between different types of events. If the base station receives no data for more than three consecutive intervals, it sends a control message to that node instructing it to increase its TTL to increasingly larger number - in tests, we used first 5, then 10, then 20. If no response is received, the node is presumed to have failed. Otherwise, the protocol will automatically adjust the TTL back to the minimum required level. This prevents the failure of a node on the shortest path between nodes a and b from severing communication, provided there exists another path between the two nodes. Similarly, when a node is first activated it sends out a message with a TTL of 20, which is then readjusted back to the appropriate level.

If a base station receives the aforementioned control message from a different base station, it is now aware that it is not the base station furthest from the node concerned. In that case, it no longer considers itself to have jurisdiction over that node and will no longer control its TTL. In this way, each node is primarily controlled by the base station furthest, in terms of hops, from itself.

In figure 2, we see a simple network composed of 5 motes and 2 base stations. Suppose that mote G has just been switched on. It sends out its first message with a TTL of 20, which is passed through the network. Both C and D receive the message from the other, but immediately discard it due to having recently seen a message with the same ID and count. Base stations A and B each receive the message, and instruct G to alter its TTL to 2 and 4, respectively. G takes the higher TTL of the two commands, and A cedes control of G after seeing the higher TTL of the control message from B, and noting that G has adjusted its TTL.

5 BUILDING MONITORING

With the sensors available being limited to only light and heat, the first task was to find a method for gathering additional data using only these readings. The heat sensors were unable to reliably detect the heat difference caused by a single occupant of a room, although they were able to determine between an empty and crowded room. However, as seen in figure 1, the heat sensors require careful calibration to use to accurately estimate temperature. Furthermore, they provide their information as a raw number which appeared not to correspond to degrees Kelvin, Fahrenheit, or Celsius. For these reasons, our investigation focused on the light sensors.

We determined two methods of detecting the passage of people through doorways or chokepoints, using a single light sensor. While certainly more cost effective than smart doors, or motion detectors, neither of these methods were able to accomplish our original goal - monitoring the occupancy of rooms in real time.

The first method had us place a sensor at the top of a doorframe in such a way that when opened, the door obstructed the sensor. We were then unable to pass through the door without triggering the sensor in some way - any attempt we made to carefully remove it caused the sensor to waver in a detectable manner.

The second method involved placing sensors at around waist height, pointing across the door. Walking towards the door crossed the sensor, causing the light levels to briefly dip. In contrast, walking parallel to the sensor caused a more gradual change in the lighting levels, allowing us to distinguish different types of motion. Neither of these methods were able to differentiate between entry and egress from a room, preventing us from tracking occupancy even in a simplified environment of only three rooms.

6 CONCLUSIONS

Based on the presented smart, IoT-enabled building monitoring system, the authors advocate that IoT starts in the lab. The success

of any IoT system, product or service necessitates thorough implementation, evaluation, and continuous monitoring. To identify vulnerabilities, IoT devices hardware, operating systems and communication protocols, must be tested with realistic applications, scale and traffic. IoT systems must be deployed and monitored to ensure successful operation. There are many feasibility aspects and challenges to consider when designing short-range, low-power, indoor wireless IoT systems. Some of the common practical problems that were encountered during this project were: lack of protection against signal interferences, physical object obstruction, dead spots, detecting unintended signals, unauthorized network access, complexity of setting up wireless links as well as large gap between actual and theoretical bandwidth due to collisions and contention-based access. Future research avenues include connecting various IoT devices in a smart building using different communication technologies. PowerLine connections and network planning using in 802.11 WLAN will be considered in future deployment of a hybrid smart building system.

REFERENCES

[1] S. Bandara, T. Yashiro, N. Koshizuka, and K. Sakamura. 2016. Towards a standard API design for open services in smart buildings. In *2016 TRON Symposium (TRONSHOW)*. 1–7. DOI: http://dx.doi.org/10.1109/TRONSHOW.2016.7842883

[2] X. Chen, X. Li, and S. X. D. Tan. 2016. Overview of cyber-physical temperature estimation in smart buildings: From modeling to measurements. In *2016 IEEE Conference on Computer Communications Workshops*. 251–256. DOI: http://dx.doi.org/10.1109/INFCOMW.2016.7562081

[3] R. Deng, Z. Zhang, J. Ren, and H. Liang. 2016. Indoor Temperature Control of Cost-Effective Smart Buildings via Real-Time Smart Grid Communications. In *2016 IEEE Global Communications Conference (GLOBECOM)*. 1–6. DOI: http://dx.doi.org/10.1109/GLOCOM.2016.7841899

[4] D. Gorni, M. d. M. Castilla, J. D. flffilvarez, and A. Visioli. 2015. A comparison between temperature modeling strategies in smart buildings. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*. 1–4. DOI: http://dx.doi.org/10.1109/ETFA.2015.7301596

[5] D. Minoli, K. Sohraby, and B. Occhiogrosso. 2017. IoT Considerations, Requirements, and Architectures for Smart Buildings. *IEEE Internet of Things Journal* 4, 1 (Feb 2017), 269–283. DOI: http://dx.doi.org/10.1109/JIOT.2017.2647881

[6] J. Ock, R. R. A. Issa, and I. Flood. 2016. Smart Building Energy Management Systems (BEMS) simulation conceptual framework. In *2016 Winter Simulation Conference (WSC)*. 3237–3245. DOI: http://dx.doi.org/10.1109/WSC.2016.7822355

[7] G. D. Putra, A. R. Pratama, A. Lazovik, and M. Aiello. 2017. Comparison of energy consumption in Wi-Fi and bluetooth communication in a Smart Building. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. 1–6. DOI: http://dx.doi.org/10.1109/CCWC.2017.7868425

[8] E. Taktak, I. Abdennadher, and I. B. Rodriguez. 2016. An Adaptation Approach for Smart Buildings. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 1107–1114. DOI: http://dx.doi.org/10.1109/HPCC-SmartCity-DSS.2016.0156

[9] X. Zhang, R. Adhikari, M. Pipattanasomporn, M. Kuzlu, and S. R. Bradley. 2016. Deploying IoT devices to make buildings smart: Performance evaluation and deployment experience. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. 530–535. DOI: http://dx.doi.org/10.1109/WF-IoT.2016.7845464