



Energy Enhancement of Multi-application Monitoring Systems for Smart Buildings

Ozgun Pinarer, Yann Gripay, Sylvie Servigne, Atay Ozgovde

► To cite this version:

Ozgun Pinarer, Yann Gripay, Sylvie Servigne, Atay Ozgovde. Energy Enhancement of Multi-application Monitoring Systems for Smart Buildings. CAiSE'16: Conference on Advanced Information Systems Engineering - EnBIS: Energy-awareness and Big Data Management in Information Systems, Jun 2016, Ljubljana, Slovenia. pp 131-142, 10.1007/978-3-319-39564-7_14 . hal-01334900

HAL Id: hal-01334900

<https://hal.archives-ouvertes.fr/hal-01334900>

Submitted on 21 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Energy Enhancement of Multi-Application Monitoring System for Smart Buildings

Ozgun Pinarer^{1,2}, Yann Gripay¹, Sylvie Servigne¹, Atay Ozgovde²

¹ Universite de Lyon, CNRS

INSA-Lyon, LIRIS, UMR CNRS 5205, F-69621, France

Email: name.surname@insa-lyon.fr

² Galatasaray University, Dept. of Computer Engineering

Ciragan Cad. No:36, 34349, Istanbul, Turkey

Email: {opinarer, aozgovde}@gsu.edu.tr

Abstract. High energy consumption is a major problem in smart building systems. Existing studies focus on energy consumption of building not the deployed wireless sensors. These approaches are often fitted to a single monitoring application, and lead to static configurations for sensor devices. Moreover, immense raw data generated by the smart building should be used in terms of service. In this paper, we focus on the energy of the monitoring architecture itself and services that use this dataset. We study impacts of services on the energy consumption of monitoring architecture. We consider a monitoring system as a set of applications that exploit sensor measures in real-time, where these applications are declaratively expressed as (service-oriented) continuous queries over sensor data streams. We focalise on multi-application, tackle the optimization of application requirements to manage energy consumption. We introduce a Smart-Service Stream-oriented Sensor Management (3SoSM) that optimizes sensor configurations and manages sensor data streams.³

Keywords: sensor data management, smart building, wireless sensor network, continuous query processing

1 Introduction

Nowadays it is well known that traditional buildings are primary consumers of significant portion of energy resources, thus, most of the world's cities are covered by smart buildings. These buildings are designed based on sustainable construction standards to consume less energy than traditional buildings and to minimize their impacts on the natural environment. Smart building technology brings in some nice features such as security, comfort and accessibility, however it also enforces extra constraints to be capable of analysing immense generated data. If one facet is related to provide useful services for the occupants such as thermal comfort, air quality, physical security etc, the other facet is related to manage/process large and complex datasets. This constraint brings us to

³ This work is supported by INSA de Lyon LIRIS Research Laboratory

recent and popular research domain: Big Data. It represents a new era in data exploration and utilization [1] and covers sensing, analysing, sharing, storing, querying etc.

Smart building technology provides less consumed energy by the building, however it consists of wireless sensor devices and major challenge is the limited energy and lifetime [2]. These devices are autonomous in terms of energy and their energy consumption determines the lifespan. Sensor devices sample physical quantity measures and transmit them with defined acquisition and transmission frequencies. Smart building application requirements make sensor device consume energy. Thus, energy consumption of deployed wireless sensor devices and monitoring infrastructure should be observed during the execution.

In this study, we propose a monitoring service to benefit generated massive raw data and search impact of application requirements on energy consumption of monitoring infrastructure. Our proposed system supports multi-application, hence, it requires handling several data stream requests with different parameters for the same device and supporting dynamic requirements of applications. Commonly adopted static configuration does not optimize energy consumption of the monitoring system. Thus, we focus on optimization of energy consumption of this infrastructure in an adaptive way. This optimization process is based on requirements and adopts adaptive approach to manage energy consumption and quality of services of application and system. In this paper, we propose a dynamic sensor configuration mechanism to avoid unnecessary data measurements and to promote shorter/compressed data transmission for sensor devices. We present *Smart- Service Stream-oriented Sensor Management (3SoSM)*, an approach to optimize interactions between application requirements and *WSN* environment in real-time. A *3SoSM* system performs energy-aware dynamic sensor device configuration to lower energy consumption while fulfilling real-time application requirements.

The remainder of this paper is organised as follows: Related work is given in Section 2. Section 3 presents monitoring system. Optimization of application requirements is explained in Section 4. Section 5 gives brief descriptions of platform. In Section 6, experimental phase is given. Finally, conclusions are presented in Section 7.

2 Related Works

Smart building is a part of a pervasive environment that consists of physical devices, wireless sensors, actuators etc. These components and platforms compose the entire building management system and its infrastructure. This system is not restricted with *WSN* but also covers several different disciplines coming from different research domains: Smart building design, sensor data management, energy issue etc.

In literature, existing studies mainly focus on design and data management sides. [3] proposes a model for monitoring system: a real-time decision unit that interacts with sensors for diagnosis of the building's state and with the build-

ing's controllers to select the appropriate interventions. [4] introduces a design of a smart building system that covers automated response, policy-driven governance, occupant awareness based on a combination of flow-based sensor event processing and rule-based business event processing. [5] introduces an occupant-centric design based on gathering/visualization of high density sensor network dataset to make the pervasive environment sustainable. [6] proposes a model to a better understanding of urban phenomena by explore and exploit heterogeneous data from various sources.

Besides, energy consumption of building system is also discussed. [7] proposes an energy management technique to handle computational needs of ambient intelligence applications by using energy minimization workload assignment policies. [8] introduces the necessity of having a monitoring/control system for a building and proposes deploying digital smart meters that communicate wirelessly. Author searches which equipments and system characteristics are responsible for the energy consumption of the smart meters.

These studies are bounded by predefined building applications and application requirement-sensor configuration relation is not established. [9, 10] are the closest study to ours. [9] presents intelligent building architecture based on self adapting intelligent gateway. This gateway serves for service decisions, device management, aggregate data, occupant-based pattern generation, provision of energy management services. Author presents a novel self-adapting intelligent system that can save approximately 16-24% energy. Moreover, [10] presents self adapting algorithm for context aware system. Proposed approach detects and analyses changes in the environment, decides how system should react respecting the given set of policies. Deployed sensor devices and actuators are capable to take adaptation decisions.

Common point of these approaches is to propose a dynamic management system for smart building environment. Even though user preferences are processed, their approach does not benefit from potential reconfiguration of acquisition and transmission frequencies: sensor configuration stays static during the system lifetime. Moreover, high energy problem is not considered as a major issue. In our *3SoSM* approach, we introduce an energy-aware dynamic sensor reconfiguration process while fully fulfilling application requirements.

3 Multi-Application Monitoring System

Smart building management systems are one of the main application area of pervasive environment research domain. Smart building systems are composed by wireless sensor devices, hence, high energy consumption and limited service lifetime are crucial problems. In this study, we focus on monitoring system, architecture of smart building application and its own energy consumption.

▷ ***Monitoring Architecture for Smart Building Environment:*** For smart building architecture, we adopt a declarative monitoring architecture. This architecture has 3 main layers:

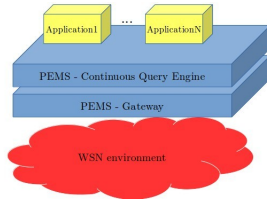


Fig. 1: Architecture

- *Application layer* where application requirements are defined, declaratively expressed as a set of continuous queries for sensor data stream requests [11];
- *Pervasive Environment Management System (PEMS)* where management of pervasive environment functionalities, integrating non-conventional data sources, dynamic and heterogeneous data sources and query executions are handled. It includes a continuous query engine that manages services provided by the lower

layer environment and requests coming from application layer [11, 12];

- *WSN environment* that consists of wireless sensor devices that acquire physical quantity measures, communicate with other sensor devices.

▷ ***Pervasive Environment Management Framework: SoCQ*** (Service-oriented Continuous Query) framework [11] is chosen for *PEMS* layer. It takes a data-oriented perspective on pervasive environments such as smart buildings. It provides a unified view and access to various and heterogeneous data resources (service providers) available in the environment. Pervasive applications can then be created in a declarative fashion using service-oriented continuous queries over such an environment. Within *SoCQ* framework, *XD-relations* (eXtended Dynamic Relations) represents standard relations, that may be updated, or data streams, that continuously produces data. *XD-relations* can interact with distributed services: service discovery, method invocation, stream subscription. Queries may be one-shot queries (like standard SQL queries) or continuous queries (with a dynamic result, like a stream). Furthermore, invocation of service methods and subscription to service streams can be parametrized.

▷ ***Query Structure:*** For instance, as a service discovery, a discovery query can search for sensor services that provides a location, a method to get the current acquired data, and a continuous stream of measures. The result of such discovery query gives ServiceID attribute, Location, and a virtual attribute for that service. Once, relevant services are listed, the continuous query to requested stream of every discovered services to build a resulting data stream (called subscription to service). If new services are discovered and/or some services become unavailable, the continuous query automatically adapt the corresponding stream subscriptions.

▷ ***Smart Service-Stream Oriented Sensor Management: 3SoSM SoCQ*** handles multi-application mechanism and several parametrized subscriptions to the same service. Moreover, it supports real-time user configuration of applications and context-aware applications. Service discovery and subscriptions are handled by a gateway positioned between *PEMS* and *WSN* environment. However, *SoCQ* does not tackle the issue of sensor configuration, thus, adopting a static configuration for sensor devices is a major challenge for energy enhancement of the system. Based on the existence of dynamically configurable wireless

sensor devices [13], we propose a novel approach to build an energy-aware application requirement driven behaviour into the *PEMS*.

From our perspective, a stream query is a set of application requirements that introduce user's expectations from the system. To improve energy consumption of the system, we propose a energy-aware dynamic sensor configuration based on real-time application requirements. An example of textual subscription is the following: "Application requests the average temperature over the last 10 minutes, with an update each 5 minutes, with accuracy 1 second and latency maximum 1 minute". From *3SoSM* approach point of view, parametrized subscription declares following application requirements:

- **temporal window size** introduces the time interval for calculating the result;
- **periodicity of update result** stands for periodicity of refreshing result;
- **data acquisition periodicity** represents the accuracy (precision) of measure;
- **maximum latency of result disponibility** presents the maximum tolerable time to process the acquired data which is the maximum time difference between the acquisition of data and its transmission to *SoCQ*;

Parameters *temporal window size*, *periodicity of update result* concern serving result to the application however, *data acquisition periodicity*, *maximum latency of result disponibility* parameters are related with sensor and data acquisition/transmission. In this study, we introduce a sensor configuration oriented pattern based on application requirements. This pattern is a sort of data acquisition/transmission schedule. This schedule is a set of timestamps that is generated by anticipation of sensor data acquisition and latency of that acquisition. Each pattern is private for a sensor device and formed by a combination of several subscription requests to that device. It dynamically configures sensor device in real-time when a new subscription or unsubscription occurs. Besides, it provides energy enhancement by avoiding system from unnecessary data measurements and by promoting data transmission shorter/compressed.

Briefly, proposed approaches in the literature adopt static sensor configuration and specialized for specific applications. Whereas, existence of dynamic sensor configuration feature offers diverse perspective such as new generation *Smart Service-Stream Oriented Sensor Management (3SoSM)* that provides less granular sensor configuration than duty-cycle and similar techniques. Our proposition, dynamic sensor configuration, is performed in a gateway layer in order to optimize consumed energy by a sensor device, independent from application layer and/or query engine.

4 Optimization of Application Requirements

In this section, we present our focus point: formalization of application requirements, handling multiple parametrized subscription requests and generation of sensor configuration oriented pattern for energy optimization.

4.1 Formalization of Application Requirements & Sensor Configuration

▷ **Query requirements:** A typical smart building is equipped with various wireless sensor devices d_i ($d_i \in D$ where D is the set of devices in the environment). Each sensor device has multiple functionalities to acquire physical quantity measures m_i ($m_i \in M$ ex: temperature, humidity etc). Presented *3SoSM* approach lets user to define application requirements: temporal window size (in sec) β , periodicity of update result (in sec) p^{update} , data acquisition periodicity (in sec) p^{acq} and maximum latency of result disponibility (in sec) $latency$. Definition of these parameters are introduced in the previous section.

A subscription request also includes information about device and requested measure. Thus, a subscription to a sensor device can be expressed as:

$$s_i = (d_i, m_i, \beta_i, p_i^{update}, p_i^{acq}, latency_i) \in D \times M \times \mathbb{N}^* \times \mathbb{N}^* \times \mathbb{N}^* \times \mathbb{N}$$

Based on this subscription request notation, application requirements can be written as:

$$\{s_i, s_j, \dots\} = \{(d_i, m_i, \beta_i, p_i^{update}, p_i^{acq}, latency_i), (d_j, m_j, \beta_j, p_j^{update}, p_j^{acq}, latency_j), \dots\}$$

▷ **Sensor configuration oriented pattern:** From multiple subscriptions $\{s_i, s_j, \dots\}$, we propose a **sensor configuration oriented pattern** P (*scop*). *scop* consists of a list of sensor events and a length of a pattern ℓ . An event is a $\langle timestamp, action \rangle$ couple. Here, we define two types of actions: either a **data acquisition** A or a **data acquisition and transmission** AT . Timestamps are enclosed by time interval $]0;length[$. The length of a pattern introduces the periodicity of that pattern that sensor executes this pattern repetitively. Thus, *scop* can be expressed as:

$$P = (\{(t_i, a_i)\}, \ell) \text{ with } \ell \in \mathbb{N}^+, t_i \in]0; \ell], a_i \in \{A, AT\}.$$

For instance, *scop* of a sensor that measures every second and transmits every 3 sec can be written as: $P = (\{(1, A), (2, A), (3, AT)\}, 3)$

▷ **Process:** Generation of a sensor configuration oriented pattern from multiple parametrized subscriptions is a process that consists of several intermediate steps. During this process, for each subscription a **data oriented acquisition pattern** ρ (*doap*) is generated. This pattern is based on p^{acq} and $latency$. It presents list of events (only acquisition actions) and p^{acq} as a length of pattern. An event, in this pattern, stands for a triple: $\langle timestamp, acquisition, latency \rangle$. Thus, *doap* can be expressed as: $\rho = (\{p^{acq}, A, latency\}, p^{acq})$

ρ is an intermediate pattern called *doap* to compose the final pattern P named *scop*. ρ is based on data acquisition and latency information, however, P presents a data acquisition/transmission time schedule with pattern length. Transmission action for P is calculated from p^{acq} and $latency$ parameters of multiple subscriptions.

4.2 Algorithm

We express each subscription request as a sensor configuration oriented pattern for sensor device. Each device executes given pattern repetitively. Even though, a

sensor device can have multiple functionalities such as temperature and humidity, for this work, we consider a single sensor device with multiple parametrized subscription requests for the same physical measure. We present *scop* for energy-aware sensor management from multiple parametrized subscriptions. Generation of a final pattern P has several consecutive steps.

1. **Generate patterns (doap)**: First algorithm generates *doap* ρ for each subscription based on parameters $p^{acq}, latency$.
2. **Normalize patterns**: Now we focus on the fusion of two patterns: generalization to n patterns is straightforward. For pattern fusion, all the patterns should have same length. Hence, a common data acquisition time on the time line is determined. This common acquisition time value represents the length of the final pattern. Lowest common multiple method is used $LCM(p_i^{acq}, p_j^{acq}, \dots)$. Then, coefficients for each pattern are calculated. This coefficient indicates how many times each pattern should be repeated in order to reach the length of final pattern. Thus, each pattern is extended with related coefficient.
3. **Merge patterns**: Merge process fuses multiple time patterns and obtain the merged pattern ρ_{merged} .
4. **Consolidate configuration (scop)**: ρ_{merged} indicates data acquisition, latency and periodicity of that pattern. Data transmission is not indicated yet. As a final step, transmission action is added to listed events of final pattern. Optimal transmission action points are calculated based on latency values of subscription requests. Since latency indicates the maximum tolerable time interval between the acquisition and its transmission, we consider the minimum latency for all subscriptions on the sensor ($\text{MIN}(latency_i, latency_j, \dots)$).

These steps are named *3SoSM GeNoMe-X* process (**Generate-Normalize-Merge-eXpanded**, improved version of our previous work) and illustrated in Figure 2.

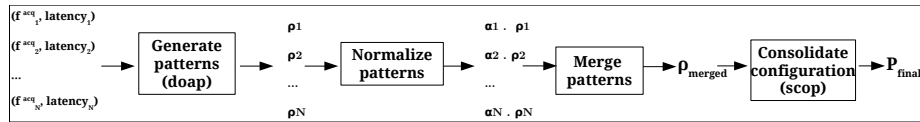


Fig. 2: GeNoMe-X Process

Example: Suppose that there are two subscription requests to the same sensor device d_i for measuring temperature m_t . First subscription requests data acquisition every 4 sec with latency 7 sec ($p^{acq} = 4, latency = 7$). Second subscription demands data acquisition every 5 sec with latency 9 sec ($p^{acq} = 5, latency = 9$). Regardless of β_i and p_i^{update} , subscriptions can be expressed as:
 $s_1 = (d_i, m_i, \beta_i, p_i^{update}, 4, 7)$ and $s_2 = (d_i, m_i, \beta_j, p_j^{update}, 5, 9)$
 Based on our algorithm, sensor configuration management follows these steps:

1. **Generate patterns (doap)**: these two subscription requests can be expressed as:
 $\rho_1 = (\{4, A, 7\}, 4)$ and $\rho_2 = (\{5, A, 9\}, 5)$
2. **Normalize patterns**: Common acquisition point is $\text{LCM}(4, 5) = 20$
 Then, for each time pattern, coefficient (α) is calculated. For ρ_1 , $\alpha_1 = 5$, for ρ_2 , $\alpha_2 = 4$. In other words, ρ_1 should be repeated five and ρ_2 should be repeated four times. *doaps* becomes:
 $\rho'_1 = (\{(4, A, 7), (8, A, 7), (12, A, 7), (16, A, 7), (20, A, 7)\}, 20)$
 $\rho'_2 = (\{(5, A, 9), (10, A, 9), (15, A, 9), (20, A, 9)\}, 20)$
3. **Merge patterns**: Now fusion algorithm can be executed to create the merged pattern. During the merge process, events are added to the final pattern in order of their timestamps. For the event that have same timestamps with different latencies, we take the one that has minimum latency:
 $\rho_{merged} = (\{(4, A, 7), (5, A, 9), (8, A, 7), (10, A, 9), (12, A, 7), (15, A, 9), (16, A, 7), (20, A, 7)\}, 20)$
4. **Consolidate configuration (scop)**: In the final step, for the entire pattern, common latency is calculated: $\text{MIN}(7, 9) = 7$ Then, transmission actions are added to relevant events according to $p^{acq}, latency$, thus final pattern *scop* is:
 $P_f = (\{(4, A), (5, A), (8, A), (10, AT), (12, A), (15, A), (16, AT), (20, AT)\}, 20)$

The final pattern is transmitted to sensor device and it is known that length of the packet has a direct effect on energy consumption of sensor device. Hence, based on computational power of sensor micro-controller, *GeNoMe-X* process could be realized on the sensor device when a final time pattern is long in terms of packet length. Moreover, instead of computing a single time pattern, real-time next-event calculation may be an alternative to be studied as a perspective of our work.

5 Experimental Platform

5.1 Continuous Query Engine: SoCQ Engine

In this study, the *SoCQ Engine* [11] is used. Perspective of the *SoCQ* framework as a pervasive environment management system was already introduced in Section 3. *SoCQ Engine* is a service-oriented continuous query engine implemented in Java language. A Data Description Language (DDL) allows to define *XD-Relations*, and a SQL-like query language allows to execute one-shot and/or continuous queries over *XD-Relations* [12]. This interface controls the query engine, allows users to visualize *XD-Relations* and their content, and to launch queries as well.

5.2 WSN Simulator: Modified WSNNet

In this work, *WSN* simulator *WSNNet* [14] is preferred. *WSNNet* is a modular event-driven simulator (discrete event simulator-DES). In *DES* models, between

consecutive events, no change in the system is assumed to occur, thus the simulation can directly jump in time from one event to the next. However, *PEMS* works on real-time. To avoid time scheduling difference, we modified the **time scheduler** of *WSNet* simulator: we introduced a time factor to run experiments in real-time or n -times faster ($\times 10, \times 20 \dots$).

5.3 Gateway: 3SoSM Gateway

The *3SoSM* Gateway stands for a technical bridge between two environments: *PEMS* and *WSN*. It manages interactions and bidirectional communication between *PEMS* and *WSN*. We implemented the *3SoSM* principles in a ***3SoSM Gateway***. It is implemented in Java and integrated with the *SoCQ Engine* and interacts with *WSNet*. *3SoSM Gateway* has two primary modules: ***Service Manager*** that manages sensor network devices and available services provided by sensors in the environment, ***Subscription Manager*** that is responsible for the real-time application requirements analysis and generation of time pattern for each sensor; and two secondary modules: ***Communication Interface*** that stands for bidirectional communication between the *Gateway* and the *WSN* environment, ***Sensor Discovery*** that is responsible for searching available wireless sensor devices (measure providers) in *WSN* environment.

6 Experiments

6.1 Experiment Setup

The simulations are performed on modified *WSNet*. We simulate the topology and deployed real sensor devices of our physical platform *SoCQ4Home* built in our research laboratory [15]: 70 simulated sensor devices are located specific positions over a floor of the building ($10\text{m} \times 60\text{m} \times 4\text{m}$, UDG propagation model, 35m transmission range). The deployed sensor devices have fixed positions during the simulation and we consider that they have enough energy until the end of simulation.

During the experiments, effects of stream management and sensor configuration on energy consumption are analysed. We adopt most known pervasive environment communication protocol *Zigbee IEEE 802.15.4* and basic radio module states. Calculation of energy consumption is based on CPU and radio components [16, 17].

6.2 Experimental Scenario

In each scenario, several applications send relational and parametrized stream queries to *PEMS*, *3SoSM* gateway manages existing data flow and generates optimal sensor configuration. Each scenario is performed with and without using *3SoSM* gateway during one day (1440min) to observe concrete performance of data stream management and evolution of energy consumptions. In this paper, we present only one of the performed scenarios.

Scenario Comfort Temperature: Smart buildings are responsible from thermal comfort of occupants. Based on the outdoor temperature, indoor comfort temperature interval is determined. During this scenario, there are two subscription requests to the same sensor device. First application requires temperature degree of each room with data acquisition 5 sec with latency 10 sec. Second application has a private condition. It demands to track more frequently temperature degree of rooms that are out of comfort temperature interval (with data acquisition 1 sec, latency 4 sec). Firstly, relational query is executed in order to discover available sensor devices that can provide temperature service. Then, subscription requests to services are sent to *PEMS* for sensor data streaming. Via *3SoSM* gateway, based on parametrized subscriptions, dynamic sensor configuration is realized.

First application receives temperature data flow from each room, however second application receives data flow from rooms in which temperature degree is out of comfort interval. If the room temperature is always in the comfort temperature interval, second application can not have a data flow according to given conditions. In other words, user wants to track temperature of these rooms more frequently until temperature is in the comfort range again. Since data transmission is the most expensive action in sensor side (in terms of energy consumption), we expect higher energy consumption for sensor devices that are in room where temperature is not in comfort range. This functionality shows the context-aware feature of *3SoSM* approach.

Experiment Result: Figure 3 shows the evolution of energy consumption of a single wireless temperature sensor and its lifetime during the simulation with and without *3SoSM* gateway. Upper graph presents outdoor, indoor temperatures and also dynamic comfort temperature intervals (dashed curves). Regions where indoor temperature is out side of comfort temperature interval are visible. Lower graph shows the decrease of energy budget and lifetime of that sensor device that is located in that room. Energy consumption increases while room temperature is outside of comfort temperature interval. While indoor temperature is outside of comfort temperature boundaries, second subscription is executed, hence, data transmission frequency is increased.

Results show that energy consumption of that sensor is higher in case where *3SoSM* gateway is not in use, as a consequence, sensor device dies earlier. Sensor device without *3SoSM* gateway nearly dies at the end of the day however, sensor device with *3SoSM* gateway has still energy. Energy budget of sensor device is initiated on purpose to observe the lifetime difference between cases. According to calculations, 27% of energy budget is saved by using dynamic sensor configuration approach.

Our proposed approach increases lifetime of a sensor device. When a basic 9V alkaline battery ($\sim 18.8\text{kJ}$) is connected to sensor device, with given parameters, it can survive 32.72 days without our approach, however its lifetime can be extended up to 41.71 days with our proposition. Here, *3SoSM* lengthens sensor lifetime $\sim 27\%$ for given scenario. Obviously, lifespan and the energy saving results depend on the application requirements.

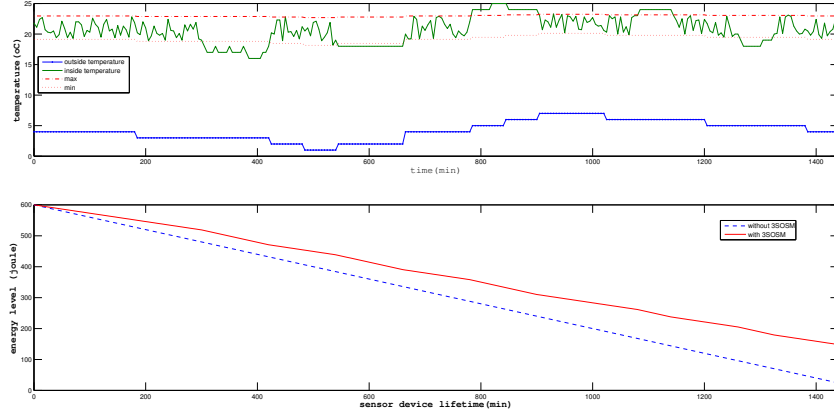


Fig. 3: Temperature of a room and energy consumption evolution of wireless sensor device for scenario-1 (24h)

7 Conclusion

In this paper, we point at a major challenge of smart building technology: energy consumption of the monitoring architecture itself while dealing with data exploration/utilization. We expose that existing studies do not tackle neither energy consumption of deployed equipments nor lifetime of the whole system. Existing approaches commonly adopt static configurations for wireless devices. Here, we focus on the lifetime of a monitoring system and introduce new mechanisms for dynamic sensor reconfiguration in terms of sensor data acquisition/transmission frequencies.

We present a sustainable declarative monitoring architecture to process massive raw data. We rely on declarative *PEMS* principles and tackle the energy optimisation of interactions between application real-time requirements and sensor devices. We introduce our approach *3SoSM* (Smart-Service Stream-oriented Sensor Management) based on data driven time patterns. We propose the *GeNoMe-X* process to optimize multiple parametrized subscriptions to a same device. We implemented a *3SoSM* Gateway that supports the *GeNoMe-X* (Generate-Normalize-Merge Expanded) process to fulfil dynamic application requirements. We conducted experiments using a modified *WSNet* simulator. Impacts of our approach on energy consumption and on lifetime are presented and discussed.

As perspectives for *3SoSM* approach, we plan to extend time patterns to support multiple measure types and to benefit from real raw data measured by physical *SoCQ4Home* platform. Besides, we also plan to integrate energy-aware dynamic sleep scheduling mechanism based on predictions, as a complementary of data driven scheduling. Furthermore, we are studying ways to decentralize parts of the optimization process into the *WSN*, on the smart devices themselves.

References

1. P. Zikopoulos, C. Eaton *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.
2. I. Aloui, O. Kazar, L. Kahloul, and S. Servigne, “A new itinerary planning approach among multiple mobile agents in wireless sensor networks (wsn) to reduce energy consumption,” *IJCNIS*, vol. 7, no. 2, 2015.
3. H. Doukas, K. D. Patlitzianas, K. Iatropoulos, and J. Psarras, “Intelligent building energy management system using rule sets,” *Journal of Building and Environment*, vol. 42, no. 10, pp. 3562–3569, 2007.
4. H. Chen, P. Chou, S. Duri, H. Lei, and J. Reason, “The design and implementation of a smart building control system,” in *ICEBE’09*. Citeseer, 2009.
5. A. Khan and K. A. S. Hornbæk, “Big data from the built environment,” in *Proceedings of the 2nd international workshop LARGE 11*. ACM, 2011, pp. 29–32.
6. S. Sylvie, Y. Gripay, D. Jean-Michel, N. Céline, J. Jacques, C. Olivier, and M. Radouane, “Data science approach for a cross-disciplinary understanding of urban phenomena: Application to energy efficiency of buildings,” *Procedia Engineering*, vol. 115, pp. 45–52, 2015.
7. M. Zapater, C. Sanchez, J. L. Ayala, J. M. Moya, and J. L. Risco-Martín, “Ubiquitous green computing techniques for high demand applications in smart environments,” *Sensors*, vol. 12, pp. 10 659–10 677, 2012.
8. M. Preisel, A. Diaz, and W. Wimmer, “Energy consumption of smart meters,” *Journal of Information and Communication Technologies*, p. 37, 2013.
9. J. Byun and S. Park, “Development of a self-adapting intelligent system for building energy saving and context-aware smart services,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 90–98, 2011.
10. T. Cioara, I. Anghel, I. Salomie, M. Dinsoreanu, G. Copil, and D. Moldovan, “A self-adapting algorithm for context aware systems,” in *RoEduNet’10*. IEEE, 2010, pp. 374–379.
11. Y. Gripay, F. Laforest, and J.-M. Petit, “Socq: A framework for pervasive environments,” in *ISPAN’09*. IEEE, 2009, pp. 154–159.
12. —, “A simple (yet powerful) algebra for pervasive environments,” in *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, 2010, pp. 359–370.
13. F. Dechesne, M. Warnier, and J. van den Hoven, “Ethical requirements for reconfigurable sensor technology: a challenge for value sensitive design,” *Ethics and Information Technology*, vol. 15, no. 3, pp. 173–181, 2013.
14. A. Fraboulet, G. Chelius, and E. Fleury, “Worldsens: development and prototyping tools for application specific wireless sensors networks,” in *IPSN’07*. IEEE, 2007, pp. 176–185.
15. “Socq4home project,” <http://liris.cnrs.fr/socq4home/>, accessed: 2016-02-04.
16. O. Pinarer and A. Ozgovde, “Improving the energy efficiency of wearable computing units using on sensor fifo memory,” *International Journal of e-Education, e-Business, e-Management and e-Learning*, vol. 5, no. 2, p. 105, 2015.
17. —, “Application specific dynamic sleep scheduling,” in *23th Signal Processing and Communications Applications Conference, SIU’15*. IEEE, 2015, pp. 1765–1768.