SOFTWARE

Open Access

A QR code based framework for auto-configuration of IoT sensor networks in buildings

Simon Soele Madsen, Athila Quaresma Santos* and Bo Nørregaard Jørgensen

From 1st Energy Informatics.Academy Conference Asia
Beijing, China . 29-30 May 2021

* Correspondence: aqs@mmmi.sdu.dk
SDU Center for Energy Informatics, The Mærsk Mc-Kinney Møller Institute University of Southern Denmark Odense Denmark

Abstract

Worldwide buildings are responsible for about 40% of the overall consumption and contribute to an average of 30% percent of the global carbon emissions. Nevertheless, most current buildings lack efficient energy management systems because such solutions are very expensive, especially when necessary instrumentation needs to be installed after the building's construction. As an alternative, we purpose the use of IoT sensor networks to retrofit existing medium and large-sized buildings to provide energy management capabilities in a cost-effective way. An IoT network auto-configuration platform for building energy management was developed. In order to efficiently manage metadata related to location and devices, a database using dynamic QR codes was created. Furthermore, we discuss the potential and shortcomings of different sensor-gateway pairing strategies that are applicable to an auto-configuring system. Lastly, we share our implementation of these concepts and demonstrate their use in a medium-sized building case study. The results show a trade-off between optimal configuration and total configuration time with a focus on the quality of the communication signal strength. The proposal provided the necessary automation for a cost-effective energy management system that can be deployed in both new constructions and existing buildings.

Keywords: QR code, Auto-configuration, Internet of things, Energy management systems, Buildings, Framework

Introduction

Over the last decades, we have witnessed a profound digitalization of society helped by the emergence of the Internet of Things (IoT). An estimative from the DBS Asian Insights calculates that the IoT installed base will grow from 6.3 M units in 2016 to 75B in 2030 (Columbus, 2018). The range of applications ranges from residential, commercial, and industrial categories spread over houses, buildings, and facilities that need to be managed efficiently, and effectively.

Globally, building sectors are said to consume 40% of the overall energy resources and contribute to an average of 30% of global carbon emission (Amasyali & El-Gohary, 2018). European Union (EU) countries utilize 32% of their energy to provide electricity for buildings (Comission, 2016). A study in 2018 on EU energy consumption finds that 50% of the energy is used for the building's space heating (ISI, ISE, IREES, EEG, TEP, 2016). Similarly, space cooling for buildings is also responsible for large energy consumption. The electricity demand across the Southeast Asia region, for example, is expected to double by 2040 (Team, T.A.P., 2018).

Additionally, buildings and their equipment are complex and massive systems. Not only are they large in the sense of physical size, but they are also large in the sense of the number of subsystems and their interactions. Moreover, modern buildings contain a significant amount of intelligence and control. While, in the past, buildings only had manual lighting switches in each room and a centralized heating system operating with fixed schedules, in the latest decades their complexity had increased enormously. The current state of the building is recorded through a complex network of sensors and meters, which measure indoor conditions such as temperature, air quality, and operating quantities such as ventilation rate, light intensity level, heating, and cooling signals. It is important that buildings are automatically and centrally managed by a building management system (BMS), which controls all building subsystems. However, low automation levels is common in most of the buildings, specially for complex features, such as B2G (building-to-grid), automatic configuration of devices, etc., (Ma & Jørgensen, 2018).

On the other hand, there is no silver bullet technology capable of improving the necessary energy performance of buildings (Jørgensen et al., 2015). Instead, several solutions, such as Building Automation and Control Systems (BACS) need to be considered (Billanes et al., 2018). Additionally, energy flexibility is possible, specially for office buildings and retail stores (Ma et al., 2018), where assets such as demand response, energy storage and distributed energy resources can efficiently respond to changes in the system (Billanes et al., 2017). Managing building energy consumption and sustaining an ideal indoor climate require extensive monitoring and sensing mechanisms. This is to ensure that accurate information can be obtained on the overall energy consumption and its indoor climate. However, most of the medium and large-sized buildings in the world have no energy management systems (Ma & Jørgensen, 2018). Several factors contribute to this, but the most prominent cause is because such solutions are very expensive, and most of the time, not economically feasible. This is especially critical when retrofitting existing buildings with wired sensors. Therefore, it is important to seek alternative solutions, such as the use of an IoT-based approach using wireless sensors.

Although there is a pressure to reduce GHG emissions by using energy efficient technologies (Ma et al., 2017), the market is very regulation driven and conservative. Several barriers prevent a broad adoption of IoT solutions towards the smart building markets, such as monetary risk, return on investment, follow up incentives, data protection, fear of uncertainty, etc., (Ma et al., 2016). The challenges in enabling IoT sensing technology into a building are the complexity of integration and the compatibility of different communication protocols, as well as data security (Vavra, 2018). Additionally, it also increases the complexity to control and manage large-scale IoT solutions, which over the years are becoming more decentralized, heterogeneous, and harder to configure manually. Bigger challenges arise when we consider cloud-based solutions and optimal configuration of tunable settings in several devices, such as gateways or router boxes.

Most of the efforts towards energy standards are applied to newly built buildings. This is detrimental for most of the existing buildings whose focus on energy renovation can improve their overall performance when taking functional, technical and economic constraints into consideration (Jradi et al., 2017a). Furthermore, most of the existing research on energy renovation and retrofitting focus on houses and residential building applications (Ferreira et al., 2016; Pikas et al., 2015). There is still a lack of investments on projects that aims to reduce the energy performance gap in public and commercial buildings, such as the work described in (Jradi et al., 2017b). In commercial buildings, the number of IoT devices often exceeds the 1–200 number limit significantly, as thousands of devices can be connected. In the standard solution, a lot of planning is needed, to connect the devices to a specific gateway, which is within range (Dzulkifly et al., 2020). This, combined with the fact that range is dynamic (people and furniture moved around), makes it of high value to have a system that automatically assigns each device to an appropriate gateway.

Therefore, the main contribution of this paper is to develop a middleware software for cost-effective large-scale deployment of wireless sensors and gateways in medium to large-sized buildings. As a second contribution, we propose a framework for installing and monitoring large-scale IoT sensor networks in buildings that supports any variety of IoT manufacturer devices.

A cross-platform mobile application for Android and iOS was implemented, which serves as an entry point for registering sensors and creating the needed association to a physical context. This is done by scanning the QR codes of the sensors and room. In order to increase flexibility, the metadata of the component is stored as an URL in the QR code that points to a database that can be altered. This provides flexibility because metadata can be changed in the database without any changes in the fixed QR codes.

In addition to this introduction, the paper is organized as follows. In section two, we place the proposed autonomic computing background in the context of the existing literature and the current advancements of IoT solutions. Section three explains the applied system architecture proposed by this work. Section four describes the most common automatic pairing strategies in the literature and how they are implemented. Section five presents key results of the conducted experiments and a general discussion in relation to the adopted implementation. Finally, conclusions and key implications are discussed.

Autonomic computing in context

In the last years, several works addressed the issues of automatic configuration and self-management of IoT networks. The most common methods to implement self-management are 1) Middleware (Familiar et al., 2012), 2) Software Defined Networking (SDN) (Huang et al., 2012), 3) Semantics (Perera et al., 2013), and 4) Autonomic Computing (IBM, 2005). Only the last one aggregates support for topology control, node registration, and heterogeneous nodes (Ashraf & Habaebi, 2015). Autonomic Computing refers to a generic self-management conceptual architecture proposed first by IBM in 2001 (Kephart & Chess, 2003). Autonomics can relieve the system administrator from meticulous and manual configurations. It detects context and conditions dynamically, reacting over its configurations to achieves optimal functionality. The architecture is based on a MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) control loop composed by several autonomic managers (Alaya et al., 2012; Movahedi et al., 2012).

The main objective is to develop network systems capable of reducing the barrier and to overcome the rapidly growing complexity of current and future IoT systems. In order to adjust their operation according to internal and external conditions, four aspects of self-management are defined (also called self-* terminologies), namely: self-configuration, self-optimization, self-healing, and self-protection. Autonomic systems are also known as intelligent management systems or plug-and-play systems (Ashraf & Habaebi, 2015). The same definition can be extended to network configurations (Autonomic Communication), i.e., how to address the unpredictable changes of topology, load, task, and the physical and logical characteristics of networks (Dobson et al., 2006). Autonomic Communications should be applied to IoT applications to optimize self-management of heterogeneous environments (Zhao et al., 2017).

Table 1 summarizes the recent work on Autonomic Computing and Communications. There were extensive surveys to characterize and classify the different approach of solving the network self-management (Dobson et al., 2006; Parashar & Hariri, 2005; Simmons & Lutfiyya, 2005; Sterritt et al., 2005). Open problems and challenges are described in several papers (Tahir et al., 2019; Papageorgiou et al., 2014; Nami & Bertels, 2007; Kephart, n.d.; Agoulmine et al., 2006). It is hard to achieve real autonomy, where a system should be able to make complex decisions without human intervention. A shift from centralized to decentralized architectures is defended by some authors, such as (Tahir et al., 2019). Centralized gateway configuration is not computing efficient because the IoT environment is heterogeneous by nature and because gateways usually have different tasks (Papageorgiou et al., 2014). Enabling technologies, such as Fog Computing, Edge Computing, Blockchain, and Smart Contracts, and Artificial Intelligence are discussed in order to fulfill the vision of an intelligent and autonomic IoT system.

Several frameworks were also proposed to handle the complexity of automatic configuration. The Framework for Adaptive Collaborative Ubiquitous Systems (FACUS) enables self-managed collaborative sessions between humans and devices. It is built upon a multi-level model approach and allows collaborative session configuration based on event communication modules (Bouassida Rodriguez et al., 2009; Sancho et al., 2008).

Table 1 Autonomics literature

Subject	Related papers
Categorization of the existing research efforts	(Dobson et al., 2006; Parashar & Hariri, 2005; Simmons & Lutfiyya, 2005; Sterritt et al., 2005)
Open problems and challenges	(Tahir et al., 2019; Papageorgiou et al., 2014; Nami & Bertels, 2007; Kephart, n.d.; Agoulmine et al., 2006)
Self-management properties	(Khalid et al., 2009; Khan et al., 2007; Claudel et al., 2006)
Autonomic frameworks and architectures	(Alaya et al., 2012; Bouassida Rodriguez et al., 2009; Sancho et al., 2008; Alaya & Monteil, 2015; Khalid et al., 2009; Huebscher & McCann, 2008; Nordbotten, 2009)
MAPE components	(Huebscher & McCann, 2008; Nordbotten, 2009)
IoT Configuration	(Perera et al., 2013; Ashraf & Habaebi, 2015; Alaya et al., 2012; Papageorgiou et al., 2014; Alaya & Monteil, 2015; Perera et al., 2014a; Perera et al., 2014b)

The FRAMESELF is a generic autonomic framework for self-management of distributed systems (Alaya & Monteil, 2015). It enables self-deployment and self-configuration of M2M applications, which gives support to IoT systems. In (Bouassida Rodriguez et al., 2009), the authors try to fill the gap of autonomic computing paradigm to build self-managed Machine to Machine (M2M) networks.

Commercial solutions that handle the self-configuration of large-scale IoT networks are still scarce. IoT Home solutions have dominated the market. In this kind of application, only one or few gateways are necessary and most of the configuration is done manually (Papageorgiou et al., 2014). The rise of Wireless Sensor Network (WSN) provided more flexibility to the system. The most commonly used communication protocols are Zigbee, Bluetooth, and WiFi (Hsieh et al., 2016). Gateways or router boxes aggregate sensors and actuators data. There are two traditional paradigms for interconnection: 1) Resource-oriented and 2) message-oriented. In the first, the most typical mechanism is REST over HTTP (Perera et al., 2014a). For the second, usually, MQTT is used (Tucic et al., 2014). Patented solutions such as OSGi (Bracke, 2007; Forum, 2018) were designed to provide mechanisms for remote configuration of gateway environments and their parameters. However, there is a lack of support regarding automation and system awareness, such as re-configuration impact, heterogeneity of the devices, system size, etc. (Papageorgiou et al., 2014).

Siemens is one of the companies that promote an IoT solution that is applicable for building, where a full IoT system can be implemented to observe the building's energy efficiency based on indoor temperature, lighting, and electricity consumption (SIEMENS, 2019). Another commercial IoT solution is known as WebNMS where its IoT solution is marketed to reduce electricity bills while increasing the energy efficiency (WebNMS, 2019). Cisco also offers a solution that enables smart automation to improve energy efficiency (Cisco, 2019). While these commercial solutions are already available in the market, there are limitations in terms of data availability, dynamic sensors integration, and lack of installation support for legacy buildings.

System architecture

This section introduces the proposed system architecture, which consists of 3 layers: 1) Edge components, 2) Backend and 3) Applications, as shown in Fig. 1. It uses a client-server architecture, where the clients are the users of a mobile application, whose purpose is to register location, gateways, and sensors, as well as their relationship to the system. Part of the solution is based on dynamic QR codes to describe physical contexts and logical associations.

It is necessary to create a methodology for the automatic configuration of large-scale IoT components in a building. Initially, tests need to be conducted to define the range between gateways and sensors to determine the number of gateways to cover a given area. Radio waves' strength decay very differently depending on the environment and technology used, which makes conclusions difficult. In this paper, we explore different pairing strategies between devices, such as 1) first come, first served, or 2) balancing the number of sensors in each room to optimize the communication signal strength.

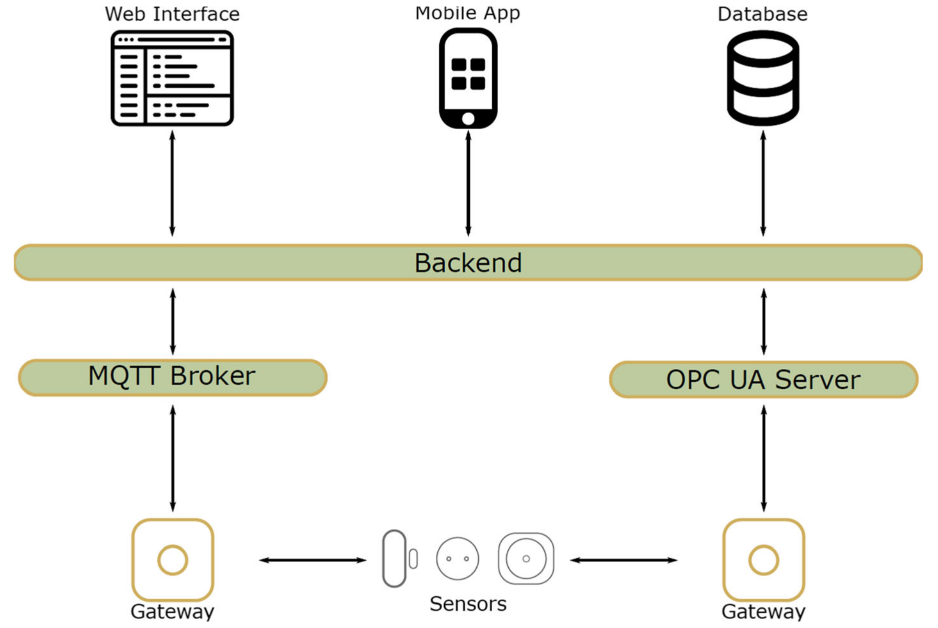


Fig. 1 System architecture

Edge components

In this category, two types of devices are used: 1) sensors and actuators; 2) gateways. A single gateway serves as a communication hub for many paired sensors. They are centralized elements with powerful processing capacity, making it possible to execute edging computing processes. Gateways also hide all the complexity of protocols and architectures of different manufacturers. The gateways publish data to the Backend middleware through MQTT and OPC UA interfaces.

Edge components, such as sensors and gateways are integrated into the Backend layer through a common interface. Frameworks such as MQTT and OPC UA help to integrate different system architecture, protocol, and manufacture solutions. The Backend is the middleware responsible to coordinate different protocols and architecture into one “spoken language”. This allows a “one solution covers all” strategy. This means that new applications or resource management need to adhere to one common requirement point. This solution facilitates the integration of new modules, improving scalability, as well as the decrease of maintenance effort for the overall system.

Applications

All applications and subsystems are connected directly to the Backend middleware. Messages can be exchanged between processes or among different edge components. The case study developed three applications on this layer: 1) web interface; 2) mobile app; 3) database. The backend exposes an API that the web interface consumes to display required data for monitoring the system. MQTT is used for getting updates on new events in real-time. The mobile app scans QR codes of new devices and sends the data to the backend for configuration. All information is stored on the database module.

Web interface

The purpose of the web interface is to communicate the status and layout of the currently configured network at a given physical site. Any changes during system runtime is reflected in the interface. Historical data is made available with a play-back feature to see changes in a given time span. The user experience goals of the web interface are listed below.

- Simplistic layout;
- Easily navigable and intuitive;
- Reflect the correct network status.

MQTT broker

The used gateways do not persist historical data from sensors to local storage. The gateways' software stack provides two ways to fetch data, 1) the MQTT protocol to subscribe and listen for new data, 2) request it via HTTP, taking a data snapshot. From a scaling standpoint subscribing to the flow of data makes much more sense. It provides low latency and ensures that only new data is received. With HTTP you construct requests that do not guarantee new data, and therefore resource wasteful to the server. As to not reinvent the wheel this paper utilizes the open source MQTT broker EMQ X (EMQX, 2021). It is a highly scalable broker that allows all gateways to facilitate messages reliably to other parts of the system, i.e. frontend or backend. Furthermore, EMQ X provides services such as event hooks and a rule engine that are useful for event-driven architecture and filtering data. In theory this component could be exchanged for an OPC UA Server or similar communication protocols. This is also seen on Fig. 1.

Dynamic installation

Installing a new IoT environment usually requires a lot of manual work and becomes labour-intensive with a large-scale operation with vast amounts of devices to configure. Vendor-specific implementations also add to the complexity of how an autonomous system should orchestrate different configurations. Therefore, in order to increase flexibility, this paper proposes QR code registration among different components of the system architecture. The QR code solution is an integral part of how we can manage the amount of labour required for the registration process. We provide a QR code abstraction for physical context and manufacturers provide QR codes for their IoT devices.

QR codes containing data of physical context are translated to a URL. This provides flexibility because metadata can be changed in the database without any changes in the fixed QR codes, see Fig. 2. Creating a QR code abstraction for the desired installation location and letting a backend service create the necessary configuration between gateways and sensors improve significantly the effort of installing and configuring large-

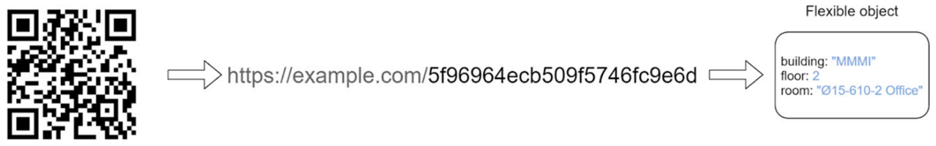


Fig. 2 QR code, URL and metadata abstraction of physical contexts

scale IoT components. Adapting QR codes as the initiation of registration would enable mobile phones and tablets as registration tools.

To accommodate the need for additional vendors a more generalized approach has to be implemented when parsing IoT manufacturer's QR codes in the mobile application. A simple solution is to associate a regular expression (regex) pattern with the manufacturer's QR code.

However, the responsibility of pairing is performed in an automated backend service, where the specific pairing between gateways and sensors is determined. By taking a URL representing the QR code, allows modification of metadata information without revisiting any physical devices. These are referred to as dynamic QR codes, where the URL itself cannot be modified but the end destination can be edited at any time. Such a configuration also means that any vendor must have some QR code integration on their sensors or additional work is necessary to generate, print, and glue them to be uniquely identified. Although there are many different protocols and QR code identification configurations, with an underlying versatile structure, only a few implementation tweaks are necessary to accommodate additional vendors. As the end goal an installation would be a 1–2-3 step process, where:

1. QR code of physical location is scanned;
2. QR codes of IoT devices are scanned;
3. Data is sent to some backend.

Backend

The backend is a web server that satisfies HTTP requests primarily from the mobile application for the registration of new sensors. Furthermore, the backend persists all mentioned objects; gateways, sensors, and physical contexts. For each new entry of a physical context an associated QR code is created and made available on the file system. These are available through a RESTful interface with the following endpoints:

- /api/locations enables registration of new locations and associates a QR code;
- /api/qr. allows for a folder of all QR codes of physical contexts or if specified a single QR code;
- /api/pair triggers a pairing service that implements a round-robin approach for specified devices.

Automatic pairing strategies

Manual pairing is a time-consuming task, as well as maintaining connection stability and ensuring proper operation, even when unintentional disconnection occurs, such as faults or anomalies. An automated algorithm could alleviate the otherwise inefficient process. After registering the desired IoT devices using a mobile device, an automated algorithm will try to determine the best possible pairing configuration.

There are many possible strategies to automate the pairing. Here we will discuss the following methods:

- FCFS (First come, first served):

It is a simple strategy, where all gateways request a pairing to the sensors concurrently. The pairing that finishes first is served. This strategy doesn't guarantee a stable connection or balanced distribution. Besides, it causes unnecessary load on the system, but it is simple to implement and will work if any gateway is in proximity.

- Historical strategy:
  It is a more clever strategy, where pairings are determined from the historical context of previous pairings. When installing a sensor in a room, we look up which previous gateway(s) have had pairings and try each gateway to determine the most stable connection. This strategy should be solid if the initial pairings are strong and would cause less stress on the system compared to FCFS.
- PCS (Physical constraint strategy) (Husz et al., 2012):
  It is an intelligent strategy, where gateways are limited to some physical contexts by supplying the system with infrastructural information. A very real nuisance of IoT environments is concrete walls. By constraining gateways to not communicate across obstructing elements eliminates the number of potential pairings. The drawback is the amount of preparation the strategy takes. One would have to coordinate all possible obstructions before having a chance of having stable connections between devices.
- Round-robin (Hidayat et al., 2020):
  The last strategy is a simple brute-force solution and guarantees the best connection. By trying every gateway and checking the signal strength one by one (hence the brute-force like). After the last gateway has been tried the sensor pairs with the most stable connection. This strategy is very slow, but pairings are usually lasting for a long time so it is ensuring stable prospects at the cost of a slow initial setup.

Algorithm 1 shows the pseudo-code for the round-robin strategy. Each time a batch of sensors is registered through the endpoint the auto-pairing begins for each sensor. The auto-pairing determines the best possible gateway and sensor pairing by the strongest signal strength.

```
Function Paring(sensor):
        // Variable for best gateway
    best_gateway;
        // Try all gateways
    for gateway ϵgateways do
        pair(gateway, sensor);
        if gateway is better than best gateway then
            best gateway = gateway;
        end
    end
        // Remember to pair with the best gateway
    if sensor is not paired with best_gateway then
        pair(best gateway, sensor);
    end
```

Algorithm 1: algorithm for gateway-sensor pairing.

It takes meticulous planning to map all devices and their respective locations, as well as the impact of construction material on the signal strength. Furthermore, it is of interest to minimize the number of gateways to reduce cost. Making a coverage analysis is very difficult due to the nature of radio waves. This is especially more difficult for more modern buildings that use materials that can interfere with wireless signals, such as metal infrastructure.

Building case study

In order to test the proposed solution, a setup is configured in a building zone on one of the floors of a research building in Denmark, as shown in Fig. 3. The MMMI building is located at the Odense Campus of the University of Southern Denmark and it was built in 1995 and today has an energy class C rank based on the Danish building standards. The facility has a multipurpose application with several room types including offices, laboratories, teaching rooms, workshops, meeting, and seminar rooms. The building operation hours are, usually, between 8 a.m. and 4 p.m. with no activities during the weekend. Four gateways were distributed in different locations to cover the whole section. Some of the internal walls are made of concrete and can attenuate the signal strength of the radio communication link (Dalke et al., 2000; Rodriguez Larrad et al., 2014). Between offices located in the same facade, drywall is used and has less impact on the signal quality. Table 2 shows the sensors installed in each of the layout rooms and their measurements and characteristics. Proximity and latency will define which gateway will be assigned to each sensor and actuator device.

All devices expose a REST API for configuration and the Zigbee protocol for communication between gateway and sensor. Depending on parameters such as distance, amount and type of walls, and interference between devices will define the signal strength on the communication. The average signal coverage for the gateways used in this experiment is between 15 and 20 m. Figure 4 shows that no open areas were considered. All offices are surrounded by obstacles as drywall or concrete walls with their structures, such as metal bars. The height of the ceiling is around 2.5 m. Because of the low-energy nature of the Zigbee protocol, a pairing can take some time. The sensors emit a ‘heartbeat’ about every 12–15 min. Only then will a connection be established to a gateway.

Figure 4 highlights the concrete and drywall. Drywall has a negligible effect on radio signals. Applying a PCS strategy in this case study would mean that gateways would pair to sensors in neighbor rooms. The northwest and northeast gateway would both be candidates of sensors in the intermediate rooms while the northwestern gateway would be the sole facilitator of the northwestern rooms.

Results and discussion

The entry point for registering devices and creating the needed association to a physical context is a cross-platform mobile application that relies on QR codes that are available at all registered locations, as shown in Fig. 5. Scanning the location and the IoT device QR code creates the information needed for the backend to auto-configure.

When persisted through the HTTP endpoint, the given devices will be configured. If this is a sensor, the backend will apply the chosen pairing strategy to bind the best

Fig. 3 Mærsk Building (Jradi et al., 2017a)

possible gateway. Otherwise, if this device is a gateway the backend will configure it to communicate with the MQTT broker. Whenever any sensors register new data, the MQTT broker will store it in the database.

To validate the proposed strategies and evaluate different configurations, the strategy proposed in Algorithm 1 was executed in a controlled environment. The case study described in Fig. 4 was used to simulate repairing sensors from one gateway to another during system runtime. Figure 6 shows the initial configuration of the system. Several sensors are distributed among four gateways, according to their proximity.

Table 2 Installed sensors

Sensor	Measurement	Protocol	Characteristics
Motion	occupancy, illuminance, temperature.	Zigbee	Shows activity on the room and illuminance in lux.
Window	activation, temperature.	Zigbee	Installed on windows and doors. Status shows if they are open or close.
Door	activation, temperature.	Zigbee	Installed on windows and doors. Status shows if they are open or close.

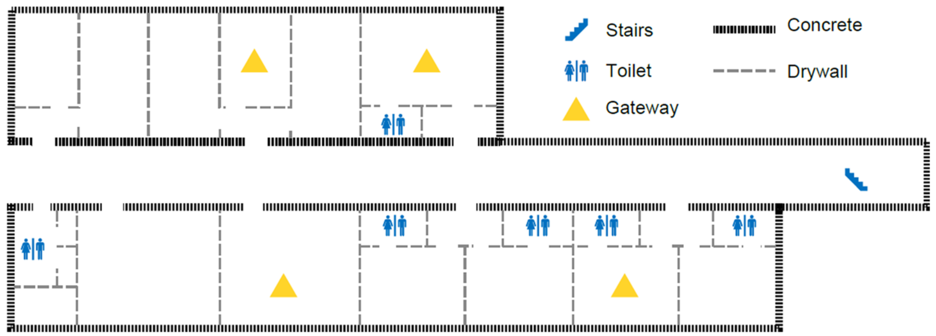


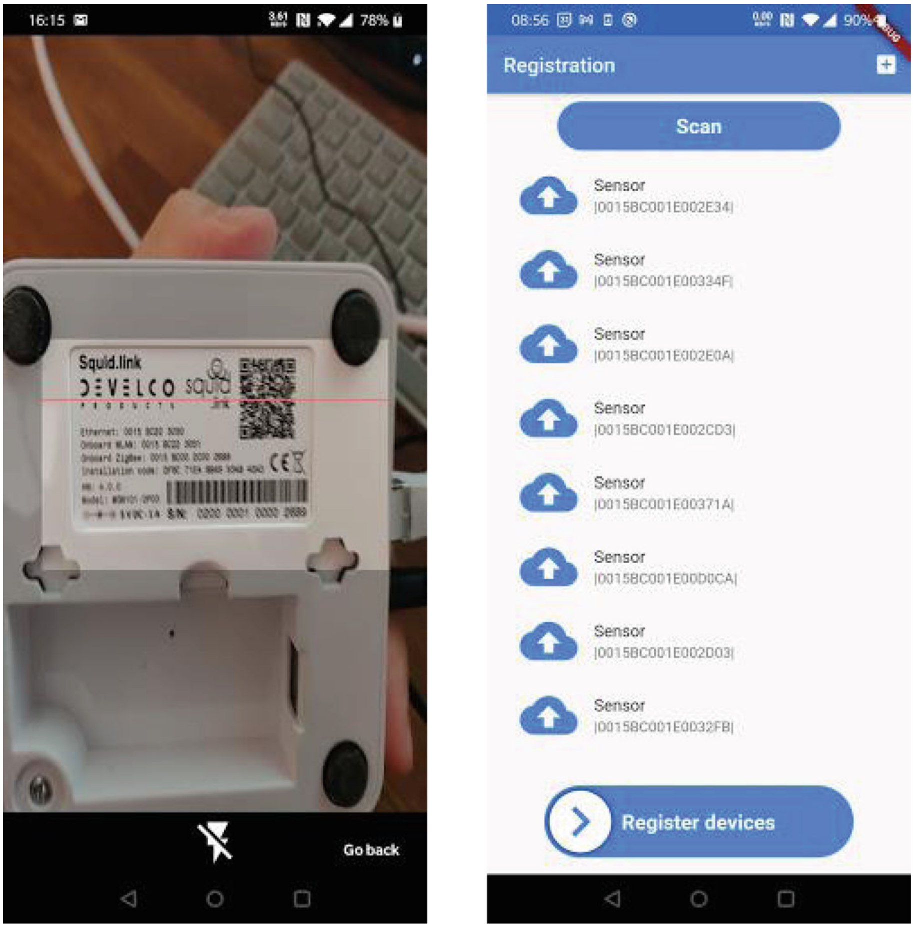
Fig. 4 Floor layout

Whenever sensors are removed from one gateway they are available to repair to new suitable gateways. Figure 7 shows the rounding robin results of a removing the gateway in room 2. By disconnecting the gateway, six sensors need to be repaired. By conducting the same procedure as for the initial pairing, Algorithm 1 managed five out of the six sensors to pair with a new gateway in the nearest room (Room 1). Thus trying to maximize the availability. The last sensor does not have an associated gateway because of a combination of distance and material obstruction that reduces the connection signal strength. This indicates that the minimum configuration with only four gateways is not enough to handle a repairing a batch of sensors. To increase reliability, it is necessary to invest in more gateways in the environment and make thorough evaluation when considering removing a gateway that might seem redundant.

For smaller environments, the strategy (Algorithm 1) is simple and effective. However, with the increase of complexity, considering larger buildings with multiple floors, the strategy might not suffice or a longer time intervals are needed to obtain the final configuration. The algorithm has a time complexity of O(n2), combined with the uncertain nature of radio waves make it hard to determine a configuration time estimation. For instance, when mocking the gateway disconnection in room 2, four sensors paired within 10 min, while the fifth paired within 1 h. A more sophisticated strategy is preferred in larger environments, such as the physical constraint strategy, where a low amount of gateways are considered for each sensor. As stated earlier, this method requires vast insight into the infrastructure of the building whereas Algorithm 1 is more adaptable.

Under normal circumstances more pairing strategies are available depending on type of device. This is done by utilizing a simple strategy pattern that executes a specific pairing strategy based on some parameters through an interface, see Fig. 8. A specific strategy is abstracted in a strategy type that during system runtime can determine the specific strategy if the device is of that type. The interface must explicitly state a fallback pairing strategy.

The features and implementation of the dashboard are discussed in the following paragraphs. For the example showcased in Fig. 9 the physical context has been mapped as follows: a building contains a number of floors, floors contain a number of rooms, and rooms contain a number of devices. The result is a flexible hierarchical data structure that can be visualized in different ways to convey the state of the network. The current



Sensor
|0015BC001E002E34|

Sensor
|0015BC001E00334F|

Sensor
|0015BC001E002E0A|

Sensor
|0015BC001E002CD3|

Sensor
|0015BC001E00371A|

Sensor
|0015BC001E00D0CA|

Sensor
|0015BC001E002D03|

Sensor
|0015BC001E0032FB|

(a) QR code scan.
(b) Mobile app.

Fig. 5 Mobile application interfaces

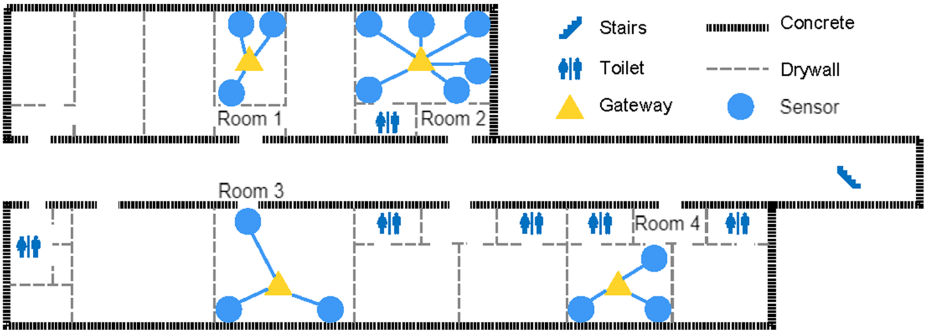


Fig. 6 Floor with sensors

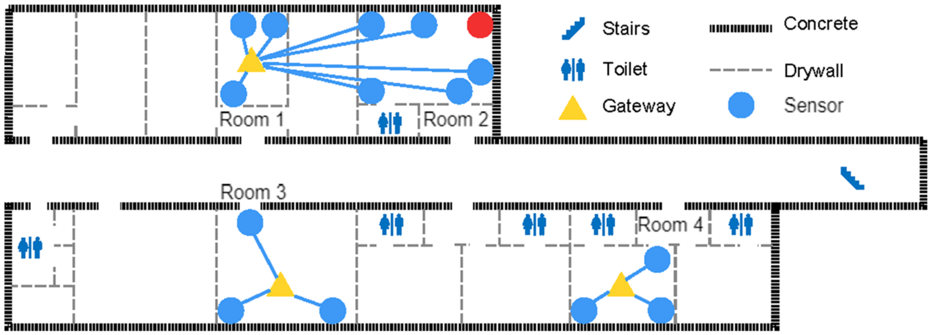


Fig. 7 Reconfiguration of sensors

web interface implementation features a tabular view of the data and a graph-based view. Both views provide unique information about the network.

The left column seen in Fig. 9 provides a tabular view of the network configuration. The current status is readily available in this view and network issues are highlighted in red. Any faulty devices are propagated through the table hierarchy such that a building is marked with a red exclamation mark if it contains any devices that require attention.

The right column contains a graph view of the network configuration. The purpose of the graph view is to reflect the current layout of the network and the connections between devices; information that is not available in the tabular view. Circles represent sensors and octagons represent gateways. All devices are color-coded in the same way as in the tabular view. Green devices are healthy, yellow devices have a non-critical issue that is being resolved by the system, and red devices require attention. The arrows between two devices represent a connection between a gateway and a sensor. The hierarchical data structure is also maintained in the graph reflected by the squares containing a sub-group of squares or devices.

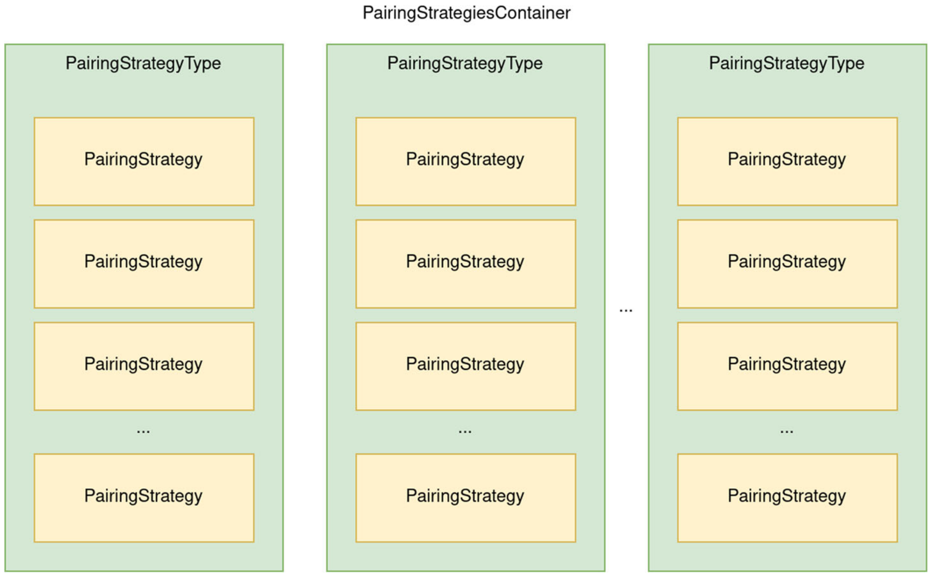
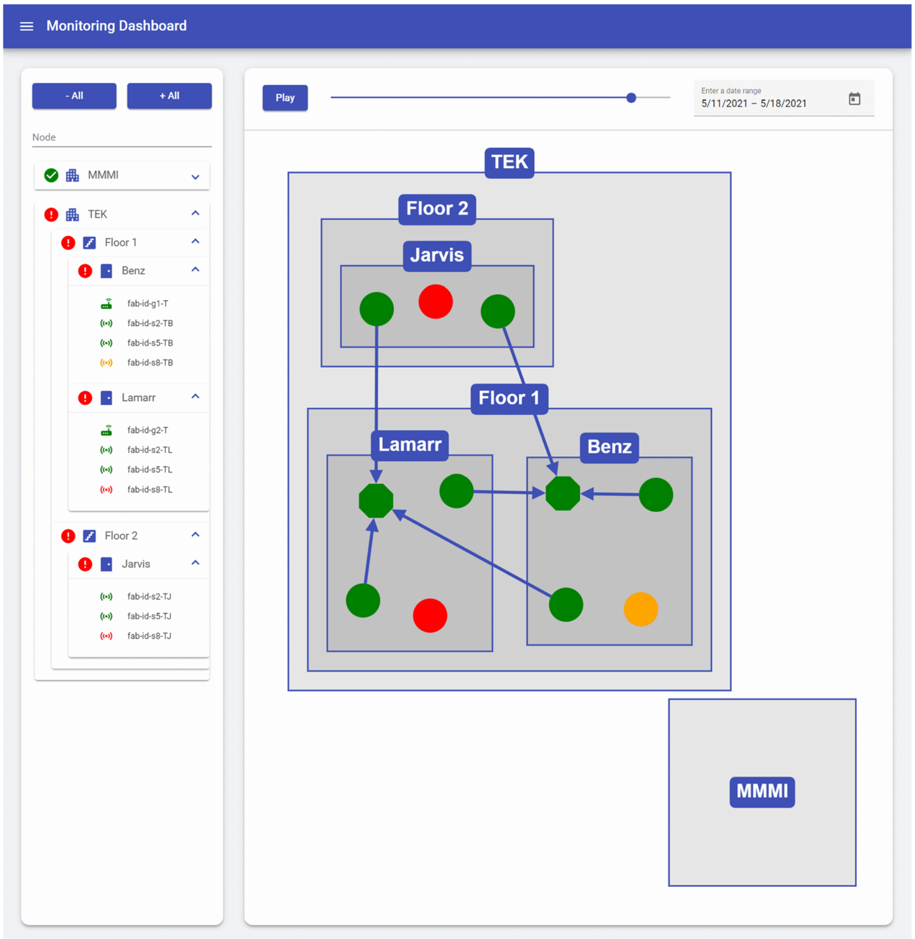


Fig. 8 Pairing strategy pattern



Enter a date range
5/11/2021 – 5/18/2021

Fig. 9 The network monitoring dashboard

The table is synced with the graph and vice versa. Expanding or collapsing any building, floor, or room is reflected in both the table and the graph. This feature enforces consistency between the two views and results in a clean user experience. A key feature of the web interface is replayability; the possibility of displaying the historic network changes that the system has undergone. Examples of network changes are device location changes, online status changes, device pairing changes, and so forth. Replayability allows the user of the web interface to learn about the network's behavior over time and if necessary react proactively to mitigate issues that might arise in the future. Furthermore, being able to review historic changes can aid in troubleshooting system faults based on this information that would not been available otherwise.

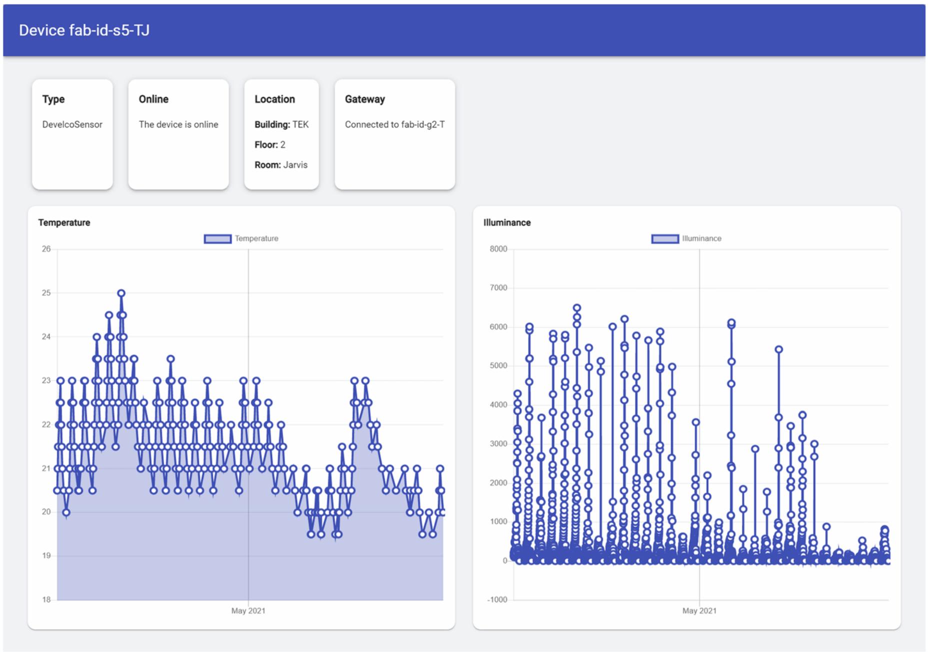
The purpose of the sensors is to record and report data about the environment they observe. This data is also available via the web interface. Each entry/node in either the tabular view or the graph view is clickable. Clicking on a device brings up a popover with data specific to that given device. An example of this can be seen in Fig. 10. The current version of the web interface displays the device type, online status, location, and the gateway it is connected to if the device is a sensor.

This information is displayed in the boxes positioned at the top. The data reported by the sensors is made available through a number of graph views depending on the number of telemetry categories as well as what data is reported by the sensors. The visualized data for this sensor is from a building at SDU and showcases the temperature changes and illuminance changes over the given time span.

Conclusion

This article has investigated an IoT network auto-configuration platform for building energy management. To this end, it was proposed a system architecture composed of three layers: 1) Edge components, 2) Backend and 3) Applications. In order to handle metadata related to location and devices, dynamic QR codes are proposed and link to a database. Finally, several automatic paring strategies are analyzed and linked with the concept of autonomic computing to provide the necessary automation for cost-effective energy management in medium and large-sized buildings.

The proposed system was evaluated in a real medium-sized building case study. The strategy was adopted showing a trade-off between optimal configuration and total configuration time. It is clear that advanced strategies need to be further analyzed to obtain faster solutions without compromising the quality of the communication signal strength. The severity of time constraints can drastically increase with the size and complexity of the buildings. The authors believe that is important in the future to store historical paring information to produce faster responses. Furthermore, a temporal network graph was explored to process the historical context of devices as well as the physical context. The result is an interface with replayability of configuration changes during the system's lifespan.



Device fab-id-s5-TJ

Type
DevelcoSensor

Online
The device is online

Location
Building: TEK
Floor: 2
Room: Jarvis

Gateway
Connected to fab-id-g2-T

Fig. 10 A data visualization example for a device

Finally, the contribution of this study goes beyond pointing at which pairing strategy is superior. Instead, we enrich the debate of autonomic computing and how it can contribute to automating different tasks in building energy management, including automating the configuration of IoT devices. We believe that with the spreading of wireless IoT solutions, such methods will become more and more discussed.

Abbreviations

IoT: Internet of Things; BACS: Building Automation and Control Systems; BMS: Building Management System; EU: Europe Union; FACUS: Framework for Adaptive Collaborative Ubiquitous Systems; GHG: Greenhouse gases; QR: code Quick Response code; SDN: Software Defined Networking; URL: Uniform Resource Locator; MQTT: Message Queuing Telemetry Transport; OPC UA: OPC Unified Architecture; HTTP: HyperText Transfer Protocol; MMMI: Maersk Mc-Kinney Moller Institute; REST: Representational State Transfer; API: Application Programming Interface; FCFS: First come, first served; PCS: Physical constraint strategy

Acknowledgments

NA.

About this supplement

This article has been published as part of Energy Informatics Volume 4, Supplement 2 2021: Proceedings of the Energy Informatics.Academy Conference Asia 2021. The full contents of the supplement are available at https://energyinformatics.springeropen.com/articles/supplements/volume-4-supplement-2.

Authors' contributions

SSM and AQS: Writing – review and editing. BNJ: Investigation, supervision, funding acquisition. All authors read and approved the final manuscript.

Funding

This work is supported by the "Cost-effective large-scale IoT solutions for energy efficient medium- and large-sized buildings" project, funded by the Danish Energy Agency under the Energy Technology Development and Demonstration Program, ID number: 64020–2108.

Availability of data and materials

NA.

Declarations

Ethics approval and consent to participate

NA.

Consent for publication

NA.

Competing interests

The authors declare that they have no competing interests.

Published: 24 September 2021

References

Agoulmine, N., Balasubramaniam, S., Botvitch, D., Strassner, J., Lehtihet, E., Donnelly, W.: Challenges for Autonomic Network Management, (2006). https://repository.wit.ie/744/1/MACE2006-final.pdf. [Online: Accessed 1 Apr 2021]

Alaya MB, Matoussi S, Monteil T, Drira K (2012) Autonomic computing system for self-management of machine-to-machine networks. In: Proceedings of the 2012 International Workshop on Self-Aware Internet of Things. Self-IoT, vol 12. Association for Computing Machinery, New York, pp 25–30. https://doi.org/10.1145/2378023.2378029

Alaya MB, Monteil T (2015) Frameself: an ontology-based framework for the self-management of machine-to-machine systems. Concurr Comput Pract Exper 27(6):1412–1426. https://doi.org/10.1002/cpe.3168

Amasyali K, El-Gohary NM (2018) A review of data-driven building energy consumption prediction studies. Renew Sust Energ Rev 81:1192–1205. https://doi.org/10.1016/j.rser.2017.04.095

Ashraf QM, Habaebi MH (2015) Introducing autonomy in internet of things, recent advances in computer science, pp 215–221, ISBN 9781618042972

Billanes JD, Ma Z, Jørgensen BN (2017) Consumer central energy flexibility in office buildings. J Energy Power Eng 11(10): 621–630. https://doi.org/10.17265/1934-8975/2017.10.001

Billanes JD, Ma Z, Jørgensen BN (2018) The bright green hospitals case studies of hospitals' energy efficiency and flexibility in philippines. In: 2018 8th International Conference on Power and Energy Systems (ICPES), pp 190–195. https://doi.org/10.1109/ICPESYS.2018.8626916

Bouassida Rodriguez I, Sancho G, Villemur T, Tazi S, Drira K (2009) A model-driven adaptive approach for collaborative ubiquitous systems. In: Proceedings of the 3rd Workshop on Agent-Oriented Software Engineering Challenges for Ubiquitous and Pervasive Computing. Association for Computing Machinery, New York, pp 15–20. https://doi.org/10.1145/1568181.1568187

Bracke, S.A.L.: TR69 based service interface for OSGi bundles (2007). https://patentimages.storage.googleapis.com/88/61/71/886f2023561a00/EP1835690A1.pdf. [Online: Accessed 1 Apr 2021]

Cisco: Cisco Smart Building Solutions (2019). https://www.cisco.com/c/en/us/solutions/smart-building.html_related-products. [Online: Accessed 1 Apr 2021]

Claudel B, De Palma N, Lachaize R, Hagimont D (2006) Self-protection for distributed component-based applications. In: Datta AK, Gradinariu M (eds) Stabilization, safety, and security of distributed systems. Springer, Berlin, pp 184–198. https://doi.org/10.1007/978-3-540-49823-0_13

Columbus, L.: IoT Market Predicted To Double By 2021, Reaching 520B dollars (2018). https://www.forbes.com/sites/louiscolumbus/2018/08/16/iot-market-predicted-to-double-by-2021-reaching-520b/?sh=641f54a61f94. [Online: Accessed 1 Apr 2021]

Comission E (2016) Building consumption by energy. Energy https://ec.europa.eu//energy//content//building-consumption-energyen. [Online: Accessed 1 Apr 2021]

Dalke RA, Holloway CL, McKenna P, Johansson M, Ali AS (2000) Effects of reinforced concrete structures on RF communications. IEEE Transact Electromagnet Compatibil 42(4):486–496. https://doi.org/10.1109/15.902318

Dobson S, Denazis S, Fernandez A, Gaiti D, Gelenbe E, Massacci F, Nixon P, Saffre F, Schmidt N, Zambonelli F (2006) A survey of autonomic communications. ACM Trans Auton Adapt Syst 1(2):223–259. https://doi.org/10.1145/1186778.1186782

Dzulkifly S, Aris H, Jorgensen BN, Santos AQ (2020) Methodology for a large scale building internet of things retrofit. In: 2020 8th International Conference on Information Technology and Multimedia (ICIMU), pp 62–67. https://doi.org/10.1109/ICIMU49871.2020.9243304

EMQX: EMQ X Github (2021). https://github.com/emqx/emqx. [Online: Accessed 1 Apr 2021]

Familiar MS, Martinez JF, Corredor I, Garcia-Rubio C (2012) Building service-oriented smart infrastructures over wireless ad hoc sensor networks: A middleware perspective. Comp Netw 56(4):1303–1328. https://doi.org/10.1016/j.comnet.2011.12.005

Ferreira M, Almeida M, Rodrigues A (2016) Cost-optimal energy efficiency levels are the first step in achieving cost effective renovation in residential buildings with a nearly-zero energy target. Energy Build 133:724–737. https://doi.org/10.1016/j.enbuild.2016.10.017

Forum, B.: CPE WAN Management Protocol Issue: 1 Amendment 6 Approval Date: March 2018 CWMP Version: 1.4, (2018). https://www.broadband-forum.org/technical/download/TR-069Amendment − 6.pdf. [Online: Accessed 1 Apr 2021]

Hidayat T, Azzery Y, Mahardiko R (2020) Load balancing network by using round robin algorithm: A systematic literature review. J Online Inform 4. https://doi.org/10.15575/join.v4i2.446

Hsieh W, Hsieh W, Chen J, Lin C (2016) Self-configuration and smart binding control on iot applications. In: 2016 18th International Conference on Advanced Communication Technology (ICACT), pp 80–85. https://doi.org/10.1109/ICACT.2016.7423282

Huang W, Hu J, Lin S, Liu T, Tsai P, Yang C, Yeh FI, Chen JH, Mambretti JJ (2012) Design and implementation of an automatic network topology discovery system for the future internet across different domains. In: 2012 26th International Conference on Advanced Information Networking and Applications Workshops, pp 903–908. https://doi.org/10.1109/WAINA.2012.77

Huebscher MC, McCann JA (2008) A survey of autonomic computing—degrees, models, and applications. ACM Comput Surv 40(3). https://doi.org/10.1145/1380584.1380585

Husz AK, Godor G, Farkas K (2012) Investigation of wlan access point placement for indoor positioning, pp 350–361. https://doi.org/10.1007/978-3-642-32808-432

IBM (2005) An Architectural Blueprint for Autonomic Computing. Autonom Comput 1:2886–2892 https://www-03.ibm.com/autonomic/pdfs/AC Blueprint White Paper V7.pdf. [Online: Accessed 1 Apr 2021]

ISI, ISE, IREES, EEG, TEP: Mapping and Analyses of the Current and Future (2020–2030) Heating/cooling Fuel Deployment (fossil/renewables) Executive Summary Final Report, (2016). https://ec.europa.eu/energy/sites/ener/files/documents/mapping-hc-excecutivesummary.pdf. [Online: Accessed 1 Apr 2021]

Jørgensen BN, Kjærgaard MB, Lazarova-Molnar S, Shaker HR, Veje CT (2015) Challenge: Advancing energy informatics to enable assessable improvements of energy performance in buildings. In: Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems. e-Energy, vol 15. Association for Computing Machinery, New York, pp 77–82. https://doi.org/10.1145/2768510.2770935

Jradi M, Sangogboye FC, Mattera CG, Kjærgaard MB, Veje C, Jørgensen BN (2017b) A world class energy efficient university building by danish 2020 standards. Energy Proc 132:21–26. https://doi.org/10.1016/j.egypro.2017.09.625 11th Nordic Symposium on Building Physics, NSB2017, 11–14 June 2017, Trondheim, Norway

Jradi M, Veje C, Jørgensen BN (2017a) Deep energy renovation of the mærsk office building in Denmark using a holistic design approach. Energy Build 151:306–319. https://doi.org/10.1016/j.enbuild.2017.06.047

Kephart JO Research challenges of autonomic computing. In: Proceedings. 27th international conference on software engineering, vol 2005. ICSE, pp 15–22. https://doi.org/10.1109/ICSE.2005.1553533

Kephart JO, Chess DM (2003) The vision of autonomic computing. Computer 36(1):41–50. https://doi.org/10.1109/MC.2003.1160055

Khalid A, Haye MA, Khan MJ, Shamail S (2009) Survey of frameworks, architectures and techniques in autonomic computing. In: 2009 Fifth International Conference on Autonomic and Autonomous Systems, pp 220–225. https://doi.org/10.1109/ICAS.2009.38

Khan MJ, Awais MM, Shamail S (2007) Achieving self-configuration capability in autonomic systems using case-based reasoning with a new similarity measure. In: Huang D-S, Heutte L, Loog M (eds) Advanced intelligent computing theories and applications. With aspects of contemporary intelligent computing techniques. Springer, Berlin, pp 97–106. https://doi.org/10.1007/978-3-540-74282-1_12

Ma Z, Badi A, Jorgensen BN (2016) Market opportunities and barriers for smart buildings. In: 2016 IEEE Green Energy and Systems Conference (IGSEC), pp 1–6. https://doi.org/10.1109/IGESC.2016.7790078

Ma Z, Billanes JD, Jørgensen BN (2017) A business ecosystem driven market analysis: The bright green building market potential. In: 2017 IEEE Technology Engineering Management Conference (TEMSCON), pp 79–85. https://doi.org/10.1109/TEMSCON.2017.7998358

Ma Z, Billanes JD, Jørgensen BN (2018) An assessment for energy management in retail stores. J Energy Power Eng 12(5):231–243. https://doi.org/10.17265/1934-8975/2018.05.002

Ma Z, Jørgensen BN (2018) A discussion of building automation and stakeholder engagement for the readiness of energy flexible buildings. Energy Inform 1(1):54. https://doi.org/10.1186/s42162-018-0061-z

Movahedi Z, Ayari M, Langar R, Pujolle G (2012) A survey of autonomic network architectures and evaluation criteria. IEEE Commun Surveys Tutorials 14(2):464–490. https://doi.org/10.1109/SURV.2011.042711.00078

Nami MR, Bertels K (2007) A survey of autonomic computing systems. In: Third International Conference on Autonomic and Autonomous Systems (ICAS'07), p 26. https://doi.org/10.1109/CONIELECOMP.2007.48

Nordbotten NA (2009) Xml and web services security standards. IEEE Commun Surveys Tutorials 11(3):4–21. https://doi.org/10.1109/SURV.2009.090302

Papageorgiou A, Zahn M, Kovacs E (2014) Efficient auto-configuration of energy-related parameters in cloud-based iot platforms. In: 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), pp 236–241. https://doi.org/10.1109/CloudNet.2014.6968998

Parashar M, Hariri S (2005) Autonomic computing: an overview. In: Banˆatre J-P, Fradet P, Giavitto J-L, Michel O (eds) Unconventional programming paradigms. Springer, Berlin, pp 257–269. https://doi.org/10.1007/11527800_20

Perera C, Jayaraman PP, Zaslavsky A, Christen P, Georgakopoulos D (2014a) Context-Aware Dynamic Discovery and Configuration of 'Things' in Smart Environments, p 546. https://doi.org/10.1007/978-3-319-05029-49

Perera C, Jayaraman PP, Zaslavsky A, Christen P, Georgakopoulos D (2014b) Mosden: An internet of things middleware for resource constrained mobile devices. In: 2014 47th Hawaii International Conference on System Sciences, pp 1053–1062. https://doi.org/10.1109/HICSS.2014.137

Perera C, Zaslavsky A, Christen P, Compton M, Georgakopoulos D (2013) Context-aware sensor search, selection and ranking model for internet of things middleware. In: 2013 IEEE 14th International Conference on Mobile Data Management, vol 1, pp 314–322. https://doi.org/10.1109/MDM.2013.46

Pikas E, Kurnitski J, Liias R, Thalfeldt M (2015) Quantification of economic benefits of renovation of apartment buildings as a basis for cost optimal 2030 energy efficiency strategies. Energy Build 86:151–160. https://doi.org/10.1016/j.enbuild.2014.10.004

Rodriguez Larrad I, Nguyen HC, Jørgensen NTK, Sørensen TB, Mogensen P (2014) Radio propagation into modern buildings: attenuation measurements in the range from 800 MHz to 18 GHz. In: Vehicular technology conference (VTC fall), 2014 IEEE 80th, pp 1–5. https://doi.org/10.1109/VTCFall.2014.6966147

Sancho G, Tazi S, Villemur T (2008) A semantic-driven auto-adaptive architecture for collaborative ubiquitous systems. In: Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology. Association for Computing Machinery, New York, pp 650–655. https://doi.org/10.1145/1456223.1456354 [Online: Accessed 1 Apr 2021]

SIEMENS (2019) IoT Solutions. Smart Buildings, USA https://new.siemens.com/us/en/products/buildingtechnologies/smart-buildings/iot-smart-spaces.html. [Online: Accessed 1 Apr 2021]

Simmons B, Lutfiyya H (2005) Policies, grids and autonomic computing. SIGSOFT Softw Eng Notes 30(4):1–5. https://doi.org/10.1145/1082983.1083081

Sterritt R, Parashar M, Tianfield H, Unland R (2005) A concise introduction to autonomic computing. Adv Eng Inform 19(3): 181–187. https://doi.org/10.1016/j.aei.2005.05.012 Autonomic Computing

Tahir M, Mamoon Ashraf Q, Dabbagh M (2019) Towards enabling autonomic computing in iot ecosystem. In: 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing. Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp 646–651. https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00122

Team, TAP: The cost of cooling down (2018). https://theaseanpost.com/article/cost-cooling-down. [Online: Accessed 1 Apr 2021]

Tucic M, Pavlovic R, Papp I, Saric D (2014) Networking layer for unifying distributed smart home entities. In: 2014 22nd Telecommunications Forum Telfor (TELFOR), pp 368–371. https://doi.org/10.1109/TELFOR.2014.7034426

Vavra, C.: Six IoT implementation challenges and solutions (2018). https://www.controleng.com/articles/six-iot-implementation-challenges-and-solutions/

WebNMS (2019) IoT Energy Management Solution. Energy Management System. WebNMS https://www.webnms.com/iot/energy-management.html. [Online: Accessed 1 Apr 2021]

Zhao Z, Schiller E, Kalogeiton E, Braun T, Stiller B, Garip MT, Joy J, Gerla M, Akhtar N, Matta I (2017) Autonomic communications in software-driven networks. IEEE J Select Areas Commun 35(11):2431–2445. https://doi.org/10.1109/JSAC.2017.2760354

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.