A SDN-based IoT Architecture Framework for Efficient Energy Management in Smart Buildings

Akram Hakiri*, Bassem Sellami†, Sadok Ben Yahia†, and Pascal Berthou‡
*University of Carthage, SYSCOM ENIT, ISSAT Mateur, Tunisia.
†University of Tunis El Manar, Faculty of Sciences, Dept of Computer Sciences.
‡CNRS, LAAS, UPS, 7 Avenue du colonel Roche, F-31400 Toulouse, France.
Corresponding author: akram.hakiri@enit.utm.tn

***Abstract*—Energy consumption has increased drastically at global scale due to the growing urbanization in cities. Energy efficiency in smart buildings can be achieved by introducing a context-aware Internet of Things (IoT) approach, where sensors can learn from their surrounding environment to control the actuators in a coordinated network. However, the IoT network requirements are constantly changing in unpredictable fashion, which needs faster and frequent on-demand network reconfiguration. Software Defined Network (SDN) has been envisioned as a new approach to enable a flexible and agile network programmability in diverse IoT scenarios. However, the focus has primarily been on the design of the SDN computation logic, i.e. controllers, while the dynamic delivery and operations service-inferred IoT resource allocation has been postponed.**

To fill this gap, this paper proposes a comprehensive architectural design that is devised to empower SDN-enabled Context-Aware IoT systems and networks to create efficient energy management in smart buildings. We investigate the provision of NFV IoT functions to support distributed automation and orchestration on IoT devices, and we present a context-aware approach to gather, filter and process data from sensing data in campus buildings. We provide a proof of concept to demonstrate successful deployment and provisioning of virtualized services in the context of Smart Campus research project.

***Index Terms*—Energy efficiency, Smart Building, Software Defined Networking; Internet of Things; Context-Awareness; Service Function Chaining.**

I. Introduction

Buildings account for over one-third the global final energy consumption [1] [2] and nearly 40% of total direct and indirect CO2 emissions [3] in many countries [4]. Energy demand from buildings continues to rise, driven by improved access to energy in developing countries, greater ownership and use of energy-consuming devices, and rapid growth in global buildings floor area. In particular, most university campuses hold their own distributed energy resources that are reconfigured to operate as micro-grid. Academic campuses are great contributors to energy consumption as lighting represents 31% and space heating accounts for 28% of total energy use [5]. Consequently, governments and industry are exploring new research directions and utility-driven energy improvement programs to drive advances in energy-efficiency. They promote the best policy practices motivating the awareness of society to become energy conscious and adopt energy conservation and energy efficiency measures. They also leverage diverse Internet of Things (IoT) technologies such smart meters, embedded sensing and networking at various levels of power micro-grids to achieve energy saving and information sharing among sensing devices and actuators.

Despite the promise, recent behavioral studies [6] have shown that it is still a challenging issue to incentive human behavior from an energy perspective [7]. For example, in campus housing and residential education buildings, which represent 9% of operating micro-grids [8], users are often unaware that changing their daily routines (e.g. doing laundry by night or setting at maintenance temperature their Heating Ventilation and Air-Conditioning (HVAC) equipment's, turning off unnecessary lighting and projectors, etc.) could impact their energy consumption patterns. Consequently, achieving energy efficiency goals despite the challenges of changing users habits becomes a key challenging issue.

Additionally, power micro-grids generate enormous amounts of raw data from different zones in buildings. The heterogeneous and dynamic aspects of IoT sensors generating these raw data pose major challenges for the underlying network by requiring support for handling heterogeneity, dynamic changes, device discovery as well as context-awareness. Existing standardized communication protocols [9], such as IEC 61850, IEEE P1547.8, and Modbus, are designed in isolation to solve a specific problem and are often retrofitted to address a new requirement. They often lack the right abstractions that address the interoperability requirements of IoT communication. Thus, if network resource utilization is a concern, the network must be flexible enough to be reprogrammed in accordance with any change in IoT application needs. Current network provisioning approaches neither address the dynamicity of IoT applications nor care about resource utilization. Therefore, in order to implement an efficient energy management system, collected data should be reasoned to unearth knowledge and reflect a meaningful context [10]. Besides, aggregating heterogeneous sensory data from different types of sources needs an agile infrastructure that embraces message brokers, context-aware framework, virtualization and softwarization for flexible, cost-effective, secure, and privacy-preserving IoT deployment in smart micro-grids.

978-1-7281-8241-4/20/$31.00 ©2020 IEEE

To address these key challenges, Software Defined Network (SDN) [11] and Network Function Virtualization (NFV) [12] show a significant promise in meeting micro-grids needs [13]. SDN can achieve fine-grained resource management, enforce traffic forwarding policies and keep the micro-grid network overhead as simple and minimal as possible. SDN can also solve interoperability issues in smart micro-grids communication, as it can deal with heterogeneous devices exchanging data formats and diverse protocols for M2M data exchange. It can also improve cooperation and capability mismatch between IoT devices to handle simultaneous connections of various communication technologies. The combination of SDN and NFV allows increasing the efficiency and capacity of smart micro-grid networks without radically making change at the hardware level.

In this paper, we introduce an intuitive system, less computation intensive, easy to implement on small modular low-cost Single Board Computer like Raspberry Pi, and amenable to online adaptation to the variations in ambient temperature, solar heat input through windows, etc. First, We present a novel IoT network virtualization approach based on SDN/NFV to offer a high degree of automation in service chaining delivery for IoT devices. Second, We introduce an IoT data model that provides a data collection facility to accommodate HVAC sensors and actuators. Third, we define a powerful approach to allow smart devices understanding their own context and adapt themselves "on-the-fly" for optimal energy-efficiency and performance under all communication scenario. Finally, we present a prototype implementation to show the proposed architecture framework can be deployed in smart campus buildings using low-cost hardware and lightweight Docker virtualization.

The remainder of this paper is organized as follows: Section II describes the related work on context-aware energy management IoT systems and highlights the use of SDN for context IoT data delivery. Section III devolves into the proposed architecture, which integrates our context-aware algorithm into a SDN infrastructure to realize efficient energy management in the network edge. Section IV presents the details of a proof-of-concept implementation along with two use case which will benefit from the proposed architecture. Section V presents concluding remarks alluding to lessons learned and future work.

II. Related work

Service oriented middleware has been introduced for integrating heterogeneous IoT domain context data into a unified context broker. For example, a context aware sensing approach for efficient energy communication in IoT is described in [14] to allow IoT devices learning contextual information, e.g. channel conditions, QoS demand, battery condition, etc. from their own experience and adapt themselves "on-the-fly" for optimal energy efficiency. Additionally, a semantic reasoning system based on Semantic Web technologies was proposed [15] [16] to facilitate interoperability among diverse IoT applications operating in a realistic context-aware environment. Likewise, [17] introduced a Semantic Web server for logging temperature values and store the gathered raw data in a time-series database.

Furthermore, SDN has been used as a context-aware network service layer to improve energy savings. For example, Rahman et al. [18] introduced the DistBlockBuilding framework that combines SDN and blockchain to improve energy consumption in smart building system. Du et al. [19] extended the OpenFlow model to IoT gateways at the network edge to attach contextual information to OpenFlow packet header. Kathiravelu et al. [20] introduced the Cassowary middleware based on AMQP as a northbound binding to integrate sensing devices into the SDN network seamlessly. An AMQP broker extracts features available from sensing devices such as motion, temperature or humidity detection and uses a context awareness reasoning model to filter and process the collected raw data.

Nonetheless, these contributions lack of a precise context reasoning model since the context is not well-defined or vaguely introduced. Our approach provides an efficient data aggregation and data query to enable centralized topology view that can be leveraged by multiple data-driven IoT applications. Our approach leverages NFV and SDN to virtualize the underlying smart micro-grid network by deploying IoT functions into software packages, assemble and chain them whenever required to deliver chained services to IoT devices. It also makes use of machine learning and data analytics for context-aware data processing, filtering, aggregation, and mining to capture the knowledge and generate high-level abstracted context information. Thus, our approach can be used to react and trigger specific events to change the behavior of IoT devices, e.g. automating and adapting the indoor lighting, closing curtains, switching off the light, ensure the comfort level by controlling the HVAC equipment's, etc.

III. Design and Implementation of The proposed solution

This section presents the architectural details of our proposed solution to realize our SDN-enabled efficient energy management system.

A. Architectural Overview

the architecture of our framework is composed of three layers: at the bottom, is the *perception layer*, which comprises the sensing layer and the aggregator. The sensing layer encompasses all smart IoT sensors, e.g. temperature, humidity, air quality sensors, and the HVAC actuators, which have direct connectivity to the network via short range PAN technologies such RF, Bluetooth, ZigBee, etc. Aggregator sinks collect data from sensing layer and act as bridges between the reste of sensing nodes and IoT gateways, which act as a network proxy with the rest of the network.

Next, we have the *fog layer*, which is located in single hop proximity of the IoT sensing devices and include all the network equipment to realize the micro-grid communication infrastructure. It consists of IoT gateways (shown in Figure 1) that interface with the perception layer to receive raw data from the sensors, and send commands to control the actuators. The gateway concept is prevalent in home ADSL models and WiFi access points. The design of IoT gateway is different since it should be able to integrate heterogeneous smart objects and expose their resources and make them available to the rest of IoT network. Additionally, IoT gateways are connected to the SDN network through SDN routers (i.e. virtual and physical), which embed an OpenFlow agent that has the capabilities to add, remove, update, and delete packets inside these routing devices.

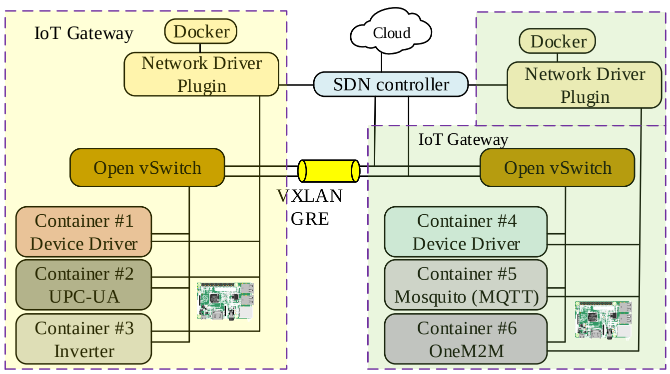

Fig. 1: IoT Gateways

The SDN routers (OpenVSwitch virtual routers in Figure 1) are connected to a *SDN control layer*, i.e. the SDN controller, which embeds all the intelligence and maintains the network-wide view of the data path elements and links that connect them. The controller contains several modules we develop to integrate to the smart micro-grid network. First, the *IoT function virtualization module* (will be described in Section III-B), which expands the micro-grids network capacity by deploying virtualized IoT functions into software packages that can be assembled and chained whenever required to deliver chained services to IoT devices. Thanks to NFV platform which can add new services without interrupting existing one or upgrading the network with new devices. The orchestrator is the NFV management and network orchestration (MANO) tool, which is responsible for controlling and managing NFV compute, storage, and network resources.

Second, the SDN control layer contains the IoT management model to perform communication with IoT sensors through IoT data management and service capability's module i.e. *IoT Data Export Plugin* for accomplishing M2M operations at scale. It also includes a *context agent* embeds the context-aware model to perform context reasoning needed for data processing, filtering, and aggregation, and to capture the knowledge and generate high-level abstracted context information. Third, the controller layer comprises a data processing module that make use of a Time Series Data Repository (TSDR) module to perform real-time data processing and analytics, data transformation and collection services: upon received at the data store, the IoT Data Export Plugin triggers CRUD handling operations to enable writing data in the data-store before connecting them to the context-aware middleware to perform data processing. The TSDR module connects to Cassandra and HBase NoSQL database management system through plugins.

B. IoT Service Chaining

The *IoT Function Virtualization (IoT NFV)* module shows our proposal for mapping IoT sensors into virtualized functions that follows the ETSI guideline for NFV architectural framework. The SDN controller is merged with the IoT management platform (i.e. NFV Platform) to perform communication between IoT gateways and their remote sensors and enforce the security of state information. The IoT-NFV module uses lightweight containers to enable the creation of multiple isolated virtual IoT gateways inside a single physical one. For example, VNF1 and VNF2 represent two independent virtualized functions chained to form a single IoT service. Constrained Application Protocol (CoAP) messages in VNF1 coming from sensors in the sensing layer are chained with DTLS service to enforce the security of IoT resources. Similarly, other group of sensors in the sensing layer that make use of HTTP/REST services in VNF1 can see their messages chained and secured with TLS in VNF2 to optimize the cooperation between IoT devices, intermediate infrastructure and the rest of the IoT network. The virtualization layer is based on lightweight containers using Docker. Thus, it becomes fast to create, install, run and deploy independent micro-services and provide simple service composition facilities.

Routing the packets among these VNF components is completely managed and controlled by the SDN controller. Thanks to Pipework and the "overlay" mode of OpenvSwitch software router. The former allows connecting together multiple containers in arbitrarily complex scenarios. The later provides a kind of private IP addresses that are only valid internally. Each IP address *P* identifies a service deployment in a separate chain, so that the SDN controller can program the flow table with the required flow entries F_P to define the following component $B = F_P(A)$ in the chain for which the traffic will be forwarded. The controller creates for each flow entry F_P the forward table entries that matches received packets against the forwarding ports A they should follow with P as the destination address.

C. Machine Learning Engine

The machine learning engine capabilities are twofold: first, they help the context reasoning framework to infer the context decisions. Second, they feed back the environment changes to the SDN controller to perform automatic traffic steering and policy placement.

For the former, the machine learning engine monitors current sensor's data delivery and predicts future data and learn the optimal policy for the network management. In particular, the energy demand is variable in time and space, since its consumption varies qualitatively and quantitatively on the time of days, e.g. working days, holidays, week-end, etc. or the location where IoT devices are deployed, e.g. laboratories, classrooms, office building, etc. The machine learning engine provides better personalized experience and give priorities to specific IoT devices that should communicate relevant information to the SDN controller. It also enables storing and processing the collected data to analyze the data-sets based on certain parameters such as location, time, and historical data.

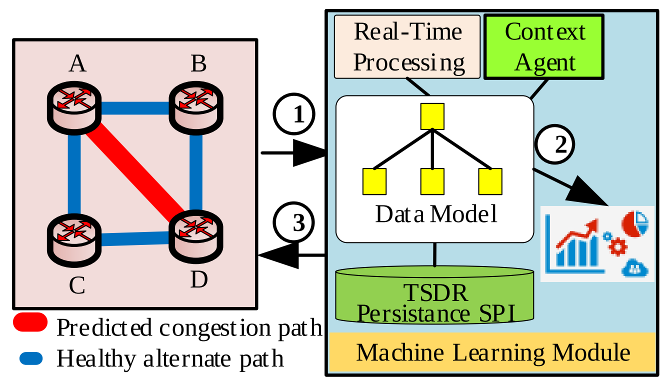

Fig. 2: Traffic congestion prediction with automated control.

For the later, Figure 2 depicts the machine learning module where the SDN controller continuously learns from data generated by virtual routers and becomes aware of the runtime status of the network. The controller collects OpenFlow statistics (circle 1), applies ML algorithm i.e. multi-layer perceptron (circle 2), and take the right decisions that adjust the policies (i.e. traffic flow redirection from $A \Rightarrow D$ to $A \Rightarrow B \Rightarrow D$ or $A \Rightarrow C \Rightarrow D$) for traffic classification and traffic shaping, dynamically change these policies according to the analytics results, and feed back these results to the forwarding path for automatic steering and policy placement. The controller can also use the machine learning capabilities to establish normalized profiles to predict traffic pattern, and perform routing optimization based on predetermined or dynamically update learned rules.

IV. Proof of Concept Implementation

This section describes the primary prototype implementation we realized to verify the feasibility of running the platform. We first introduce the platform implementation, then we highlight two application test cases.

A. Platform

IoT gateways are based on a single-board computer RPi 3 with 1 GB of memory and a quad-core ARMv8 BCM2837B0 Cortex-A53 ARM Cortex-A53 CPU running at 1.2 GHz. The gateway connects to the network using its integrated 2.4GHz 802.11n interface. It also contains *hostapd* user space daemon software we used to create virtual wireless networks inside the same physical one.

We deployed several sensor boards that act as Cluster Head (CH) nodes to collect raw sensor data and measure the surrounding air quality from all Cluster Members (CM). CMs sensor nodes periodically measures temperature, humidity and Co2 levels, compare them with the last measurement, if results had changed, they send an advertisement message (ADV) to the cluster head. Examples of CMs we used include *DHT22* temperature and humidity sensors; the *MQ-135* Co2-Gas sensor; and the *K30 CO2* module that gathers the level of oxygen inside our campus buildings and student housing residence. We also used smart energy meter to detect and report power consumption to CH nodes and IoT gateways. As Cluster Heads (CH) we used multiple NodeMCU IoT platforms running on-top of the ESP8266 Wi-Fi SoC, which also integrate a TCP/IP protocol stack that allows access to our virtual Wi-Fi network running on IoT gateways. We successfully connected these CHs through the MQTT broker running on IoT gateways. We used also CoAP API to connect some other sensors to ESP8266 board. COAP servers are deployed inside IoT gateways in a form of lightweight containers.

B. Service Function Chain Composition in Education Building

For the education buildings, we consider diverse CH nodes in charge of collecting temperature, humidity levels and CO2 concentration from CM sensors of occupied indoor spaces. We consider three VNFs implemented inside docker images: $Tensor_{VNF}$, $Mosquitto_{VNF}$, and $OneM2M_{VNF}$ as described in Figure 3. The first VNF monitors the current sensor's data delivery, uses TensorFlow machine learning models to predict future data from correlated sensor's readings, and learn the optimal policy for the network management by avoiding redundant readings. The second VNF gathers sensors readings periodically, listen to network events through MQTT protocol, and send commands to control fans and HVAC system. Finally, the third VNF establish the access to IoT resources through the hierarchical containment M2M tree.

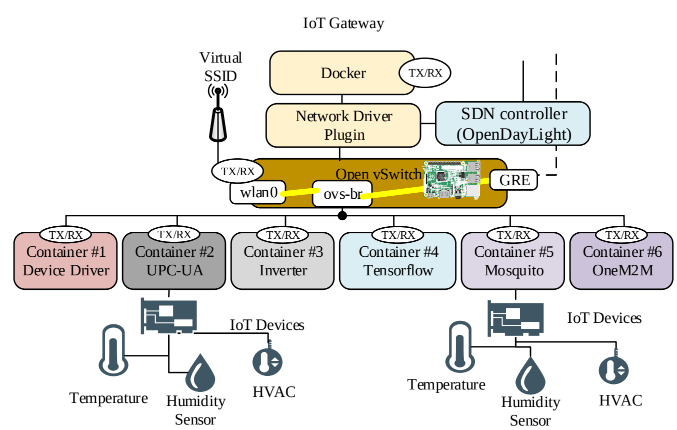

Fig. 3: Network topology used as the target application.

In order to interconnect different micro-services running in containers, we used Docker engine v19.03. Docker engine offers advanced network capabilities to manage connectivity between containers. As shown in Figure 3, we used Docker Network Driver Plugin to create virtual docker network that connects to virtual SDN routers (OpenvSwitch) and handles all coordination between virtual hosts. We also configured persistent docker volume to store alarms and other debugging events outside docker containers. We also used Vagrant for automating the creation of virtual machine instances, building and maintaining portable virtual Docker containers.

We deployed Docker swarm as our container's orchestration tool to manage different VNFs we deployed in our tests. Docker swarm offers a high level of availability for the running application. We defined one of our containers as a leader to manage membership and delegation among the other containers, which we configured as workers. It is worth noting that we can use Kubernetes (K3s) system for automating deployment, scaling, and management of VNFs. We configured Docker Swarm to use docker-compose configuration files and scripts we created to tell the docker daemon (running inside each IoT gateway) how to pull the appropriate container image from the Docker Hub repository, how to establish networking between VNFs, how to mount storage volumes, and where to store logs for a given container.

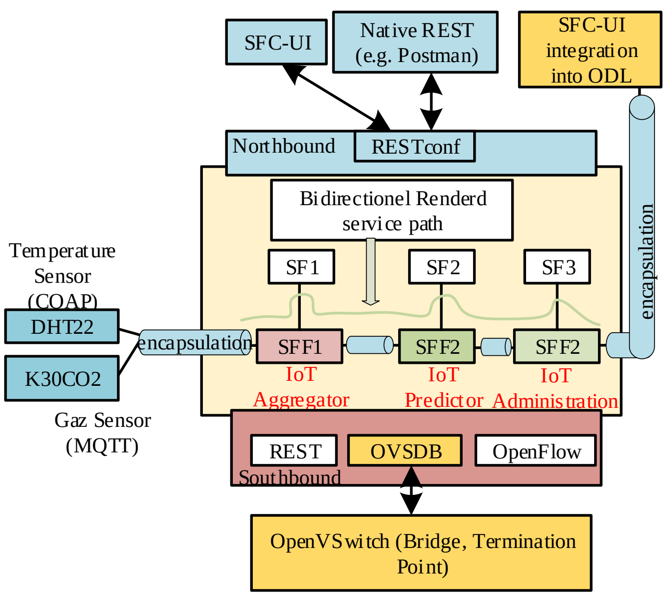

Fig. 4: SFC composition for Temperature and CO2 measurements as an IoT service.

Figure 4 describes the approach we used to configure the OpenDaylight SDN controller Service Function Chaining (SFC) to define an ordered list of network services (i.e. $Tensor_{VNF}$, $Mosquitto_{VNF}$, and $OneM2M_{VNF}$) which we stitched together to create a service chain. This SFC is arranged as: i) physical network function (PNF) that contains the IoT temperature and gaz sensors, ii) three IoT VNFs depicted as Service Function Forwarders (SFF) in Figure 4, and iii) the SFC-UI integration into the SDN controller. When the IoT SF1 is updated with new sensor measurements, an update SFC composition is triggered by the docker swarm orchestrator to pull and install a new image (or migrate an existing one) for the required VNF ($Mosquitto_{VNF}$, $Tensor_{VNF}$, or $OneM2M_{VNF}$), respectively.

Our implementation shows that our approach is able to create, instantiate and deploy new customized on-demand virtualized IoT services that gather, process, estimate and supervise the air conditioning inside campus buildings. Our approach successfully collects room temperature, send these values to IoT VNFs through IoT gateways, which forward them to the SDN controller to switch ON or OFF the HVAC appliances based on temperature and CO2 threshold. This result is very flexible and reconfigurable as it can instantiate and deploy VNFs as needed for the scalability and allows saving up to 70% of the energy consumption in the campus buildings.

C. Activity Management in Residential Building

Monitoring users activity in residential building is very critical to understand how they interact with IoT home appliances (e.g. TV boxes, PlayStation (PS), washing machine, Laptop, Lights, etc.) because different users activities usually call for different services. Therefore, we add an Activity Recognition (AR) model to the context-awareness model.

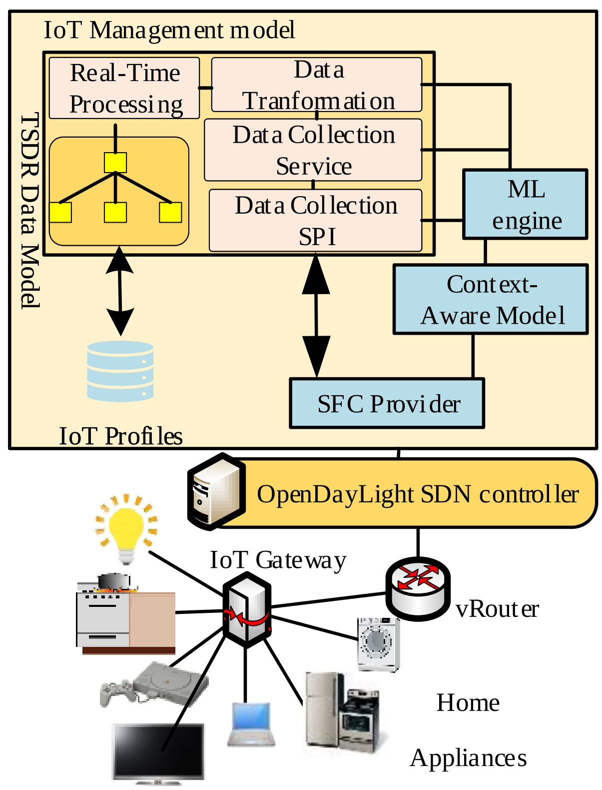

Fig. 5: Typical smart home network as an IoT service.

The AR model describes the user behavior inside smart residential home as shown in table I. The AR model is trained using semi-supervised learning model (which is included in the ML engine in Figure 5) to forecast the appliances a user is using during the activities describes in table I. For example, if a user activity is "*Sleep*", only night light of bedroom should be ON and all other appliances should be OFF. Similarly, if the weather is getting warmer, the HVAC or fans should be

TABLE I: Daily Occurring User Activity in smart residential building

Activity	Sleep	Use Laptop	Study in office	Watch TV	Read Book	Play PS	Prepare food	Washing clothes	No One at home
Occurrence %	33	10	8.3	12.5	6.8	7.1	9.2	4.3	8.8

ON automatically; if no one at home all lights and appliances should be OFF.

Figure 5 depicts the IoT gateway, which can be easily turned into a virtual SDN router using OpenvSwitch and Docker Network Driver Plugin services. The SDN controller can manage the whole Home Area network. The controller monitors link states periodically via the link layer discovery protocol and creates the network topology. Additionally, the approach we develop in this paper allows slicing the Home Area network into several separated logical networks, each network partition can deal with different QoS requirements. For example, a network partition with strict QoS requirements can be configured for CCTV Camera and video streaming, etc.

This result of architecture can successfully discover a lot of useful data from the AR model. It also success in creating several separated logical networks, which improve the network agility, availability and performance. Moreover, the proposed architecture performs energy saving by automatically switching ON/OFF unnecessary home appliances based on the context-awareness and AR models. The SDN controller collect useful data for context-aware service provisioning and manages the forwarding tables of SDN routers to provide comfort and assistance to building occupants, and apply powerful learning models on collected data to derive behaviors that impact high energy consumption.

V. Conclusion

In this paper, we proposed a secure, context-aware, and QoS enabled IoT architecture framework that leverages key enablers technologies such SDN, NFV and Machine Learning, to offer a flexible, efficient, and reconfigurable energy management in smart buildings. We introduced a IoT service chaining solution for creating and deploying customizable IoT services on-demand.

Our solution is designed with reuse in mind and can easily be adopted in other IoT themes. A wide adoption and acceptance of this idea will open up new opportunities for virtualizing IoT systems and applications.

Acknowledgments

This work was partially funded by the Tunisian Ministry of Higher Education and Scientific Research (MES) under the Young Researchers Incentive Program (19PEJC09-04) and C.V Raman program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of MES and C.V Raman.

References

[1] E. I. A. (EIA), "International energy outlook," September 2019.

[2] U. D. of Energy's, "Increasing efficiency of building systems and technologies," September 2015. [Online]. Available: https://bit.ly/2CodZgd

[3] E. E. Agency, "Progress on energy efficiency in europe," Feb 2019. [Online]. Available: https://bit.ly/2OygVJN

[4] I. E. Agency, "Tracking buildings 2020," 2020. [Online]. Available: https://www.iea.org/reports/tracking-buildings-2020

[5] M. Jafary, M. Wright, L. Shephard, J. Gomez, and R. U. Nair, "Understanding campus energy consumption – people, buildings and technology," in *2016 IEEE Green Technologies Conference (GreenTech)*, 2016, pp. 68–72.

[6] S. C. Staddon, C. Cycil, M. Goulden, C. Leygue, and A. Spence, "Intervening to change behaviour and save energy in the workplace: A systematic review of available evidence," *Energy Research & Social Science*, vol. 17, pp. 30 – 51, 2016.

[7] H. Rashid, P. M. Mammen, S. Singh, K. Ramamritham, P. Singh, and P. Shenoy, "Want to reduce energy consumption? don't depend on the consumers!" in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*, 2017.

[8] A. Hirsch, Y. Parag, and J. Guerrero, "Microgrids: A review of technologies, key drivers, and outstanding issues," *Renewable and Sustainable Energy Reviews*, vol. 90, pp. 402 – 411, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136403211830128X

[9] S. Kumar, S. Islam, and A. Jolfaei, *Microgrid communications - protocols and standards*, ser. Energy Engineering. Institution of Engineering and Technology, 2019, pp. 291–326.

[10] A. G. Kyselova, I. V. Verbitskyi, and G. D. Kyselov, "Context-aware framework for energy management system," in *2016 2nd International Conference on Intelligent Energy and Power Systems (IEPS)*, June 2016, pp. 1–4.

[11] S. Singh and R. K. Jha, "A survey on software defined networking: Architecture for next generation network," *Journal of Network and Systems Management*, vol. 25, no. 2, pp. 321–374, Sep 2016.

[12] T. Zhang, "Nfv platform design: A survey." *arXiv: Networking and Internet Architecture*, 2020.

[13] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Decentralized cloud-sdn architecture in smart grid: A dynamic pricing model," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1220–1231, 2018.

[14] S. Sen, "Invited: Context-aware energy-efficient communication for iot sensor nodes," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2016, pp. 1–6.

[15] A. I. Maarala, X. Su, and J. Riekki, "Semantic reasoning for context-aware internet of things applications," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 461–473, April 2017.

[16] A. Guner, K. Kurtel, and U. Celikkan, "A message broker based architecture for context aware iot application development," in *2017 International Conference on Computer Science and Engineering (UBMK)*, Oct 2017, pp. 233–238.

[17] N. Najem, D. B. Haddou, M. R. Abid, H. Darhmaoui, N. Krami, and O. Zytoune, "Context-aware wireless sensors for iot-centeric energy-efficient campuses," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, May 2017, pp. 1–6.

[18] A. Rahman, M. K. Nasir, Z. Rahman, A. Mosavi, S. Shahab, and B. Minaei-Bidgoli, "Distblockbuilding: A distributed blockchain-based sdn-iot network for smart building management," *IEEE Access*, pp. 1–1, 2020.

[19] P. Du, P. Putra, S. Yamamoto, and A. Nakao, "A context-aware iot architecture through software-defined data plane," in *2016 IEEE Region 10 Symposium (TENSYMP)*, May 2016, pp. 315–320.

[20] P. Kathiravelu, L. Sharifi, and L. Veiga, "Cassowary: Middleware platform for context-aware smart buildings with software-defined sensor networks," in *Proceedings of the 2Nd Workshop on Middleware for Context-Aware Applications in the IoT*, ser. M4IoT 2015, 2015, pp. 1–6.