Poster Abstract: SECY APP: Self Configuration and Easy Management for Software Defined Smart Homes

Gaia Maselli
Sapienza University of Rome - Italy
gaia.maselli@uniroma1.it

Mauro Piva
Sapienza University of Rome - Italy
mauro.piva@uniroma1.it

ABSTRACT

In this paper we address configuration and management issues of smart homes. Current platforms requires the user to deal with several management inconvenience problems, such as increasing devices, operating between devices, and using new devices. From a user perspective, system configuration and management are major issues: ordinary consumers want to use systems performing minimal configuration. To address this issue, we propose a platform, composed of a web application and Software Defined Network (SDN). While the user interacts with an easy-to-use interface on a smart device, the app automatically generates and installs SDN rules. Our platform, besides facilitating configuration and management, results more efficient — up to 4 times faster — and reliable — able to operate even in case of no connection with the cloud — than current solutions.

ACM Reference Format:
Gaia Maselli and Mauro Piva. 2018. Poster Abstract: SECY APP: Self Configuration and Easy Management for Software Defined Smart Homes. In *SenSys '18: Conference on Embedded Networked Sensor Systems, November 4–7, 2018, Shenzhen, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3274783.3275201

1 MOTIVATION

In the latest years, smart homes have caught increasing attention and witnessed rapid development. Smart Homes are outfitted with myriad of devices — sensors, actuators, smart devices, etc. — that are used to optimize energy consumption and improve the quality of life. Current approaches to smart-home management can be classified into two categories. The first includes *traditional* home-automation platforms (e.g., Kblue), where a smart hub, deployed inside the home, acts as server and manages all devices. The management logic is local to the home, and it is possible to interact with and configure the system from a remote smart device. The local and centralized approach is limiting as it represents a single point of failure. The second category includes *cloud-based* platforms (e.g., Alibaba Smart Living, Google Home, etc.), where the management logic is implemented in the cloud and a home local forwarding hub interfaces home devices with a remote server. This approach results cheaper and more powerful but if there is no connection to the cloud the system becomes unusable. More recently, a few proposals leveraged interesting developments in Wireless SDN [1] that distribute forwarding rules inside the environment. SDSH [2] is an example of SDN application in smarthomes. A centralized controller in the cloud keeps the rules to manage the resources based on user requirements. Local hubs or switches are deployed inside the home to bridge the smart home local network and the cloud controller. While these platforms have shown the ability to improve management efficiency, a major question that remains not addressed is how to easily and automatically configure SDN controllers and switches, without requiring significant management and configuration work from the user.

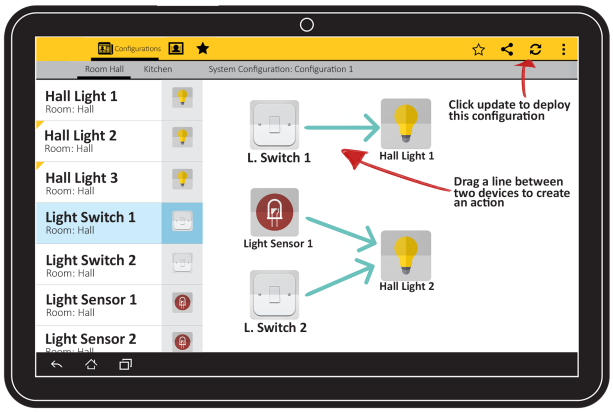

Figure 1: SECY app interface.

2 SECY APP

We propose an application for SElf Configuration and easY (SECY) management of smart homes based on SDN. The application requires minimal configuration from the user, asking to connect sensor devices — such as buttons, switches, remotes, joysticks, presence, temperature, humidity, wet, smoke, and light sensors — to actuator devices — lights, shutters, cooking appliances, TV, air conditioners, heating systems, game consoles, alarms, music devices, cleaning devices. Figure 1 shows how the user simply dragging a sensor — a wireless light switch — on the actuator — a smart light bulb — connects the two devices. In this way the user can configure all operation between devices. While the user connects devices, SECY app runs a engine for *automatic creation of SDN rules* (see Fig. 2). These rules are then fed to the SDN controller and installed on the SECY gateways, which are deployed inside the home. SECY gateways have a triple task: i) act as sinks of data from sensor devices deployed in the environment; ii) forward data based on SDN rules;

SenSys '18, November 4–7, 2018, Shenzhen, China

ACM ISBN 978-1-4503-5952-8/18/11…$15.00
https://doi.org/10.1145/3274783.3275201

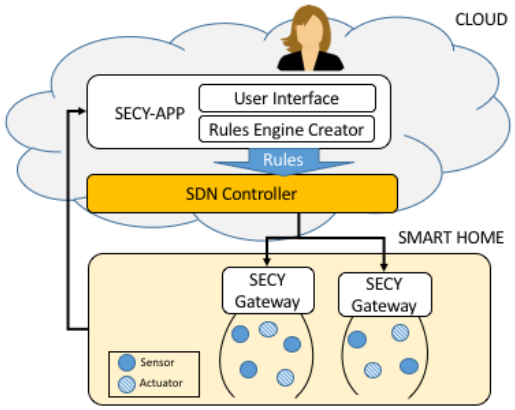

Figure 2: SECY Architecture

Table 1: Forwarding Rules for smart light examples.

#	Device	Src	Dst	NextHop	...	Action
1	L.Switch1	L.Switch1	Light1	SECYgat1	...	FORWARD
2	SECYgat1	L.Switch1	Light1	Light1	...	FORWARD
3	L.Switch2	L.Switch2	Light2	SECYgat1	...	FORWARD
4	SECYgat1	L.Switch2	Light2	Light2	...	FORWARD
5	L.Sensor1	L.Sensor1	Light2	SECYgat1	...	FORWARD
6	SECYgat1	L.Sensor1	Light2	Light2	...	FORWARD

Table 2: Action Rules for smart light examples.

#	Src	Dst	Action
1	L. Switch1	Light1	TOGGLELIGHT
2	L. Sensor1	Light2	SAVE(L. Sensor1)
3	L. Switch2	Light2	(GET(L.Sensor1)^BITMASK)^ TOGGLELIGHT

and iii) act as conventional routers for Internet traffic.
In a medium size environment, the user does not need a specialized router, like an OpenWRT one; we can exploit already available and fast spreading SDN routers.
How can the rule creator obtain SDN rules from a user drag? Our key idea is to use only two types of rules: i) *forwarding* rules, which are represented with a triple <source address, destination address and next hop address> and encode the conventional SDN FORWARD; and ii) *action* rules, which are represented with a triple <source address, destination address and action>, where the action corresponds to the action field in OpenFlow. Two types of rules are sufficient because all home operations can be divided into two groups: those involving a simple sensor-to-actuator command and those in which multiple sensors trigger an action on an actuator. Buttons, switches, remotes, joysticks, and some environmental sensors trigger simple actions on actuators. A button may switch on/off a light, another button can open/close shutters, a TV remote may switch on/off a TV or change channel, etc. These rules all belong to the sensor-to-actuator class. An example of more complex rule is given by an alarm, which may be fired by multiple presence sensors only when a button has already activated it (when inhabitants leave home). Instead, while someone allowed is inside the home, even if presence sensors are fired the alarm should not start.

Rules are parametric. The user's drag allows the rule creator engine to understand the type of rules to create and the values of its parameters. If the user drags a light switch on a smart light (see Fig. 1), then the rule creator defines a new set of forwarding rules, one for the sensor device (rule # 1 in table 1) and one for each SECY gateway along the path between the sensor and the actuator (rule # 2 in table 1). The rule creator generates also a configuration rule for the action on the actuator (rule # 1 in table 2). If instead the user drags a light sensor and a light switch on a smart light (see Fig. 1), then the rule creator defines a rule for the light switch (rule # 3 in table 1), a rule for the light sensor (rule # 5 in table 1), two rules for the each SECY gateway (rules # 4 and 6 in table 1), and two rules for the smart light (rules # 2 and 3 in table 2).

This approach of deriving networking rules directly from the application layer leads to a number of advantages for smart home users. First, network performances is highly improved: after setup, all data between any couple of home devices will flow directly from one to another, following the best available path and without unnecessary elaboration. If the user sets up a camera to stream the captured video to a local smart phone, with current solutions video data would be first sent to the cloud and then downloaded by the smart phone. With SECY, data flow is optimized, as with SDN packets from the camera are directly forwarded to the screen, with a consistent latency reduction. Moreover, working at the network level allows SECY to operate, and improve performances, of all the higher level protocols, e.g. MQTT. Second, distribution of rules mitigates the single point of failure problem of centralized architectures, which requires additional hardware for redundancy, in a simple and cost-free way. Third, with SECY we provide personalized Openflow Actions able to modify and forward received packets. In this way, we can change packets format, allowing for interoperability between devices that implement different protocols. Last but not least, privacy is improved: if a user sets up a rule for video streaming between a camera and a local screen, all SDN routers will forward the captured video only to the screen; data packets are kept local to the home and do not reach the Internet.

3 EVALUATION

From a quantitative perspective we evaluated SECY through Mininet simulations, and compared it to traditional and cloud-based solutions. The scenario considers a 3-floor home — with a SECY gateway for each floor — and a workload representing the behavior of a typical family of 4 people. Results show that SECY is 4 times faster than cloud-based and approaches, and 2 times faster than traditional systems. From a qualitative perspective we can observe that SECY presents: i) *reliability*, as the rules are installed inside the home, the system works even in case of no connection to the cloud; and ii) *high SDN compatibility*, as the system can be implemented with OpenFlow.

REFERENCES

[1] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo. 2015. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In *IEEE INFOCOM 2015*. 513–521.
[2] K. Xu, X. Wang, W. Wei, H. Song, and B. Mao. 2016. Toward software defined smart home. *IEEE Communications Magazine* 54, 5 (May 2016), 116–122.