SeSAMe: Software Defined Smart Home Alert Management System for Smart Communities

Rohit Abhishek[1], Shuai Zhao[1], David Tipper[2], Deep Medhi[1]
[1]University of Missouri-Kansas City, [2]University of Pittsburgh
{rabhishek, shuai.zhao, dmedhi}@umkc.edu, tipper@tele.pitt.edu

***Abstract*—Future Smart Cities require new ways to manage services that benefit the end users. An important issue is how to connect homes in a community and create an alert management system in Smart Cities with coordination among different entities. In this paper, we present an architectural vision of a software defined home alert management system for Smart Cities. This alert management system would make the residents aware of any incidents in the neighborhood such as fire. In this work, we use the features of software defined networking to design a manageable and flexible smart home for a smart community to provide services such as smart alarm systems.**

I. Introduction

Providing smart services in Smart Cities has become an emerging interest in the past few years. The goal is to improve and simplify the life of citizens living in a city by integrating information and communication technology (ICT) and Internet of Things (IoT) solutions to create better service facilities for the citizens [6]. There are six major components in a smart city as shown in Fig. 1. In this paper, we focus on smart homes—a major component of smart living in smart cities.

A smart home, in its core, can be defined as an automated home where all devices and appliances are networked together; they coordinate to make intelligent decisions and can be controlled remotely by the owner of the home. One of the important features in home automation is that the home owner decides the reaction of the device. Initially, smart homes were thought to be to controlled environmental systems, but recent development in technology has enabled it to cover almost any electrical device within the home [13]. That is, smart homes aim to improve the lives of the residents by (1) creating secure homes, (2) saving energy, and (3) improving home accessibility in a convenient and flexible way. A smart community can be viewed as a virtual environment where smart homes are networked together in a local geographic region to continuously monitor various aspects of the community and provide feedback to improve the safety, security, quality and emergency response abilities of the community [12].

In a smart home, different electronic devices are networked together through a Home Area Network (HAN). For example, motion sensors communicate with light sensors and thermal sensors to switch lights on/off and adjust the temperature in the home. The smart homes can study the resident's daily activities pattern and accordingly adapt to it.

The overall goal of our work is to enable features for a smart community where smart homes are able to communicate with each other and exchange information. This can be used to communicate important public safety information/alerts to each other, such as fire in any home or an amber alert.

Whenever an incident like a fire takes place in a home, a fire hazard severity zone can be formed around the place of the fire. If any home lies within the danger zone, it has a high probability of catching fire. During these situations, a major concern is to inform the people, who live in these danger zones, about the fire. Generally, to do this, the authorities send out alerts to the public in that zone and broadcast it on the news as well. However, this might not reach all the people who need to be informed.

In this work, we propose SeSAMe as an architectural vision for software defined smart community home alarm management based on software defined networks (SDN). We present the protocol messages and system components for the operation of SeSAMe. With our approach, should any alert/event such as a fire occur, an automated notification is sent to all the homes in the neighborhood and to the fire department and the police department about the fire. At the same time, alerts can also be forwarded to the police and the fire departments.

In future smart homes, we anticipate a number of sensors for monitoring a variety of information about the homes. We believe the use of SDN would give the flexibility of adjusting to old and newly added sensors and traffic that arises from them. An advantage of using SDN is that the configuration of highly complex sensor devices can be made easy through the centralized SDN controller.

The rest of the paper is organized as follows. Section II presents the related work followed by a brief overview of SDN in section III. Section IV presents our proposed architecture, section V describes the system initialization, and section VI shows our experimental result followed by conclusion and future work in section VII.

II. Related Work

Li et al. [12] introduces the smart community as a new application of the Internet of Things. In this work, the smart community architecture has been defined and how to realize secure and robust networking among individual homes has been described. Xu et al. [14] proposes the software defined smart home (SDSH). The authors use the core idea of SDN to design an SDSH and list the advantages of SDN such as centralization, optimization, and virtualization in designing the smart home. The SDSH discussed in the work focuses on

978-1-5386-0728-2/17/$31.00

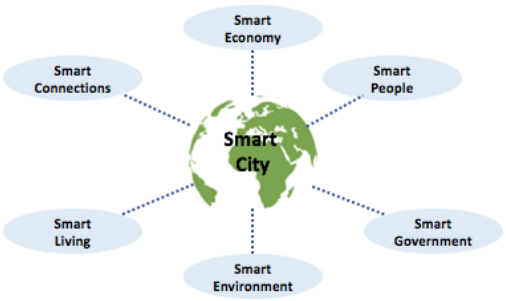

Fig. 1. Smart City [8]

building a feasible system for easy operation and open APIs to connect with third-party services. The work proposed in this paper deals with managing the network inside the home, whereas in our work, we propose to use SDN for the network in a smart community connecting the homes together for home alert management.

In [10], the authors designed a bandwidth allocation framework for an SDN based smart home. The bandwidth allocation framework proposed in this work is based on SDN architecture and is able to manage IoT devices for each smart home by designating an ISP to optimize the bandwidth allocation on both internal home traffic and external Internet traffic. In our work, we take advantage of the flexibilities of using SDN described in [10] to design a more manageable and flexible smart home alert management system for sensors in a smart home.

III. SDN:Overview

Software-Defined Networking (SDN) provides a dynamic, flexible, and controllable platform for making it an important network architecture for the dynamic nature of today's network applications. In our approach, the underlying network function for our smart home management system (SeSAMe) is provided using SDN. SDN is a network architecture where the data plane and the control plane are decoupled from each other, and the data plane is managed remotely by the control plane. The four main features of SDN are [9] as follows: (1) control plane and the data plane are separated, (2) the control plane can be centralized, (3) the control plane can be programmed, (4) the application programming interfaces (APIs) are standardized.

Fig. 2 depicts an overview of the SDN architecture design that shows three layers: the data plane, control plane, and application plane (management plane). The decoupling of the data and the control plane makes the network administration becomes flexible and manageable, as well as significantly lowers the cost of the physical data forwarding hardware.

The centralization of the control plane has brought in several significant advantages. As compared with low-level device-specific configurations, the centralization of the control plane makes the overall network architecture less error prone to modify network policies through high-level languages and software components [11]. It allows the control plane to get a global network view via dynamic, automated SDN programs [5], [9]. The SDN controller, as the control entity of SDN architecture, manages the network flows to the data plane that lies on the southbound protocols (ex. OpenFlow) and to the applications that lie on the northbound interface by using different API calls [7].

By providing great flexibility of managing network application flows, SDN can manage all the network services via the control plane to allow the dynamic response to network needs [1], which is a key factor in the prospect of Smart Cities. For example, when there are many nodes in the network sending messages simultaneously, our proposed SeSAMe home management system can control the network based on a global network view and assign network resources accordingly.

IV. SeSAMe: Architecture

The architectural vision of SeSAMe is shown in Fig. 3. It shows homes that are located in a neighborhood; for each neighborhood, a fog based/localized SDN controller may be assigned.

Each home has a home gateway that monitors the reading from the sensors in the home. The home gateway is connected to a centralized SDN controller. The overall network in a neighborhood has a tree structure that is common for connecting residences to the rest of the system. When any event such as a fire is detected, the fire sensor sends the reading to the home gateway. The home gateway would then try to notify the different homes in the neighborhood as well as the police and fire departments about the occurrence of the event by communicating through the SDN controller. It installs flows on the route so that the home where the fire has occurred is able to notify the other entities. Fig. 4 and 5 depict a fire scenario. In Fig. 4, when the sensor detects the fire in one of the homes, it notifies the home gateway, which in turn attempts to alert different homes in the neighborhood, the fire department, and the police department about the incident. Since it may not know the route to other homes and the fire and the police deparments, it sends a notification to the controller about the alert, the type of alert (e.g., fire, acts of nature, or theft), its location, and asks it for the route to reach other homes.

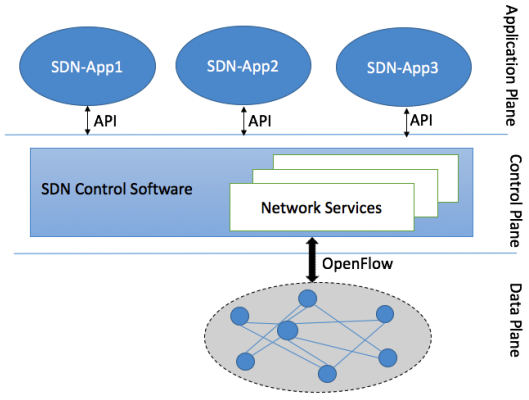

Fig. 2. The SDN Framework

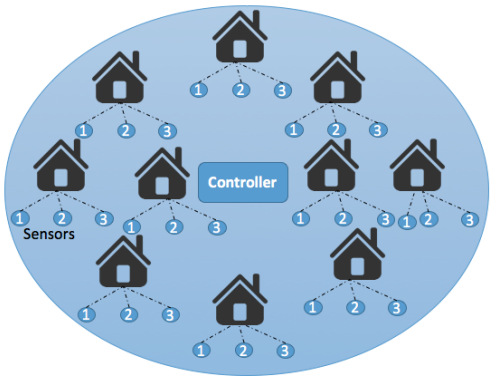

Fig. 3. SeSAMe: Connected Smart Home Architecture

The controller then alerts the police and fire departments and informs the home about the route to other homes and installs flows on them as shown in Fig. 5. Once the flows are installed, the fire affected home sends alerts to the different homes that include the alert id and its location.

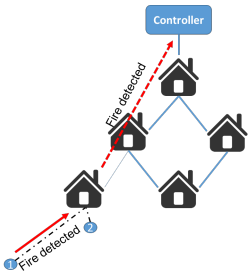

Fig. 4. Fire notification from home to controller

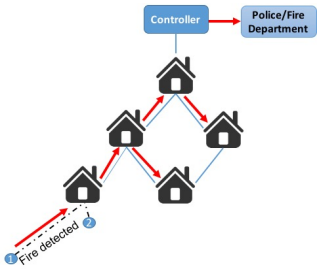

Fig. 5. Notification from the fire effected home to different homes

For system resilience, each sensor in a house is also associated with a secondary home gateway from another home in the neighborhood. This way, if a sensor's primary home gateway is non-responsive (e.g., the fire disabling the home gateway as well), then the secondary home gateway can still communicate about the distress message generated by the sensor.

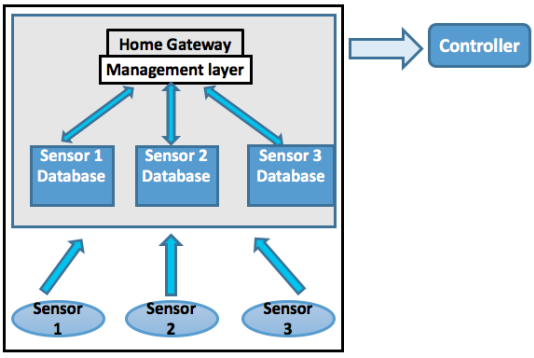

Fig. 6. Home Architecture

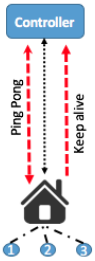

Fig. 7. Connection between home and controller

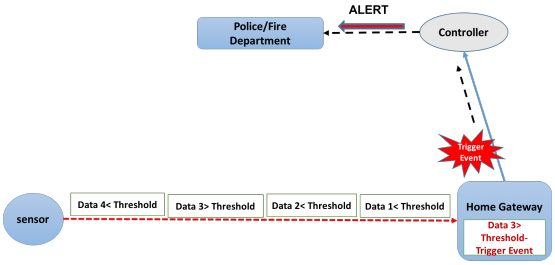

Fig. 8. Trigger Event

Fig. 6 shows the high level architecture of a smart home in SeSAMe. It can be categorized into two categories: home gateway and sensors. The sensors include different sensors that are part of the home, e.g., fire sensor, temperature sensor, light sensor, motion sensor, and so on. All sensors send their data to the home gateway. The controller creates a database of the readings from various sensors. As shown in the figure, there are three sensors in the home and the controller creates a database for each of the sensors. The home gateway consists of a database where the reading from different sensors is stored, at least temporarily.

The management layer is the core of a smart home. It continues monitoring the data coming from different sensors. Based on the data collected from the sensors, it creates a triggered event that is sent to the controller along with the type of data and the reading (or a notification that the fire has been detected).

A. Message Types

Fig. 7 depicts the connection between a smart home and the controller. The smart home continue to send keep alive messages to the controller at regular intervals to ensure that the link between the home and the controller is working and the home is connected to the controller. At the same time, the controller periodically sends a PingPong message to the homes to ensure they are connected to the alert system. Keep alive messages sent by the home gateway are a type of a one way communication, while PingPong messages are a type of a two way communication.

In SeSAMe, we define four different types of messages as shown in Table I.

- **Update** messages are sent from the home to the controllers to make it aware of its reachability and any

TABLE I
MESSAGE TYPES

Message Type	Sender-receiver	Description
Update	Home-to-Controller	Sends updates to the controller
Trigger	Home-to-Controller	Event triggered message to notify the controller of any alert
Announce	Home-to-Homes	Notify the homes of any alert
Keepalive	Home-to-Controller	Notify the controller that the home is connected

changes made to any device at home. There are two types of timers used here. One is the update timer and the other one is the holddown timer. The holddown timer value is several times the value of the periodic update timer. If the updates sent by the homes to the controller are lost, the controller would not assume that the home is down. Instead, it waits until the holddown timer expires. When no updates are sent by the home to the controller (when the holddown timer exipres) it assumes that the devices at this home are not accessible.

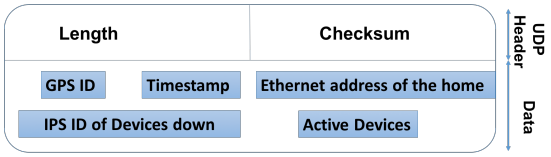

Fig. 9. Update message fields

The different fields of update messages are shown in Fig. 9. We assume that all sensors and the home gateways are equipped with an Indoor Positioning System (IPS) [2] for an accurate GPS ID of the locations. The GPS ID of the home gateway represents the GPS location of the home. Updates are sent every fixed regular interval. Active devices is an optional field that represents all the devices that are working properly.

- A **Trigger** message is generated by the homes to notify the controller of any alerts. The information in the trigger message includes the alert type. Trigger events here denote the conditions when the home gateway sends alerts to the controller. Fig. 8 shows the triggered events. In the management layer, the threshold levels for different readings are already defined. Thus, when the management layer finds any of the readings going above the threshold, it creates a triggered event. This triggered event is used to notify the controller about the event (for example, fire). Fig. 10 shows the fields included in the trigger message. Alert ID represents the ID of the sensor reading that is sent to the controller. Different sensor alerts will have a unique alert ID code, which would be attached with the trigger event packet and sent to the controller. For example, the fire alert may have an alert code "101 Fire". By checking the alert code, the controller knows what kind of alert it is. The Value field represents the reading of the alert device.

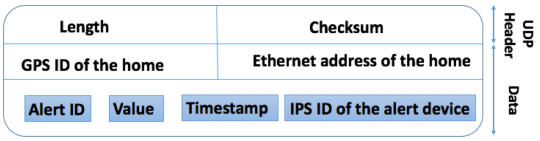

Fig. 10. Trigger message fields

- An **Announce** message is generated by the home where the event occurred and will be sent to all other homes in the neighborhood. When an event takes place in a home, it notifies the controller. The controller in turn installs flows on the route connecting the affected home to all other homes in the neighborhood. Once the flows are installed, the affected home announces the message to the other homes. The announce message fields are shown in Fig. 11. An alert ID again represents the type of alert. The GPS ID of the alert home represents the location of the home where the event occurs.

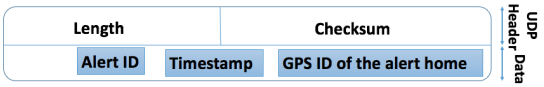

Fig. 11. Announce message fields

- **Keepalive** messages are sent by the homes to the centralized controller to check that the connectivity between them is working or to prevent the link from being broken [3]. It is a one way communication. If the controller does not receive the keepalive message from the home after a predefined interval, it would mean that either the home or the link between the controller and the home is down. It is useful in scenarios where a link is down and the traffic has to be rerouted via another path. The message fields are shown in Fig. 12.

Fig. 12. Keep-Alive message fields

The home gateway communicates with the central controller using two protocol PingPong protocols. The PingPong protocol is initiated by the controller by sending the first message. After receiving the message, the client, which is the home gateway in this case, responds to the server. The PingPong protocol makes sure the home gateway is working and is connected to the controller. If the controller does not receive any response from the home gateway, it first sends a pull request. If the home does not respond to the request, it assumes that the home gateway is down.

The controller keeps listening for the triggered events sent by the home gateways. Also, after repeat intervals, it keeps using the PingPong protocol. If any home sends a trigger event to the controller, the controller then forwards it the addresses of the homes where the alert should be broadcasted and also installs flows on the route.

B. Sensor Data

Different sensors in the home will send data to the home gateway and the secondary home gateway assigned to each sensor. Each of the sensor datum will consist of 6 different parameters: timestamp, location, home address, phenomena, value, and unit. The timestamp denotes the time when the reading of the sensor was taken; the location denotes the IPS location; the home address would be the address of the primary gateway to which the sensor is connected; the phenomena represents the kind of reading that is being measured; the value is the reading of the sensor and the unit is the unit associated with the sensor.

C. Policy Rules

The central controller also maintains various policy rules. For example, certain types of information available from the sensors at a home may only be sent to the police and the fire department, but not to the others in the neighborhood. Similarly, if a home does not wish to be a secondary home gateway for sensors in the neighborhood, this can be specified through a policy rule.

V. System Initialization and Operation

The SeSAMe setup includes setting up the home gateway, and then the sensors are installed. Once the sensors are installed, it sends out a 'discover' message to the nearby gateways. The gateways, after receiving the 'discover' message, send an 'offer' message in reply to it, which includes the location of the gateways. As there can be more than one gateway in the neighborhood sending the 'offer' message to the sensor, it uses the election protocol to select the primary and the secondary gateway. It selects the gateway that is closest as the primary gateway (which should be the home where the sensor is installed) and the secondary gateway is the second nearest gateway. After the primary and the secondary gateways are selected, the sensors send a 'request' message to the selected gateways, indicating the gateways to add them as an authorized device. The home gateways send an 'ack' message reply to the sensor indicating that it has been added as an authorized device and the gateway is ready to receive the data from the sensors.

As a result of the primary and the secondary home gateways, at any particular time the data transmitted by the sensor are received by these two gateways. Therefore, the controller receives information about the sensor status of homes from the primary as well the secondary controller at any instance of time. A few scenarios shown below detail the use of the primary and secondary controllers:

- Normal condition: When the primary home gateway is up, the centralized SDN controller receives information from the primary as well as the secondary home gateway. In this case, the information received from the secondary controller is ignored.
- Home gateway failure/ Link failure: When the primary home gateway loses connection to the centralized SDN controller, which can either happen because of the home gateway failure or due to the failure of the link between the home and the centralized controller, the secondary controller takes the role of the primary controller. So the centralized controller would receive information from the secondary controller as a backup way to update the sensors.

A. Issues

SeSAMe could be an opt-in service. In this case, the home owners, by opting-in, agree to share their home locations and sensor information. It could also include registering the user's phone number/numbers that can be used to send alert messages. While as a service, there are benefits of SeSaMe, it also could present privacy risks for the owner. If the location of the user's house is shared with the controller and is maintained by a third party provider, it raises privacy risks. Services for which the location/information of the user are shared, require the use of a secure communication and privacy preservation. If the server becomes a point of attack, the user's information could be compromised.

VI. Simulation Setup and Preliminary Results

The SeSAMe testbed was setup on GENI [4]. The experimental topology is shown in Fig. 16. There are 12 homes as shown in the figure. We tested the time to notify different hosts in the topology. The capacity of each link was set to 100 Mbps. We tested the normal scenario as well the home gateway failure scenario where home 1 loses connection to the controller and home 2 acts as the secondary controller. The result for both the scenarios are the same as both homes are in the same cluster.

As the message size increased and the number of sensors sending the message increased, the time to receive the message increases. We measured the time it took to send a message from home 1 to home 3 that was on the same node cluster and to home 4 and home 7 that were on different node clusters. Table II shows our simulation results based on each case being independently replicated 10 times and $\pm$ represents the 90% confidence interval. One-to-One means the message was sent to just one host. For example, h1-h3 means when home 1 is sending the alert just to home 3. The number of sensors represents the total number of sensors that are sending messages at a time (with each sensor sending one message). One-to-Three signifies the time it takes for the alert to reach home 3, home 4, and home 7 when home 1 sends the alerts to three hosts at the same time. The time for One-to-Three is greater (as compared to One-to-One) because the alert is sent to the three hosts at the same time, so the link gets congested

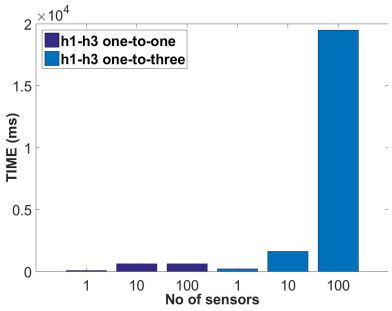

Fig. 13. h1-h3

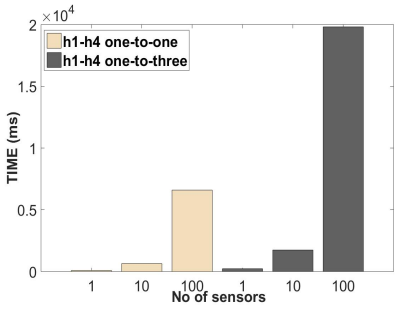

Fig. 14. h1-h4

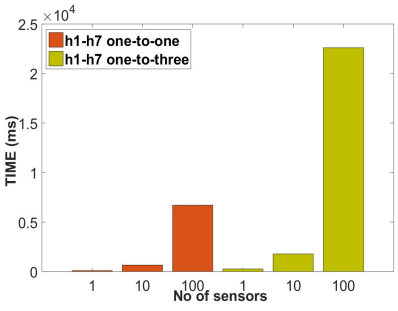

Fig. 15. h1-h7

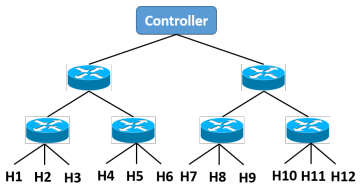

Fig. 16. Experimental Topology

as the capacity of the link is limited. The graphs shown in Fig. 13, 14, and 15 show the time to receive the message when the message of 1 MB is sent. For each of the graphs shown, we compared the time to send the message for one-to-one Vs. one-to-three for 1,10, and 100 sensors. We can see that the time to send the message was not too significant (keeping in mind that the capacity of the link was 100Mbps) as the message size increased and the distance to which the message being sent also increased.

TABLE II
RESULTS

		One-to-One (ms)		
	No of Sensors			
		100 bytes	**10000 bytes**	**1 MB**
h1-h3	**1**	8.467 ±0.65	8.52 ±0.25	84.56 ±0.57
	10	9.83 ±1.10	12.65 ±1.87	626.08 ±57.61
	100	112.86 ±50.71	93.79 ±43.50	6376.47 ±108.16
h1-h4	**1**	9.70 ±0.31	9.59 ±0.30	88.90 ±1.24
	10	11.16 ±1.12	13.88 ±5.30	647.58 ±60.66
	100	150.10 ±61.41	110.41 ±59.17	6598.44 ±123.40
h1-h7	**1**	10.80 ±0.17	10.72 ±0.22	96.92 ±4.62
	10	13.12 ±1.75	16.80 ±0.68	658.74 ±54.68
	100	111.18 ±51.46	119.88 ±63.79	6703.80 ±124.43
		One-to-Three (ms)		
		100 bytes	**10000 bytes**	**1 MB**
h1-h3	**1**	17.19 ±2.36	19.96 ±2.42	218.75 ±16.56
	10	117.70 ±10.81	124.19 ±6.00	1632.31 ±84.42
	100	437.66 ±46.40	513.55 ±79.80	19494.4 ±640.81
h1-h4	**1**	24.70 ±2.59	25.49 ±2.97	229.46 ±14.53
	10	131.11 ±3.60	133.02 ±8.83	1745.22 ±178.19
	100	458.76 ±42.14	531.38 ±44.75	19815.8 ±627.82
h1-h7	**1**	27.99 ±3.86	29.54 ±4.72	268.98 ±37.33
	10	139.437 ±2.94	133.66 ±7.26	1784.10 ±188.14
	100	476.729 ±33.73	543.911 ±41.22	22604.9 ±273.35

VII. Conclusion and Future Work

In this paper, we propose SeSAMe, a software defined smart home alert system, based on SDN, which is an automated system to send notification alerts to other homes, and the police and fire departments. The preliminary results show that using the SDN approach alerts can be communicated very quickly. Moreover, they give the flexibility of programmable control functions, lower operating costs, and centralized management, to name a few.

In the future, we plan to implement a system of distributed controllers in place of a single controller when a large neighborhood area is involved. Here, all controllers for different neighborhoods would be connected to each other. Thus, each controller would choose another controller that would act as its backup controller. A backup controller will be used only when the controller reaches its processing threshold, i.e., it is no longer able to process any requests. During such scenarios, it will forward all the requests to its designated backup controller.

Acknowledgement

This work is partially supported by the National Science Foundation Grant # 1526299.

References

[1] http://www.quotecolo.com/smart-cities-how-sdn-and-nfv-are-changing-the-way-we-live/.
[2] https://en.wikipedia.org/wiki/Indoor_positioning_system.
[3] http:https://en.wikipedia.org/wiki/Keepalive.
[4] https://www.geni.net.
[5] Software-defined networking (SDN) definition.
[6] https://en.wikipedia.org/wiki/Smart_city.
[7] https://www.sdxcentral.com/resources/sdn/sdn-controllers/.
[8] http://www.smartbrantford.ca/TheSixComponents.aspx.
[9] R. Jain and S. Paul. Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine*, 51(11):24–31, 2013.
[10] H.-C. Jang, C.-W. Huang, and F.-K. Yeh. Design a bandwidth allocation framework for sdn based smart home. In *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*, pages 1–6. IEEE, 2016.
[11] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.
[12] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin. Smart community: an internet of things application. *IEEE Communications Magazine*, 49(11), 2011.
[13] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche, and C. Loge. The smart home concept: our immediate future. In *2006 1st IEEE international conference on e-learning in industrial electronics*, pages 23–28. IEEE, 2006.
[14] K. Xu, X. Wang, W. Wei, H. Song, and B. Mao. Toward software defined smart home. *IEEE Communications Magazine*, 54(5):116–122, 2016.