An Implementation of Harmonizing Internet of Things (IoT) in Cloud

Md. Motaharul Islam[1,2(✉)], Zaheer Khan[3], and Yazed Alsaawy[2]

[1] Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh
motaharul.islam@bracu.ac.bd

[2] Islamic University of Madinah AlMadinah, Medina, Kingdom of Saudi Arabia
yalsaawy@iu.edu.sa

[3] Khana-e-Noor University, Pole Mahmood Khan Shash Darak kabul, Kabul, Afghanistan
zaheeriut13@gmail.com

Abstract. With the evolution of Internet of Things (IoT), everything is going to be connected to the Internet and the data produced by IoT, will be used for different purposes. Since IoT generates huge amount of data, we need some scalable storage to store and compute the data sensed from the sensors. To overcome this issue, we need the integration of cloud and IoT so that the data might be stored and computed in a scalable environment. Harmonization of IoT in Cloud might be a novel solution in this regard. IoT devices will interact with each other using Constrained Application Protocol (CoAP). All the IoT devices will be assigned IP addresses for unique identification. In this paper, we have implemented harmonizing IoT in Cloud. We have used CoAP to get things connected to each other through the Internet. For the implementation we have used two sensors, fire detector and the sensor attached with the door which is responsible for opening the door. Thus the proposed implementation will be storing and retrieving the sensed data from the cloud. We have also compared our implementation with different parameters. The comparison shows that our implementation significantly improves the performance compared to the existing system.

Keywords: Harmonization · Cloud · Internet of Things
Constrained application protocol · Smart environment · Arduino shield

1 Introduction

Nowadays everything is getting smarter. With the evolution of the Internet of Things (IoT), the demand for users to use smart things are increased. In order to make things and environment smarter and connected the researchers are thinking to assign IP addresses [1] to all the smart devices [2]. These devices will have the capabilities of sensing and actuating the environment. Since huge amount of

A.-S. K. Pathan et al. (Eds.): SGIoT 2018, LNICST 256, pp. 3–12, 2019.
https://doi.org/10.1007/978-3-030-05928-6_1

data is sensed by the sensor from the smart environment we have to store and process it for further data analytics. To store the data cloud storage is a very good and reliable alternative.

Data from sensed environment would be transferred to the storage through Hypertext Transfer Protocol (HTTP) which is designed for the Internet devices. It consumes more time in case of GET and POST method. In this connection, a novel framework has been suggested to cooperate cloud-level IoT services on a smart device with the concept of Intention [3]. Captured data from sensors were stored in local machines. A repository might be used for storing the status of the sensors. The sensed data might be stored in the cloud using HTTP if required. In this paper concept of local repository is eliminated. A cloud based architecture is proposed where data will be stored in the cloud. Using cloud storage instead of physical storage will make it more reliable. If the data is stored in the cloud the aspect of physical damages and security will be enhanced as well. Internet and some gateways [4] might be used to connect the sensors to the cloud for storing and retrieving the required information. In order to provide the connection between the sensors some well-known protocols might be used.

For sensor-based connections two protocols are preferable Message Queue Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP). In this paper we have introduced CoAP for the connectivity purpose instead of using HTTP. Using CoAP protocol in the connectivity area will make the system faster. On the other hand, CoAP protocol is designed for resource constraint smart devices where HTTP was designed for the traditional internet devices. In this paper CoAP is used to harmonize sensors with cloud in such a way that data taken from one of the sensor is a command towards next sensor in order to make an alert for smart doors in smart environment. Harmonization is the process of joining more than one devices to perform a single task. Harmonizing means that input is applied from different sources and output is combined by a group of sensors. Here, we harmonize two sensors to perform the task of alerting the doors if fire exists.

In the harmonization process cloud acts as a remote storage and IoT devices (e.g. sensors) act as actuators in the smart environment. The sensed data is collected from the environment and transferred to the cloud through CoAP in order to update the status of the sensors in the cloud. In this paper smart doors' sensors checks its' status in the cloud. If the status's value is above the defined threshold. Smart doors will be opened automatically for the fire exit. Two types of sensors will be used to control this scenario, smoke sensors and the sensors attached with the smart doors. The smoke sensor will be responsible to detect the existence of fire. If fire exists, it will send the sensed value to the cloud storage. In the storage the sensor will have a status and the value of its status will be initialized to the standard value. This value indicates that the temperature up to this threshold will be normal. If the sensed value passed to the storage is greater than defined threshold value then the status will be updated to that sensed value. The sensor attached with the door continuously checks the status of the smoke

detector sensor. If it figures out that the status value is greater than defined threshold value, the door sensor will alert the door to be opened urgently.

Since the smart doors are opened automatically it may affect the security of the building. To ensure the security sensors may be placed inside the building and more specifically inside the room so the smart doors are controlled by the sensors inside the room. No sensor outside the building will be able to control the door.

The main contributions of this paper are as follows:

- We have proposed cloud based architecture for harmonizing IoT.
- We have implemented the architecture using CoAP.
- We have analyzed and compared our implementation with HTTP using different cloud platforms.
- We have compared CoAP with MQTT and have shown the suitability of CoAP over MQTT considering some text comparisons.
- We have compared CoAP with HTTP and the comparison shows that by using CoAP, performance has been increased by 33.76% and 11.45%, 10.83% and 19.98%, 10.52% and 16.92% using GET and POST methods respectively for Google cloud, Thingspeak and 000Webhost.

The rest of the paper is organized as follows. Section 2 reviews related works. System architecture is presented in Sect. 3. Section 4 describes algorithmic analysis. Section 5 represents the layered architecture of the proposed system. Section 6 shows the way of implementation. Section 7 shows the experimental evaluation. Finally Sect. 8 concludes the paper.

2 Related Works

Seung Woo Kum et. al. proposed IoT device delegate system [3]. This IoT device delegate has a database to store the information of devices. Device delegate creates data channel from the devices so that data can be streamed. An Intention Manager is also proposed. It makes decision for which action will be performed based on request from the devices. For acquiring data via data channel, they have used HTTP.

IoT devices have limited communication and processing capabilities due to the energy consuming communication, small physical factors, battery operations, and long lifetime expectations. The concept is that, devices transfer data using sensors for communication between them. There are various types of sensors for various purposes. It is difficult to use same protocols for all of them. A multi-protocol receiver is proposed [10]. This receiver retrieves data simultaneously from different communication technologies like WiFi, ZigBee, and Bluetooth. The data is stored and processed in local database (i.e MySQL). For back end analysis, they have used cloud and for transferring data to cloud HTTP has been used. However, the receiver has been used in terms of multi-protocol context. The data was also stored in local machines.

Cloud Computing and IoT are currently two of the most popular ICT paradigms that are expected to shape the next era of computing. The convergence between cloud computing and IoT has become a hot topic over the last few years because of the benefits that IoT could have from the distributed nature of cloud computing infrastructures. This paper [13] proposes a new platform for using cloud computing capacities for provision and support of ubiquitous connectivity and real-time applications and services for smart cities' needs. We present a framework for data procured from highly distributed, heterogeneous, decentralized, real and virtual devices (sensors, actuators, smart devices) that can be automatically managed, analyzed and controlled by distributed cloud-based services.

The authors had proposed IoT and cloud based healthcare system [11]. This was a wearable system. In this system, they had used textile accelerometers, a temperature sensor, a heartbeat chest and Arduino open hardware microcontroller platform called LilyPad. Arduino microcontroller collects data from sensors and data is passed to android-based mobile phone through Bluetooth interface. An application is also developed for android that collects data and sends it to cloud.

HTTP is specifically designed for internet devices not for the constraint devices. In this paper we migrate the local storage to the cloud storage. The intention is alarming one device based on other device. There is no need of intention manager. In order to establish the connection between the sensors and the cloud storage CoAP is used. As devices used (e.g. sensors) are constrained, CoAP is more suited protocol over HTTP. Here multi-protocol receiver system is not required as we will use single device with a single protocol. Researchers used local database for storing data. Here storage is cloud.

In this paper sensors will be connected through WiFi. The data will be stored and processed in cloud. As the sensor will act on processed data that are being sent from cloud intention manager is not required. Additional hardware requirements are eliminated (e.g. local storage devices). This reduces the cost and helps to prevent hardware revisions. As the data is automatically passing from one sensor to another, there is no need of user interaction. Here CoAP is used as an application layer protocol. CoAP is more suited to very small sensor deployments with tiny hardware and completely different security.

3 System Architecture

The proposed architecture is depicted in Fig. 1. The step by step functionality of the proposed architecture is explained as follows:

i We have used two sensors as shown in the architecture given in the Fig. 1 for fire detection and for controlling the door (i.e. Alerting the door to be opened if fire exists).
ii In order to connect the sensors to the Internet for exchanging the data Arduino is used as an intermediate device. Arduino will be programmed to

perform the required task. Arduino and the sensors are connected through a WiFi shield.

iii CoAP protocol is used for exchanging data between sensors through cloud.

iv The fire/smoke detector will store the sensed data into the cloud storage, where the status of the sensor is stored to point out the existence of fire.

v The sensor attached with the door is responsible to alert the door to be opened in case the existence of fire. This sensor will be getting the alert information regarding fire existence from the cloud storage. The door sensor will be continuously checking the status of the fire sensor, if it detects the status to be on (i.e Indication of existence of fire) immediately smart doors will be opened for the exit.

vi For Cloud Service there can be several platforms (e.g. Google Cloud, Thingspeak, 000Webhost, Amazon, Microsoft Azure etc.). In our work, we will use Google cloud for getting the cloud facilities.

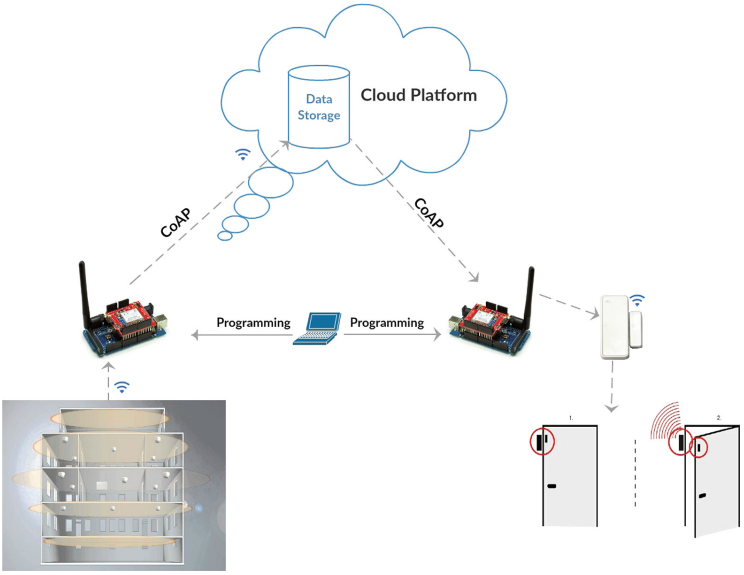

Fig. 1. Proposed architecture of harmonizing IoT in Cloud.

4 Algorithmic Analysis

We have proposed two algorithms to show the overall functionality of the system.

Algorithm 1 for smoke detector shows the overall responsibility and the functionality of the smoke sensor. The variable sensor status given in Algorithm 1, shows the status of the smoke sensor. This variable is shared between the smoke sensor and the sensor attached with the door. As given in the algorithm, the

sensor is sensing the environment continuously and if the sensor's sensed value is greater than defined threshold value (i.e. Indication of fire existence) then it will update its status (i.e. Shared variable) in the storage.

The Algorithm 2 explains that how the sensor attached with door will get notification regarding existence of fire. The sensor attached with the door is continuously checking the status (i.e. shared variable) of the smoke sensor and if it finds out that the value of the shared variable is greater than defined threshold value then immediately it will alert the door to be opened for the fire exit.

Algorithm 1 Smoke Detector

```
1:  procedure SMOKESENSOR
2:      unLock the sensorStatus variable in the cloud
3:      initialValue ← the standard temperature value assigned to sensor ▷ if it exceed this value it means there is an existence of fire so the door sensors should be triggered.
4:      while TRUE do
5:          sensedData ← sensing the environment
6:          if sensedData is greater than initialValue then
7:              sensorStatus ← sensedData ▷ update the sensor status in the cloud
8:              Lock the sensorStatus variable in the cloud so that no sensor could update it within this period of fire exit
9:              break
10:         end if
11:     end while
12: end procedure
```

Algorithm 2 Sensor Attached With The Door

```
1:  procedure DOORSENSOR
2:      initialValue ← the standard temperature value ▷ if it exceed this value it means there is an existence of fire so the door sensors should be triggered and open the door immediately.
3:      while TRUE do
4:          sensedStatus ← get the sensor status from the cloud storage continuously
5:          if sensorStatus is greater than initialValue then
6:              Alert the door to be opened
7:              break
8:          end if
9:      end while
10: end procedure
```

5 Layered Architecture of the Proposed System

The architecture of IoT as shown in Fig. 2 is considered to be multilayered. There are basically three layers with Perception layer, Network layer and Application layer. There are two more layers: Middleware layer and Business layer [7,8].

Perception Layer. Perception layer is the lowest layer of the architecture. The main task of this layer is to sense the data. At first the object must be identified and then collect the data from the objects for example sensors. In our model, fire sensor will continuously sense the environment to figure out the existence of fire.

Network Layer. From Perception layer data is passed to network layer. Network layer gathers data from lower layer and sends it to the Internet. In network layer there is a gateway, with two interfaces, one connected to the sensor and other to the Internet [6].

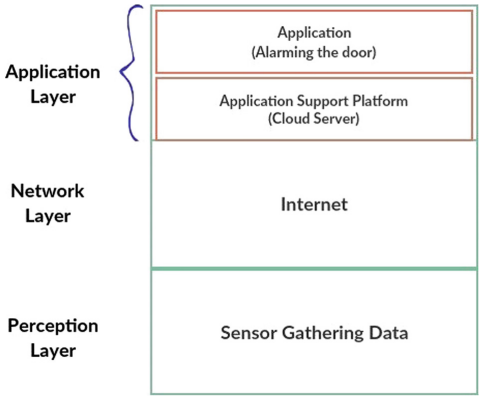

Fig. 2. Layered architecture of harmonizing IoT.

Application Layer. Application layer is for the user level interaction, it provides the service for the collected data from the sensors. This layer consists of application support platform and concrete applications [7]. In our proposed model the application layer has been divided into two parts, the application itself and application layer support platform.

Application (Alarming the door). This part will provide the service of opening the door in case of any emergency exit due to existence of fire in the building.

Application Support Platform. We have used Cloud Server as an application support platform for the end sensors.

6 Way of Implementation

Data from IoT pass to cloud for being stored and later on the stored data is used in order to perform the desired task (i.e. alerting the door to be opened in case of any fire existence). Source Code for the implementation is written in arduino programming and is installed into a WiFi arduino shield to interact with the devices. Devices for example sensors produce a lot of data at every moment. It is irrelevant to store all the data as it is not necessary. In order to overcome this problem a smart gateway would help for better utilization of data and cloud resources [4]. There are two types of architecture for web services, REpresentational State Transfer (REST) and Service-Oriented Architecture (SOA). SOAs are often used to model and realize complex business flows. In this paper REST architecture is used. The RESTful protocols are CoAP and HTTP. HTTP uses HTTP-similar standardized methods. For interacting with things for example doors and sensors, CoAP is used which is designed for constrained devices. Many commercial and open source platforms are available for IoT and cloud implementation [5].

7 Experimental Evaluations

We have done some experiments on both the methods. Comparing CoAP with HTTP using GET and POST methods, CoAP requires less amount of time. The experiment is done on Google cloud storage [15], Thingspeak [16] and 000webhost [17].

Figure 3 x-axis and y-axis represents per request and time in milliseconds respectively. As shown in the graph, HTTP requires more time than CoAP both in GET and POST methods. Here Google Cloud storage is used to store the data. In case of GET and POST methods, HTTP requires 554 ms, 1520 ms and CoAP requires 367 ms and 1346 ms respectively.

Figure 4 shows the comparison between HTTP and CoAP when Thingspeak storage is used. As shown in the graph for GET method, HTTP requires 1902 ms where CoAP requires 1696 ms and in case of POST method, HTTP requires 1266 ms where CoAP requires 1013 ms.

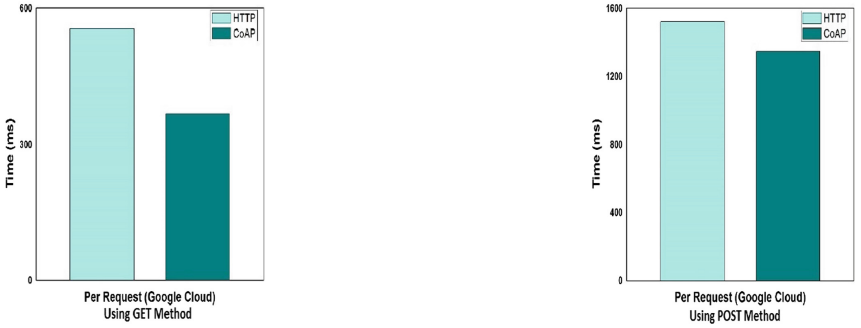

Fig. 3. Comparison of HTTP and CoAP protocols based on GET and POST method using Google Cloud as a Storage.

Fig. 4. Comparison of HTTP and CoAP protocols based on GET and POST method using Thingspeak as a Storage.

Fig. 5. Comparison of HTTP and CoAP protocols based on GET and POST method using 000webhost as a Storage.

Figure 5 shows that for GET and POST methods, HTTP requires 2415 ms, 1389 ms and CoAP requires 2161 ms, 1154 ms respectively using 000webhost as a storage.

Finally we can say that in each stage of comparison HTTP is more time consuming while interacting through GET and POST methods. We have introduced the implementation of our work using CoAP where it requires less time and is faster than HTTP.

8 Conclusions and Future Works

In this paper we present an idea to harmonize two sensors such that they will interact with each other and act on output produced by one of the sensor. One is to detect the existence of fire and the other one acts on the output of the fire detector sensor. Which is used to control the door in case of existence of fire. Here we have stored the sensed data in the cloud. We do not have local storage. Hence the system becomes cloud based. In our implementation the performance has been increased compared to the existing system. In our future work, we are going to implement and commission the proposed idea in a smart environment for commercial uses.

References

1. Islam, M.M., Huh, E.-N.: Sensor proxy mobile IPv6 (SPMIPv6) – a novel scheme for mobility supported IP-WSNs. Sensors **2**(11), 1865–1887 (2011)
2. Islam, M.M., Huh, E.-N.: A novel addressing scheme for PMIPv6 based global IP-WSNs. Sensors **11**(9), 8430–8455 (2011)
3. Kum, S.W., Moon, J., Lim, T., Park, J.: A novel design of IoT cloud delegate framework to harmonize cloud-scale IoT services. IEEE International Conference on Consumer Electronics (ICCE), pp. 247–248 (2015)

4. Aazam, M., Hung, P.P.: Cloud of things integrating internet of things and cloud computing and the issues involved. In: The proceedings of 11th IEEE International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 414–419 (2014)
5. Botta, A., de Donato, W., Persico, V., Pescapé, A.: On the integration of cloud computing and internet of things. In: The 2nd International Conference on Future Internet of Things and Cloud (FiCloud), pp. 23–30 (2014)
6. Chong, G., Zhihao, L., Yifeng, Y.: The research and implement of smart home system based on internet of things. In: International Conference on Electronics Communications and Control (ICECC), pp. 2944–2947 (2011)
7. Khan, R., Khan, S.U., Zaheer, R., Khan, S.: Future internet the internet of things architecture, possible applications and key challenges. In: 10th International Conference on Frontiers of Information Technology (FIT), pp. 257–260 (2012)
8. Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., Du, H.-Y.: Research on the architecture of Internet of things. In: 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 5, pp. V5–484 (2010)
9. Gunasagarana, R., et al.: Internet of things sensor to sensor communication. In: IEEE Sensors, pp. 1–4 (2015)
10. Su, J.-H., Lee, C.-S., Wu, W.-C.: The design and implementation of a low-cost and programmable home automation module. IEEE Trans. Consum. Electron. **52**(4), 1239–1244 (2006)
11. Levä, T., Mazhelis, O., Suomi, H.: Comparing the cost-efficiency of CoAP and HTTP in Web of Things applications. Decis. Support Syst. **63**, 23–38 (2014)
12. Daniel, L., Kojo, M., Latvala, M.: Experimental evaluation of the CoAP, HTTP and SPDY transport services for internet of things. In: Fortino, G., Di Fatta, G., Li, W., Ochoa, S., Cuzzocrea, A., Pathan, M. (eds.) IDCS 2014. LNCS, vol. 8729, pp. 111–123. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11692-1_10
13. Suciu, G., Vulpe, A., Halunga, S., Fratu, O., Todoran, G., Suciu, V.: Smart cities built on resilient cloud computing and secure internet of things. In: 19th International Conference Control Systems and Computer Science (CSCS), pp. 513–518 (2013)
14. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.: Internet of things for smart cities. IEEE Internet Things J. **1**(1), 22–32 (2014)
15. Google Cloud Homepage. https://www.cloud.google.com. Accessed 20 Feb 2018
16. Thingspeak Homepage. https://www.thingspeak.com. Accessed 20 Feb 2018
17. 000Webhost Homepage. https://www.000webhost.com. Accessed 20 Feb 2018
18. Thangavel, D., Ma, X., Valera, A., Tan, H.X., Tan, C.K.: Performance evaluation of MQTT and CoAP via a common middleware. In: IEEE Ninth International Conference on Intelligent Sensors Sensor Networks and Information Processing (ISSNIP), pp. 1–6 (2014)