Implementation of server side service layer in Internet of Things

Jaison Hezakiel[1], Dr. Vijayakumar R[2]

[1]M Tech student, School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala
jaison7391@gmail.com
[2]Professor, School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala
vijayakumar@mgu.ac.in

Abstract— *Internet of Things (IoT) is one of the most emerging technologies in this decade. Billions of devices are now connected to Internet which can also communicate with each other. There are several applications using this technology in a variety of fields such as medical, agricultural, industrial, defense etc. IoT has some major roles also in our day to day life. Using the advanced possibilities of this technology many real world processes can be automated. A large number of researches have been conducted on in this field. This paper deals with the server side service layer in IoT. It offers the cloud functionalities to the Internet of Things protocol stack.*

Keywords— Service Layer, Internet of Things, low power devices, low memory devices.

I. OVERVIEW

Internet of Things is the technology which consists of many smart devices which are connected together and each of them is called as the 'thing' [1]. It could be a television, a refrigerator, a door, a shirt etc. To make a device smart following modules are needed: a sensor, a radio module, an actuator, a gateway.

To work with these low power-low memory devices specially designed operating systems are used such as Contiki, Brillo, TinyOs etc. The Contiki operating system is based on Linux. Coniki's simulator is known as COOJA. It is an Event-Driven system. This means that the applications executing on the OS are triggered by events.

The service layer is a conceptual layer which provides the middleware services and the applications at the higher application layers. In addition the service layers provide an interface to core networks at a lower resource layer.

The services in this Internet of things environment are known as REST-ful (Representational State Transfer) services [2]. REST is an architectural style which is used for building distributed hypermedia systems. When REST is used to build a system, the services provided by the system are called as REST-ful services.

This paper is organized as follows: Section 2 gives details about the motivation. In Section 3 the challenges in implementation the service layer are discussed. Section 4 deals with the implementation details of the proposed system. Section 5 gives the details about the future works and the Section 6 is the summary.

II. MOTIVATION

Due to the advances of technologies, the sensor deployments have increased over last five years. Many Internet of Things networks have deployed at various locations in the world. As per the researches of European Commission, there will be up to 100 billion devices connected to the Internet by 2020 [1]. As a result of these developments, there will be many sensors available that can be used. Nowadays many everyday objects are integrated with internet of things though the usage and the data are restricted to that object itself only [1].

The motivation of doing this project is that the high cost and effort of deploying and accessing the sensors and the sensor services. As a solution, a combination of IoT and cloud computing can be used. The basic idea behind cloud computing is to concentrate resources such as hardware and software into few physical locations and offer those resources as services to a large number of consumers. These consumers may locate in many different geographical locations around the globe. This allows the consumers to consume a service from a service provider by paying only for the amount of resources they use. This is known as Sensing as a service [1].

This is an efficient way compared to the traditional methods. In the traditional methods the consumers have to buy resources in predefined discreet quantities with higher expenses [1]. For example, in retail online business there is a peak and off-peak seasons. During the peak seasons, the business has to buy significant amount of resources to satisfy the customer needs in the traditional methods. But during the off-peak season, these resources become idle which makes the business process inefficient. Since online retail applications are hosted in servers the business is only required to pay for the resource it consumes in the concept of Sensing as a service [1].

III. CHALLENGES

Internet of things is composed of a number of small devices having memory only about hundreds of KBs. The power consumption of these devices is also very restricted. Because of these constraints conventional Internet protocols like IPV4 or IPV6 cannot be used directly in IoT.

The radio transceiver is typically the most power consuming component in low-power wireless systems. So power efficiency translates into efficient radio duty cycling. The duty cycle is the time during the radio is active. As a solution IETF (Internet Engineering Task Force) developed a set of specialized protocols: IEEE 802.15.4 PHY, IEEE 802.15.4 MAC, 6LoWPAN (IPV6 over Low power Wireless Personal Area Networks), RPL (Routing Protocol for Low Power and Lossy Networks) and CoAP (Constrained Application Protocol). The protocol stack for the IoT applications must has some core abilities which are discussed thoroughly in [2]. They are:

1. A low power communication stack
2. A highly reliable communication stack and
3. An Internet enabled communication stack.

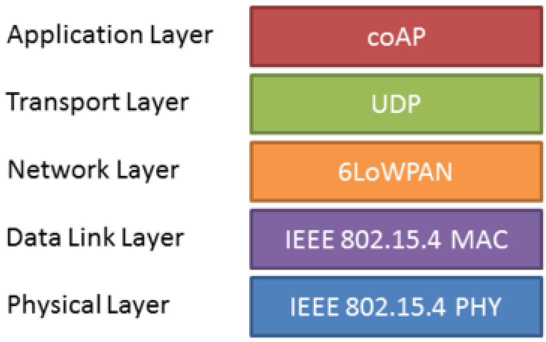

FIG 1: IETF PROTOCOL STANDARDS OF INTERNET OF THINGS [2]

Another challenge in this field is the high level of heterogeneity among the devices in the network [6]. There may be thousands of devices of different characteristics; each of them might have distinct protocols. So we need to handle these heterogeneous behaviors very wisely. A standardized protocol set is necessary. The IETF protocol standards for Internet of Things have been used commonly. There are also other standardized protocols that are not popular as that of IETF, that are defined by European Standards Organizations (ETSI, CEN, CENELEC, etc.), Standardization Institutions (ISO, ITU) and global Interest Groups and Alliances (EPCglobal, etc.)[3].

IV. IMPLEMENTATION

The server side of the service layer is implemented in Contiki OS. Contiki is written in C language. It is a light weight, event driven system that is also capable of dynamic loading of the code [4]. Here Z1 motes are used to simulate the required environment. These motes are deployed to sense the atmospheric temperature and the acceleration. The sensor data are converted into JSON (Java Script Object Notation) objects. The server is able to push the JSON file into cloud so that the client can access the sensor data. There is a security system consisting of user names and corresponding passwords to control the data access. This system authenticates the valid users to access the data.

In order to work with the server layer, initially create an RPL border router in COOJA simulator using Z1 mote. This RPL (Routing Protocol for Low Power and Lossy Networks) router allows the COOJA to communicate with the outer world. Now create another Z1 mote which is loaded by the proposed Server side service layer code. This second mote is able to sense the data (Say, temperature in case of Z1) and also capable to PUSH the sensed data to the cloud. To make the cloud connectivity, these two motes will be within one another's range. Now start the Serial server socket of the RPL Border router. Now the RPL Border router has a connection with the local virtual port. To connect the router with the local machine a Tunnel slip is used in Contiki OS which is named as tunslip6. Use the local Linux terminal for starting the Tunnel slip. If everything is alright, the connection between the server mote and the local host will be established. Now start the simulation, if there is a successful connection the sensed data from the server mote will be pushed to the client on the cloud. For seeing the pushed data the corresponding client should be ON at that time. The sensed data values can be seen at the server side using the log files. So that we can verify the data. For individual evaluation of the server side, Linux log files are used. By accessing these log files the sensor data can be seen and analyzed.

V. FUTURE WORKS

Security is a big concern in cloud data. The data stored in the cloud are susceptible to various threats [5]. Since it is a vast area this project does not deal with the security of the data that are pushed into the cloud data base. In future this can be dealt accordingly.

The sensor data are stored in JSON format. This method has its own merits and demerits. In future, a data base can be provided in order to store and manage the data. Using an adequate data base simplifies the management of huge amount of data obtained from the deployed sensors. By using such a method many useful functionalities can be provided both for the server as well as for the client side.

VI. CONCLUSION

This project attempted to implement the server side service layer for Internet of Things. This follows the cloud computing concept 'Sensing as a Service' model. There is a client side for this part of the project which consists of a web interface and other back end processes to access the services. These two individual parts (i.e., the client side and the server side) need to combined together to form the concept of Service layer in completion. In order to verify the results at the server side log files are used.

REFERENCES

[1] CharithPerera, ArkadyZaslavsky, Peter Christen, DimitriosGeorgakopoulos,*"Sensing as a Service Model for Smart Cities Supported by Internet of Things,"* in Transactions On EmergingTelecommunications Technologies,Trans.Emerging Tel. Tech. 2014.

[2] Maria Rita Palattella, Member, IEEE, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Member, IEEE, Luigi Alfredo Grieco, Senior Member, IEEE, GennaroBoggia, Senior Member, IEEE, and MischaDohler, Senior Member, IEEE, *"Standardized Protocol Stack for the Internet of (Important) Things,"* in IEEE communications surveys & tutorials, vol. 15, no. 3, third quarter 2013.

[3] D.-L. Yang, F. Liu, and Y.-D. Liang, *"A survey of the internet of things,"* in International Conference on Ebusiness Intelligence (ICEBI-2010), ser. Advances in Intelligent Systems Research. Atlantis Press, 2010,pp. 358–366.

[4] Adam Dunkels, Bi¨ornGr¨onvall, Thiemo Voigt, *"Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors,"* Swedish Institute of Computer Science.

[5] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle, *"Vision and challenges for realising the internet of things,"* European Commission Information Society and Media, Tech. Rep., March 2010, http://www.internet-of-things-research. eu/pdf/IoTClusterbook March 2010.

[6] Edward A Lee, Sanjit A Seshia, *"Introduction to Embedded systems – A Cyber Physical System Approach ,"* lulu.com, 2011.