Web of Things an intelligent approach to solve interoperability issues of Internet of Things communication protocols

V. Thirupathi1,2  Dr K. Sagar3
1Research Scholar, OU
2Dept of CSE, S R University, Warangal, Telangana, India
3Dept of CSE Chaitanya Bharathi Institute Technology, Hyderabad, Telangana, India

Email: v.thirupathi@sru.edu.in

Abstract. Internet of Things (IoT) is an emerging technology now in a days. It allows each and every physical thing to communicate with each other through internet. To establish communication among physical things they require some communication protocols like hypertext transfer protocol (http). But these devices have constrained computational resources like RAM and processor speed. Due to constrained resources they cannot able to communicate using http. So they require special communication protocols like CoAP, MQTT and AMQP. Various manufactures can build their products using their proprietary architectures and communication protocols, when they try to communicate problems raised due to proprietary architectures and protocols. This is called interoperability problem. To solve this problem we propose a solution using Web of Things (WoT), WoT enables each and every device can connect to a server as web pages. So that we can access any device through web using internet as simple as we access web pages.

1. Introduction

Internet of Things (IoT) [8] enables every device (Thing) to communicate with each other through internet. Every device/thing has a sensor node which is capable of gathering data, processing and transferring to other nodes with the help of sensors, actuators and communication protocols. Different vendors manufacture variety of devices by using their proprietary architectures and communication protocols. Due to different architectures and communication protocols, devices cannot establish successful communication with other devices. IoT devices have less computational resources like Random Access Memory (RAM) and processor speeds. So they require special protocols at each layer. For instance, 6LowPAN being used at network layer, TCP/UDP at transport layer and COAP/MQTT at application layer.

IoT reference model uses CoAP, MQTT, AMQP, XMPP and DDS protocols at application layer to transfer messages among different devices as communication protocols. If a sender wants to send a message using CoAP protocol, receiver wants to receive using MQTT protocol then communication problem will be raised due to heterogeneity between sender and receiver protocols.

Web of Things (WoT) defines an established group of principles by the W3C consortium to solve the interoperability problems of various IoT (Internet of Things) applications at different levels. Web of Things enables every physical thing to be added to a server. So that it can be accessed from
anywhere through internet. It is a main advantage of WoT over Internet of Things (IoT). Things which are of IoT have some limitations as they use proprietary hardware and software components. When the things uses proprietary components they may or may not connected with each other. So we require a special technology to make connection among all things without making any hardware or software component even the devices are using proprietary tools. A few researchers proposed solutions to solve interoperability problems through middleware technologies, cloud services and web services etc . . . . But these solutions are not scalable.

The WoT (Web of Things) encourages the knowledge of web technologies to provide communication among heterogeneous devices. The advantages of these web technologies are two. First, it helps us to access every device by using tools and technologies related to web. Most of the things makes use of proprietary hardware and software technologies. It can be used to establish communication among the devices which are using heterogeneous communication protocols. So that it enables interoperability. Second, WoT reduces the gap in between physical and virtual world by making use of existing web technology tools in developing web of things (WoT) applications. Now a days people are controlling their air conditioner through social media. Even a flower pot can send a tweet about its status as whether it is dry or wet. Some non-living things like bottles are being added a QR code so that users can send messages to their favourite bottles these messages can be viewed at bottles blog.

1.1 Web of things architecture

Web of Things (WoT) is place where all physical things connected. These things can be accessed through internet as any other web pages. Figure 1 shows WoT architecture [10], it has four layers viz access, find, share and compose. Network layer is responsible for transferring data from one thing to others using communication medium. Things might connected using either WiFi, Bluetooth, zigbee, or internet. But these are constrained devices and IPV4 is limited, so a new protocol being used by these things i.e. 6LoWPAN (IPV6 low power personal area network). It is an IPV6 protocol used for Internet of Things.

i. Access: This layer explains how things connect to the web. Application layer protocols like http, CoAP, MQTT [5], AMQP, XMPP, Web sockets and DDS are used to connect with web server. It allows things to follow REST architecture, as this every resource has its own URI.

ii. Find: Find allows senders to identify required things on the web. The things can be identified through URI and Meta data. Meta data describes everything about the device/thing. Sender must read the Meta data so that he is able to operate a respective device using proper user interface component. If we do not define a proper user interface users will get lot of problems like what to send, where to send and how to send. With the help of semantic data necessary information can be added to the things.

iii. Share: This layer is responsible for exposing web things to others in a secure way. It is very important if the things do not have a security, hackers can easily hack the physical things and access them remotely.

iv. Compose: It explains how physical things can be connected to the web services and cloud. It also explains which web tools are need to be used to construct web things.
Layer 4: Compose
   Web Application Node-ED
      IFTTT

Layer 3: Share
   Social Network  API Tokens
      JWT

Layer 2: Find
   Semantic web  Search Engines
      REST crawlers Encryption

Layer 1: Access
   HTML  HTTP  CoAP
   MQTT  AMQP  DDS
   Websockets

Networked Things
   6LoWPAN, NFC, Bluetooth, Zigbee,
   QR beacons, WiFi

Figure 1 WoT architecture

2. Related work

Paper [1] explained a solution to provide interoperability solution between CoAP and MQTT protocol using “Application Protocol Abstraction Layer” (APAL). As described in APAL framework a message broker and separate plugins are required for each protocol. When a request come from a client/sender, message broker receives that message and intimate all the plugins. Later event dispatcher sends the message to the required plugin. But translation of protocols from one type to other types needs lot of efforts as communication protocols are increasing in type and number. It is suitable when a less number of protocols available. It is not scalable.

Authors in the [2] explained inter communication between CoAP and HTTP protocol, where HTTP is a web protocol and CoAP is an IoT application layer protocol. To translate HTTP requests into CoAP and vice versa, separate proxy servers being used for each protocol. Protocol proxy is a good solution when a few number of protocols exist. It is not scalable due to diverse and proprietary protocols are available.

“Semantic Gateway as Service” (SGS) [3] explains a proxy architecture for multiple protocols. Every client sends data to the same protocol interface i.e CoAP clients sends data to the CoAP interface. From the interfaces, message brokers receives all messages and directs to the respective receivers through routers. But an intelligent message broker is required to route the messages to respective receivers from senders. [4] Discussed a multiprotocol translator which has low latency. It is being implemented using “arrowhead framework” based on “Service Oriented Architecture”.

3. Methodology

Web of Things allows each and every physical thing connect to a server as web pages, so these can be accessed as simple as web pages. Each and every physical thing must be added description is called web thing description. It is in java script object notation (JSON) format. It explains everything about a physical thing i.e. what operations we can perform on a particular physical thing, what are the commands required to operate a thing and how to operate a thing. Web thing description also tells us the uniform resource identifier (URI) [7].

Figure 2 Accessing remote devices

Figure 2 explains how to access/control physical things remotely. Here we try to access the sensor using web client which uses http protocol, where sensor sends its data to the server using MQTT protocol. If observe the devices are operating on different communication protocols like http and MQTT. Irrespective these communication protocols being used by devices, they can communicate with each other with help of Web of Things (WoT) API.

Web of Things adopted REST (Representational state transfer architecture), which uses http protocol to make communication among different web pages. Representational state transfer (REST) architecture aligned with hypertext transfer protocol (HTTP). REST provides its features as follows.

i. Identifying things through URI (Uniform Resource Identifier). Each and every thing/device must consist of URI. These URIs are hierarchically arranged.

URIs can be defined as follows:

Resource= (RS, IR, RB, nf, CID, DD) where set of resources can be represented using RS, IR represents set of resource identifiers, set of root identifiers is RB (RB ⊆ IR).

nf ---- IR ---- RS is a function which maps identifiers to resources. CID is client identifier and DD is a group of data values.

URI= protocol scheme “:” web address / resource path [“?” request query]

Figure 3 depicts how URI address assigned to a physical thing. Here in this example if a room1 has fan1 and air conditioner (AC) they can be shown as follows

URI=https: / home_automation/ room1/ fan1?” on”
ii. REST provides four operational methods they are POST, GET, DELETE and PUT.

iii. Resources can be represented and accessed using various formats like XML, HTML or JSON. Resources sends or receives messages either of the above format.

iv. As we know that HTTP protocol is a stateless protocol. So it cannot track resource identity.

4. Results

We have done an experiment where we try to access physical devices which operating on CoAP and MQTT communication protocols from HTTP (hypertext transfer protocol). Figure 4 shows the communication between MQTT and CoAP through Web socket protocol. Devices have communicated and transferred messages successfully.

The following table shows how much time taken to transfer messages among devices which are using heterogeneous communication protocols. Time considers in micro seconds. Figure 5 and 6 explains how much time taken among different protocols.
Table 1. Time taken for number of requests

| No Of Requests | http | MQTT | CoAP |
|----------------|------|------|------|
| 5              | 1    | 2    | 5    |
| 25             | 25   | 50   | 125  |
| 50             | 50   | 100  | 250  |

Figure 5 Response time

Table 2. Time taken protocol to protocol communication

| Client protocol | http | MQTT | CoAP |
|-----------------|------|------|------|
| http            | 1    | 2    | 5    |
| mqtt            | 2    | 1    | 6    |
| CoAP            | 5    | 6    | 1    |

Figure 6 Protocol to protocol communication
5. Conclusion and future scope

Interoperability is an emerging problem in all aspects as vendor specific communication protocols are getting increased. New technologies need to be invented to solve interoperability issues, this solution must solve all problems. Whenever try to connect heterogeneous devices security [9] problems will be raised. As future extension one can try to provide strong security solutions. To balance flow of traffic, we can use a technology like software defined network (SDN) [6].

6. References

[1] Dessislava Petrova-Antonova, Georgi Andreev and Sylvia Ilieva 2017 Unified connectivity of IoT devices through abstraction of application protocols 7th International Conference Association for Computing Machinery 56-61.
[2] Paul Grace, Brian Pickering and Mike Surridge 2015 Model-driven interoperability: engineering heterogeneous IoT systems. Ann. Telecommun. (0000) 71:141–150
[3] Pratikkumar Desai, Amit Sheth and Pramod Anantharam Semantic Gateway as a Service architecture for IoT Interoperability.
[4] Hasan Derhamy, Jens Eliasson and Jerker Delsing, 2017 IoT Interoperability—On-demand and low latency transparent multiprotocol translator IEEE INTERNET OF THINGS JOURNAL 4(5) 1754-1763.
[5] Thirupathi V and Sagar K 2018 Implementation of home automation system using mqtt protocol and esp32 International Journal of Engineering and Advanced Technology 8(2C2) 111–113.
[6] Thirupathi V, Sandeep C, Naresh Kumar S and Pramod Kumar P 2019 A comprehensive review on sdn architecture, applications and major benifits of SDN International Journal of Advanced Science and Technology 28(20) 607–614.
[7] Saritha B, Chitti S, and Thirupathi V 2018 Constructing a system to monitor and control indoor environment International Journal of Engineering &Technology 7 247–250.
[8] Sandeep C and Thirupathi V 2017, Internet of things based biometric electronic voting machine. International Journal of Research.
[9] Pramod Kumar P, Naresh Kumar S., Thirupathi V. and Sandeep C 2019 QOS and security problems in 4G networks and QOS mechanisms offered by 4G. International Journal of Advanced Science and Technology 28(20) 600-606.
[10] Vinicius A. Barros, Sérgio A. B. Junior and Sarita M. Bruschi 2019 An IoT Multi-Protocol strategy for the interoperability of distinct communication protocols applied to Web of Things. DOI: 10.1145/3323503.3349546