Implementation of LWM2M Protocol in Constrained IoT Devices

P. K. Ghibitha Bebin, T. Gifta Irine Sophiya, T. Vijayanandh

Abstract: ***An emerging Lightweight machine to machine had been indulged with a high speed portable client-server specification. The LWM2M was helpfull for constrained networks. It has systematic machine manipulation also with an invulnerability venture, was supported in IoT applications. Research activities also focus on the server domain was in process in LWM2M. In LWM2M end-nodes are always resource-constrained. The client-side authorized LWM2M functionalities are not only critical and crucial also challenging. To approach the client-side set-up in LWM2M, it has a proper authenticity environment embedded in hand with IoT node. Its interconnection was predetermined to associate with the lightweight protocol stack and response was figured up by the LWM2M v1.0 specification. The usability and effectiveness of Lightweight protocol validated using a real-world application. Building a home automation product is one of the most effective parts is to think about protocols. Thus the LWM2M protocols are one of the efficient ways to communicate to gateways, servers, and sensors.***

Keywords **:** *Light Weight Machine to Machine (LWM2M).*

I. INTRODUCTION

Light Weight Machine to Machine is considered as a protocol used for the device management. The main objective of LWM2M is to progress a quick movable client-server designation to indulge machine to machine services. In LWM2M machine handling perform as a replacement for use in the machine to machine data transformation. Machine handling along with the invulnerability system for IoT applications also uses the same protocol. This suggests the M2M magnification, integrity, and standardization. The encircled M2M that is constrained devices needs observance and administration. It also used to carry the data over numerous transport and porters. Here Application Program Interfaces was used for several purposes such as layout, comparability, observation, invulnerability, firmware update, sustaining and so on. It was indulged in the LWM2M specification. Thus for attaining distinctive captivating for the internet of things.

Revised Manuscript Received on February 27, 2020.

P.K. Ghibitha Bebin, Electronics and Communication Engineering, National Engineering College, Kovilpatti, India. Email: ghibitha@gmail.com

T. Gifta Irine Sophiya, Electronics and Communication Engineering, National Engineering College, Kovilpatti, India. Email: giftairenesophia@gmail.com

T. Vijayanandh M.E, Electronics and Communication Engineering, National Engineering College, Kovilpatti, India. Email: vijayanandh@nec.edu.in

The Constrained Application Protocol fabricates numerous interfaces. This aids to support guidance for distant stationary devices. In the emerging Internet of Things these CoAP accomplished to carry off the connected appliances. For handlying the facility of the distant devices and for putting the devices into operation these CoAP is widely used in LWM2M.

II. FEATURES OF LWM2M PROTOCOL

1. Intelligible system layout with the span of elements which is precise in the LWM2M requirements.
2. Fabricating, improving, canceling, and regaining of elements were provided.
3. It supports Type Length Value, JavaScript Object Notation, and numerous deflating layouts.
4. The LWM2M also supports User Datagram Protocol and Short Message Service.

III. LITERATURE SURVEY

A. Recent survey

In the paper [18], Van der Stok et al. has proposed three new CoAP methods which were handed down for reading and enhancing the new methods. FETCH allows users to read only a part of an object instance. Whereas, PATCH and iPATCH can be used, unlike PUT, to partially update an object. The primary CoAP methods do not admit to acting on numerous elements.

Thus it provides the operator to carry out deeds on numerous elements in a machine by transporting a distinct demand. A reverse interaction model in LWM2M was enabled.

B. Current technology

The automatic control and service advancement for the device to device frameworks and the end-user customer, resource control including distant existence maintenance was used. In contrary to the host concept declares about the overlook of huge editions of connecting devices. It was described here with the machine to machine access based architecture. This case shows the interior framework of the access point and application program interface to operate the end-user and constrained devices.

Collection of web services as the gateway and how to illustrate the frameworks of the automation control and to carry on through the software such as smartphones and any other devices which have recurring programs of intertwining and monitoring the type of their architectures was portrayed.

C. Related works

LWM2M tenant mechanism structure framework was determined. In provinces, the general custom avenue was also defined. The mobilization of LWM2M in accessing the internet, junctions are sited on the clear authorization. It is for authorization of data entitizes to indicator elements. The work of LWM2M architecture provides an efficient end device domain. This was modeled to gain the consequent primary concerns to repose the activity enhancement and fusion of buffer stage defined features:

1. The acceptable device controlling performance from the file management system.
3. Error-free Application Program Interfaces
4. Compact, modular and flexible architecture
5. Programmed without trouble
6. Easy to port and integrate
7. Constrained devices become part of the increasing inefficiency.

D. Industry standard

Keeping constrained devices in mind, for the machine to machine communication, OMA Specification works developed the LWM2M standard. The LWM2M standard provides low-cost device management for service providers and vendors. For wireless connections, the service enablement protocol architectural principles function well. Lightweight M2M has been specially designed for the management of resource-constrained devices, security challenges, and cross-standard interoperability.

For creating LWM2M the AV System has been actively participating as a member of OMA Spec Works. An LWM2M server works under the AV System's CoIoTe IoT Device Management platform takes advantage of the key functionalities of the OMA LWM2M to implement a leading solution suitable and virtually for the whole of the IoT markets. AV system was an open-source software. It created an LWM2M client. It allows the easy implementation to support the LWM2M protocol in any device.

IV. PROBLEM STATEMENT

1. LWM2M stack, code name IOWA, is a reference implementation of the OMA and IPSO standards, enabling wide range of services for IoT solutions to be deployed quickly in a secure fashion.
2. The LWM2M stack is commercially deployable now and is easily installed in embedded devices, gateways and server infrastructures.
3. It is adapted to any compilation tool chain and
4. The LWM2M device can help to integrate the LWM2M technology into IoT solutions.

V. BLOCK DIAGRAM

A. Description

There are four interfaces available in LWM2M architecture:

1) Bootstrap interface
2) Registration interface
3) Management interface
4) Information Reporting.

The fundamental of transmitting the data by using transfer custom across Internet Protocol uses CoAP in LWM2M interfaces. For the realization of these interfaces, a CoAP message header along with a small subset of options was used. LWM2M needs an object instance or resource URI path to identify the interface. The query string needs URI components and path segments in LWM2M. It is used for the request and to encode in the URI-Path option. It aids to proceed with self-referential data. The requisition was detached from the packing and the whole interrogation criterion is fixed in the URI request. Likewise, with the help of CoAP methods, the LWM2M operations were get mapped with each interface. In URI request every notifies operations either be confirmable or not confirmable but generally in LWM2M notify operations were not confirmable.

B. Block diagram

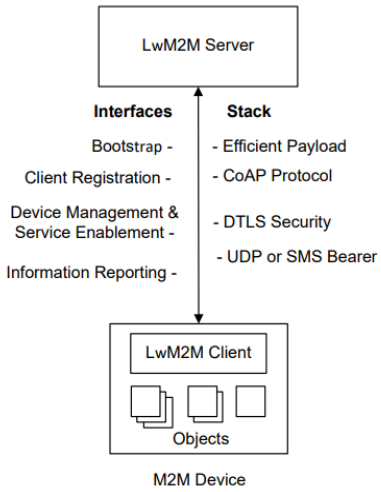

Fig 4.1: Block diagram

C. Interfaces

Each interface the methods were classified into two methods, they were uplink as well as downlink methods. Here, in the transport layer bindings and encoding, each interface methods, are determined, then mapped to protocol mechanisms.

D. Bootstrap interface

In the Bootstrap attachment, the methods are of two types, an uplink and downlink methods, “Bootstrap-postulation” and “observation”, “creating”, “cancelling” and “Bootstrap-exhaust” respectively.

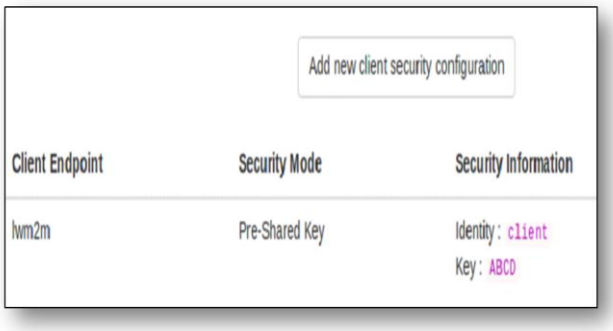

Fig 4.2 Bootstrap interface

LWM2M Client gets registered with one or more LWM2M Servers each object was initialized with the help of uplink and downlink operations. On the bootstrap interface, the "Write" operation, writes the value included in the payload. In transmission mode when the Server was addressing the LWM2M Client about the bootstrap information, the Servers MUST operation informs the LWM2M Client. After this, the information is sent over to the server as a bootstrap finish command.

E. Client registration

Here, the client registration interface have only one operations, uplink operations, "Registration", "Update" and "De-register".

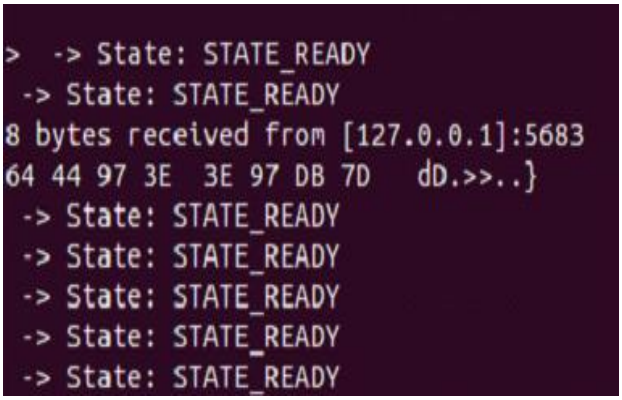

Fig 4.3 Client registration

This is registered with one or more LWM2M Servers with the help of this interface. This device management and service enablement maintain each registration. It also de-register from an LWM2M Server. The "Register" operation performs when registration takes place in the LWM2M client. It provides the supported Objects and existing Object Instances in the LWM2M server as well as Endpoint Client Name. The registration and communications sessions were maintained by LWM2M client with each LWM2M Server. An update of its registration information is periodically carried off by the "Update" operation.

F. Resource maintainance and supply improvement

In the Device Management and service enablement interface, the operations are downlink operations called "examination", "observing", "cancelling", "developing", "implementing", "observation-character", and "Determining".

Fig 4.4 Device management and service enablement

The present values are examined by the use of "Read" operation; To discover character and to find out assets which are accomplished in a certain Object is operated by "Discover" operation; For updating the values the "Write" operation is used; The change in attribute values is made by "Write-Attributes" and an action is initiated by the "Execute" operation. The create or delete Instances are operated by the "Create" and "Delete" operations.

G. Information Reporting

Information Reporting interface has two operations were downlink operations and uplink operation, "Observe" or "Cancel Observation" and "Notify".

Fig 4.5 Information reporting

Both downlink and uplink operation has parameters such as "Observe" or "Cancel Observation" and "Notify". The host sends a current assessment correlated to a value on the customer with the help of this interface. The operations and interface relationships were listed in Table 1. If any changes occur in the LWM2M server, the Information Reporting Interface performs to observer changes in a resource on a registered LWM2M Client. The values are notified in receiving notifications. An "Observe" or "Observe-Composite" operation initiated to observation relationship in the LWM2M Client for an Object. When the performance ends "cancel observation" or "cancel observation-composite" operation is performed.

Table- I: Relationship between operations and interfaces

Interface	Direction	Operation
Bootstrap	Uplink	Bootstrap request
Bootstrap	Downlink	Write, discover, delete, bootstrap-finish
Client registration	Uplink	Register, update, de-register

Device management and service enablement	Downlink	Create, read, write, delete, execute, write attributes, discover
Information reporting	Downlink	Observe, cancel observation
Information reporting	Uplink	Notify

VI. LWM2M STANDARD

Lightweight M2M solution with other device management standards is called an Enabler. The M2M device and M2M service or platform or application consist of M2M Enabler. This enabler works under CoAP protocol with User datagram protocol/short message service transport bindings which was used by a client-server architecture. At constrained devices, the bandwidth is used efficiently with the targeted enabler. More powerful embedded devices were benefited from efficient usage of the enabler and also communication was utilized by LWM2M.

In the LWM2M standard, the LWM2M communication model is used. Each LWM2M client can use each of the object's communication model requires security. “Get”, “Put”, “Post”, and “Delete” are the CoAP methods. These parameters have bindings over User Defined Protocol or Short Message Service. It works under the transport layer. This protocol makes the constrained devices to organize logically by the given resources. Some resources in the LWM2M enabler contain firmware update which is used for the firmware object.

VII. LWM2M ARCHITECTURE

A. IoT integration custom aperture

Lightweight IoT connectivity protocol stack consists of the following layers:

1. The machine control and supply advancement operations were implemented by LWM2M Protocol Engine Core and the application programming interfaces were provided(APIs).

2. A well defined generic APIs were implemented and provided by the interface layer to contact numerous intermediate layers such as Resources, the primitive integration layer (CoAP) and from the LWM2M client protocol platform.

a. CoAP contains message header, request or response codes, message options, and retransmission mechanisms.

b. For security purpose, the exchange of messages between the server and client was provided by datagram transport layer security.

c. UDP and SMS binding was mandatory.

B. LWM2M client engine

Managing other protocol necessary core functionalities were implemented by LWM2M management with the interface of the LWm2M client engine. The application platform requirements were created for the managing protocol. The bootstrapping configuration was responsible for device management. The registration, automatic registration update and deregistration procedures with the configured LWM2M servers were implemented by the Client Registration Management Module. The automatic notification was observed by "Observe" notifications, automatically the value was added to the module and also it observes the value of LWM2M servers as defined in. Create, delete, read, write, execute, discover, etc. such operations were enabled by the interfaces of device management and service enablement

C. Interface layer

The open mobile alliance introduces a third party which registers the value of the object that makes the interfaces to update or modify resources value in the LWM2M server and client. This OMA standard object implements the device interface. To use the underlying constrained application protocol implementation available in the Connectivity Interface layer was provided. The request from the registered Host provides the interfaces which can make use of objectives of firmware update executions. Selective configuration was supported by Object/Device Configuration and the use of application of particular Objects was available and also supported by Object/Device Configuration.

D. LWM2M client interface

In the North part of the network for permitting the Light Weight protocol, it practices the programming to enable customers' performance in the machine by using this intermediate layer implements APIs. In the South-side Interface, the underlying CoAP protocol stack enables the platform-dependent modules.

E. LWM2M objects

The Open Mobile Alliance provides us various objects. Each object has been used for unique purposes. Every object has each resource. The objects can be edited by the editor in the OMA standard. The internet protocol on smart objects also provided objects based on the Open Mobile Alliance. These objects were created to achieve efficient communication by transmitting the data between the devices and application software. Here the common set of objects can communicate with the devices. The objects are standard independent format protocol. Object defines anything that provides data about any device. Each object is defined by the object ID and each of the objects has resources which contain resource ID.

F. LWM2M test compliance

The interoperability test cases were tested against the LWM2M client protocol stack. For deriving benefit from the constrained application protocol, the LWM2M test enabler is used to carry out the LWM2M protocol and objects related to CoAP.

VIII. EXECUTION OF HOME AUTOMATION

A. LWM2M client

We use the Linux environment as the recipient(client) device. This enables the recipient to manage elaborate connectivity loads that upright the protection. Here we use the ubuntu machine to check the performance of the LWM2M client. The connectivity load recurring straight through the CoAP layer. This LWM2M client sample application for the Leshan implements the lightweight M2M library and establishes the IPv4 and IPv6 connections to the LWM2M server using OMA lightweight M2M technical specifications.

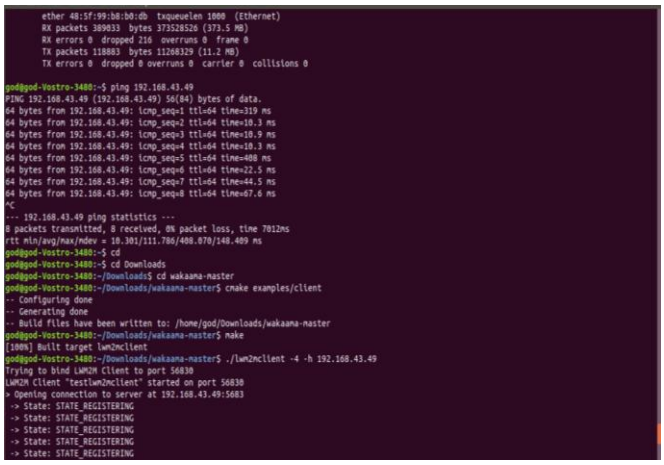

Fig 7.1 LWM2M client setup

B. LWM2M server

Leshan works in java client and server implementation. The Leshan contains both client and server libraries that can be used to develop the LWM2M client and server using the java programming language. To view, the Leshan software official repository uses the GitHub page. One can couple on Leshan User Interface from any common web browser using HTTP://<Server-IP_addr>:8080. The list of connected clients is responsible for a very simple User Interface form Leshan and interacts with client resources.

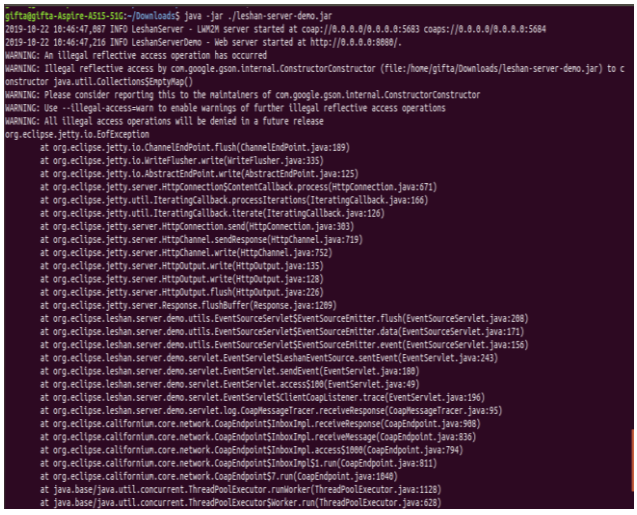

Fig 7.3. Leshan server

C. Development Kit

The Development kit tool is used to explore the LWM2M protocol and to test LWM2M server implementations. The OMA developer tool kit contains a list of tools that aim to assist developers to understand and use OMA specifications. OMA tools such as technical specifications, client/server tutorial, object/resources editors and so on. It allows end-users to interact with the messages sent between the client and server, DevKit. This Dev Kit contains the list of platforms with sensors, IDE, libraries, that can be used with LWM2M.

D. Application Scenario

In-home automation the crucial thing is controlling the various functionalities of electronic devices. This can be managed by using the LWM2M protocol. The LWM2M used to handle and control device management. Home automation was demonstrated for managing a home appliance with the help of the internet of things.

An element was created which was considered as the resource named as a light resource. It is tied together to the resourceful sensor of the recipient(client) device. The recipient(client) device was enrolled with the host (server – Leshan) and to examine the scene of the resources in the client called light resource. In this case, the LWM2M client device was successfully able to operate the bulb from far distant which was done with the help of some intermediate like web interface maintained by the Leshan server.

The goal is to operate the bulb from a distant place or area.

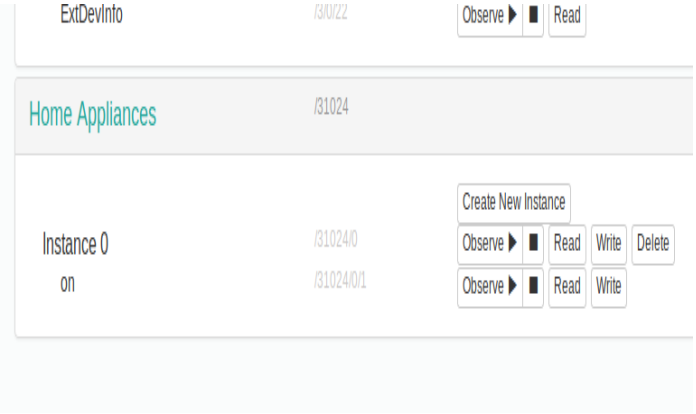

Fig 7.6: Monitoring Home Appliances

Raspberry pi was connected to the LWM2M server which is streaming on a Ubuntu, which is correlated to the host. LWM2M server was registered by LWM2M client, to control the light resource with the help of some intermediate led by the host(LWM2M server) or on 8080 TCP port by a standalone tablet. In LWM2M to controlling the light resource, we can also monitor the performance of the light bulb using the Leshan LWM2M server.

IX. RESULTS

The working of home automation was done using the LWM2M protocol. Here “internet of things” was used for providing service enablement and device management applications and to connect the appliances. The IoT is used for switch on, configured and to update with software. Home automation can be done without the use of microcontroller also. Here the Home automation is done with the help of the Leshan software.

The home automation was demonstrated by the help of OMA LWM2M DevKit. The testing purpose of home automation was provided with the help of Leshan software were client, server and bootstrap were demonstrated.

To initiate the LWM2M server the local address was configured, to load the models and register a listener. The OMA DevKit supports the OMA Lightweight M2M protocol by enabling manual interaction with an LWM2M Server directly from the Web browser. In this way, we explored the home automation.

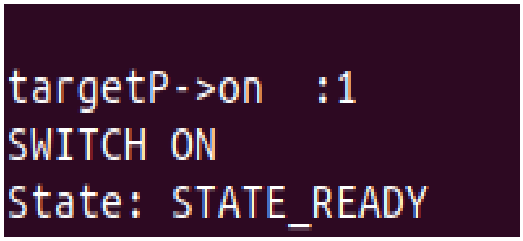

Fig 9.1: Information reporting on server side

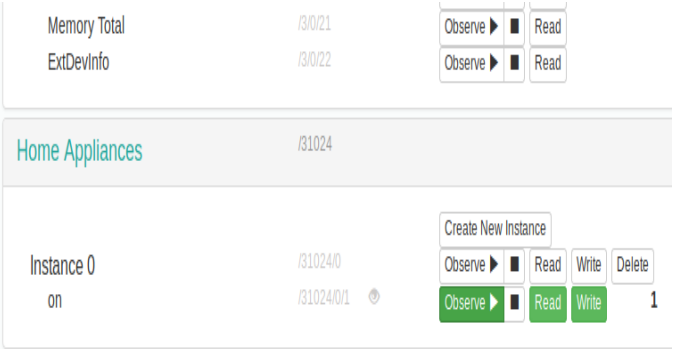

Fig 9.2 a: Home appliances when switch is ON

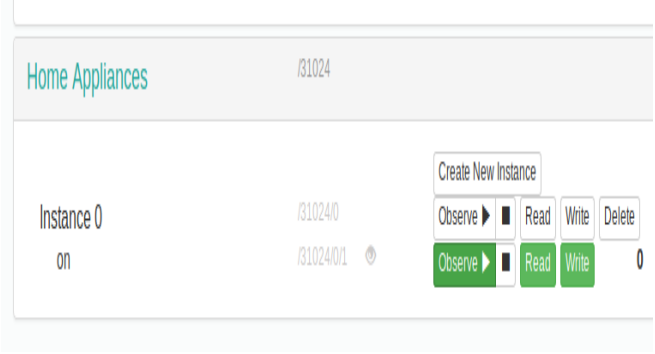

Fig 9.2 b: Home appliances when switch is OFF

X. CONCLUSION

In Today's world due to the onrush of creating automation innovation every industry had been working towards to make the life easy and simpler by implementing automation. Today automatic frameworks are being favored over manual frameworks. So here we developed home automation using LWM2M protocol which helps your home to perform many tasks automatically. The IoT based Home automation is used in LWM2M. The home automation in managing the light bulb was our first step to make electronics devices to work automatically. This will also be monitored by the LWM2M client side. The Switch "ON" and " OFF" was our major concept for home automation using LWM2M. The easy-peasy advantage of a smart home is benevolence and benefit. The main thing about home automation is controlling every device of your home should lie in control of your smartphone and it should be manageable.

REFERENCES

1. Z.Sheng et al, "Lightweight Management of Resource Constrained Sensor Devices in Internet-of-Things," IEEE Internet of Things Journal, April 2015.
2. S.Datta and C.Bonnet, "A Lightweight Framework for Efficient M2M Device Management in oneM2M Architecture," in Proc. IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing ISSNIP 2015, April 2015, Singapore.
3. A.Sehgal et.al, "Management of resource constrained devices in the internet of things," IEEE Communications Magazine, vol.50, no.12, pp.144,149, December 2012.
4. G.Klas, F.Rodermund, Z.Shelby, S.Akhouri and J. Höller, "Lightweight M2M: Enabling Device Management and Applications for the Internet of Things," White Paper, February 2014, available online at http://community.arm.com/docs/DOC-8284.
5. The Constrained Application Protocol (CoAP), IETF RFC 7252, June 2014, available online at https://tools.ietf.org/html/rfc7252.
6. K.Hartke, "Observing Resources in CoAP", draft-ietf-core-observe16, work in progress, December 2014. [11] TI CC2538 - Wireless Microcontroller System-On-Chip for 2.4 GHz IEEE 802.15.4, 6LoWPAN, and ZigBee Applications, available online at http://www.ti.com/product/cc2538.
7. Contiki - the open source OS for the Internet of Things; the official git repository available online at http://www.contiki-os.org.
8. RPL border router example on Ubuntu OS, available online at https://github.com/contiki-os/contiki/tree/master/examples/ipv6/rplborder-router.
9. Leshan an OMA LWM2M implementation in Java, available online at https://github.com/eclipse/leshan.
10. OMA LWM2M DevKit extension for Mozilla Firefox, available online at https://addons.mozilla.org/en-US/firefox/addon/omaLWM2M-DevKit/
11. Terminology for Constrained-Node Networks, IETF RFC 7228, available online at https://tools.ietf.org/html/rfc7228.
12. P.van der Stok, C.Bormann and A.Sehgal, "PATCH and FETCH Methods for the Constrained Application Protocol (CoAP), " April 2017

AUTHORS PROFILE

P.K. Ghibitha Bebin, was born in Nagercoil in 1998 was currently pursuing her Bachelor of Engineering in National Engineering College, Kovilpatti, India. She attended many technical workshops and participated in many events conducted by different college. She was an active member of IEEE. Her area of interest is Networking. she completed her mini project in the domain of networks

T. Gifta Irine Sophiya was born in Tirunelveli in 1999 was currently pursuing her Bachelor of Engineering in National Engineering College,Kovilpatti, India. She attended many technical workshops and participated in many events conducted by different college. She was an active member in IETE. Her area of interest is Networking. She completed her mini project in the domain of Networks

T. Vijayanandh received the B.E. degree in Electronics and Communication Engineering(ECE) from National Engineering College, Kovilpatti, Tamil Nadu, India ,in 1999, M.E. degree in Communication Systems from Mepco Schlenk Engineering College, Sivakasi, India, in 2002. He is currently working as Assistant Professor(Senior Grade), Department of Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamil Nadu, India. His current research interests include Medical Image processing, Multimedia compression and Information security. He is a Life Member of IETE, New and has published papers in National/International journals.