

RPL AND COAP PROTOCOLS, EXPERIMENTAL ANALYSIS FOR IOT: A CASE STUDY

Sharwari Satish Solapure and Harish H.Kenchannavar

Assistant Professor, CSE Department, Walchand College of Engineering, Sangli - Maharashtra –India.

Professor and HoD of IS Department, Gogate Institute of Technology, Belgavi-karnataka-India.

ABSTRACT

Internet of Things (IoT) in recent days playing a vital role in networking related applications. However, there are several protocols available for building IoT applications, but RPL and CoAP are important protocols. There is a customized protocol requirement for specific IoT applications, while working on specific research problems. Further, adequate platforms are required to evaluate the performance of these protocols. These platforms need to be configured for the protocol, which is very crucial and time-consuming task. At present, there is no collective and precise information available to carry out this work. This paper discusses two different open source platforms available for IoT. Also, various IoT research ideas need to design of IoT protocols. A few IoT communication technologies are mentioned in the paper. The detail analysis of, two common protocols, namely Routing Protocol for Low-Power Lossy Networks (RPL) and Constrained Application layer protocol (CoAP) is carried out with respect to latency delay and packet delivery ratio. The results, discussion and conclusion presented in this paper are useful for researchers, who are interested to work with IoT protocols and standards.

KEYWORDS

6LoWPAN, IoT, RPL, CoAP, WSN, 6LoWPAN

1 INTRODUCTION

Wireless Sensor Network (WSN) [4] technologies exist across a number of fields in today's modern lifestyle. The communicating, actuating network of sensors and actuators connected to the Internet is known as Internet of Things (IoT) [1]. It is the collaboration of physical and virtual world together mostly without human interaction. The different essences of this basic are very famous in the research community such as: Industrial IoT (IIoT) [1], Internet of Everything (IoE) [1], etc. It is an emergent area, which connects all types of vehicles, industrial components, and sensors to the Internet with the help of cloud process. It has the capacity to transform everybody's life. IoT devices use both wired and wireless for communication. Numerous smart applications of IoT available for homes, cities, logistics and agriculture [2]. The smart device intelligence makes IoT vision very similar to reality. IoT mechanism depends on a reliable less-power wireless technology and a communication protocol stack for the devices. The IETF (Internet Engineering Task Force) [1] has defined a protocol stack for a limited resource devices. It includes IPv6 low-power wireless personal area network (6LoWPAN) ,IEEE802.15.4, RPL [23], and CoAP [24][21]. 6LoWPAN architectures are of different types as described in [3][4]. The Fig.2.1 in paper [4] depicts 3-layer architecture model. In Fig.2.2 of [4] other IoT architectures of 5 layers are

shown. The Software defined wireless network (SDWN) [4] is best suitable for low mobility. A ZigBee and 6LoWPAN perform better in dynamic situation [4].

IoT devices have constraint of memory and storage. Thus, the operating system(OS) used by IoT devices should be aware of all these constraints. IoT OS provides the software interaction techniques and without this IoT devices would be just a dumb device. The tiny tendencies of IoT gadgets are not suitable for the heavyweight protocols. Therefore, an aim is to plan green and reliable lightweight protocols. IoT protocols for the functions: routing, QoS, resource allocation, protection, interoperability needs to be light-weight, scalable and reliable. The limitations of IoT explores new openings towards different research areas [2] for researchers. A scope to upgrade functionalities of existing protocols in each area. The evaluation of newly designed IoT protocols prior to their deployment requires proper testing and evaluation using various tools. An IoT research process starts with an innovative ideas and ends with real-time deployment. It involves both simulated and physical elements. Then there is a requirement to develop a scheme for analyzing real time methodology using simulation and experimentation on a test-bed. The aim of this paper is to just give a good prerequisite to researchers for their painstaking research in IoT protocols and standards.

This article highlights a technical discussion, which will help the researcher to select a proper platform and do research work efficiently. Section 2 explains the need of designing/implementing new protocols & gives an overview of IoT research related ideas. Section 3 discuss in detail survey of IoT, and its most useful protocols. In Section 4 IoT protocol details are explained. Section 5 gives details of IoT platforms. Results & performance evaluation of selected protocols on both platforms are open in Section 6. Thereafter Section 7 concludes the paper.

2. NEED OF DESIGNING NEW PROTOCOLS

Research in IoT demands, modification in existing IoT protocols/algorithms. In today's world, everybody is interested in smart applications. To make an application user friendly is not an easy job. To develop a smart application, involvement of wireless sensor devices, communication protocols, power consumption and smart machine learning algorithms is necessary. According to this, there are some research areas [1] [2] where the researcher needs to focus on redesign and analyze protocols. Some of the research areas are listed as follows:

- I. Deployment of Internet Protocol (IPv6) [1]: IoT growth is very difficult to handle with IPv4[1] addressing due to its limitation of address space. They need advance technologies such as IPv6. This protocol makes the management of the network easier due to large address space, auto configuration capabilities and extra feature of security.
- II. Sensor energy: sensors are battery operated independent devices. The replacing of batteries in the billions of deployed devices across the world would not an easy task. The sensors should generate electricity from surrounding environment such as the vibration of light/sound, sunlight and airflow etc.
- III. Heterogeneous Things: IoT devices are mostly dissimilar in nature with respect to data location, data collection and data storage. It is a tough challenge to build a protocol, which supports communication between these heterogeneous devices
- IV. Power: The IoT devices use wireless communication. Demanding and continued research efforts have been dedicated for sensor monitoring type applications [6]. These types of

applications need energy efficient protocols for communication.

- V. Real-time Solution: It is too difficult to execute the ‘Anytime, Anywhere, Anything’ concept of IoT in reality. Real-time systems need to be executed efficiently to significantly respond with respect to time.
- VI. Intelligence: Machine to Machine communication significance in Smart technologies needs to be highly intellectual to handle automatic systems. The cognition is talented by cognitive computing, artificial intelligence and machine learning techniques.
- VII. Secrecy: The security arrangements for each workspace must be established as per the present demand and requirement.
- VIII Addressing and Identification: All devices are interconnected with each other and it makes them vulnerable to malicious attacks. The various addressing and configuration problems are handled by designing various tiny protocols and schemes.

3. LITERATURE SURVEY

3.1 GENERAL SURVEY

IoT middleware makes connections of sensors and actuators to the net. These middleware has to satisfy the numerous needs of different stakeholders (device suppliers, designers, and consumers) in the system. The evaluation of these platforms is done accordingly in [3]. The evaluation always focused on user-friendly options than the underlying technologies. The Model Networks have a broad implementation in testing NGN-services [5]. IoT is widely used in a telecommunication field. Therefore, it is suitable to build IoT application on model networks. This article [5] have presented the structure of the model network. The model network is shown in Fig 1 of the paper [5]. The model network is made up of five segments as follows: self-organizing network, flying ubiquitous sensor network (FUSN)[5], Indoor positioning system, Software-defined networking (SDN) [5] and Cloud IoT-platform. The network design of devices with limited resources is a big deal. The industrial authorities will offer a network infrastructure for real-world applications. A summary of commercial ecosystem, detail strategy, corporate device handling techniques, and most recent research movement used for WSN system is presented [6]. The efficient approach for infrastructure is a cross-layer design.

In IoT various communication media and protocols for data are usually unsuited for each other and they need coordination. An increase in the data usage creates a huge data volume and it is a big issue in IoT. The necessities for real time IoT analysis is not considered by an existing analytics network. The survey examines the cutting edge of the analytical network best practices for real time IoT [7]. The essentials of the live IoT evaluation, software platforms and various use cases are described by the author. It also explains the limitations of the network methodologies. Further investigation guidelines towards in time IoT analytics are mentioned in the paper. The Fig.3.1.3 explains in brief the architecture of IoT[7]. It is a connection of sensing, distributing and analytics network.

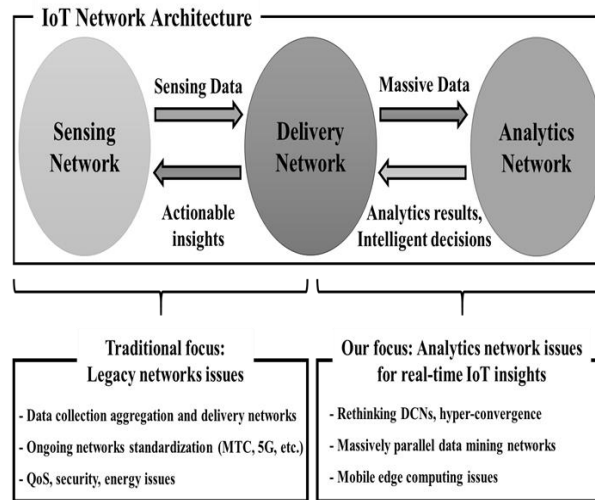


Fig.3.1.3 The IoT Network Architecture

3.2 PROTOCOL SURVEY

3.2.1 RPL

A smart city idea is built with the help of monitoring information gathered by WSN. A new technique to overcome limitations, by combining sensing technologies in a cost effective way is proposed in [8].

The paper [9] discussed a clustering function. It is implemented on each layer by considering routing metric like ETX and energy consumption for hierarchical routing.

The main focus is on two main constraints of the network, i.e. packet loss and power depletion. The mentioned system [10] is using the following steps to solve the problem:

1. Make use of Context aware objective feature, which computes the status of the next node
2. Design of Routing metric, considering the status of utilization of power and queue.
3. Selection mechanism of new parent.

In the battery powered WSN, energy is an important resource. An improved and relatively balanced routing protocol IRPL has overcome the drawback of RPL[11]. An efficient clustering algorithm is used for the protocol design. RPL, is a distance-vector protocol for IoT routing. It builds a graph called DODAG (Destination Oriented Directed Acyclic Graph) [12], which establish the paths towards the sink. DODAG regulates the route based on specific metrics of the objective function. The paper [12] mentions an evaluation of RPL performance, with Cooja simulator [25], with respect to the number of sinks in the network. This change affects the RPL performance. The packet loss and energy consumption are minimized. The packet delivery ratio(PDR) and throughput is maximized [12]. The calculation of path using a specific method in LLNs between a Border Router and the nodes for QoS constraints is mentioned in [13]. Some modification in RPL protocol is proposed here. A parameterized polynomial algorithms are evaluated in the term of execution time, scalability and the number of missing nodes. The quality of the paths from the Border Router to each node is built in DODAG.

3.2.2 CoAP

IoT is based on the sensor devices, which are low powered and are of different clock speed. Due to different operating frequency synchronization techniques [14] are in high demand. The CoAP protocol regulated by IETF and it is widely applicable for home automation. Several application layer protocols are introduced for the resource-constrained network. The comparison of application layer protocols used in IoT is done [15]. The main emphasis is on four protocols, CoAP, MQTT, XMPP, and Web Socket [15]. Their performance evaluation is done for smart parking application by open source software for response time by changing traffic load. In recent days, smart parking has gained a great demand. People waste lot of energy and time to find an empty slot for parking. To solve such a real time problem IoT technique is used. Smarter IoT Devices are used to forward data in networking. Reservation and navigation based parking system is proposed in [16]. In this system lots of sensor device transmits data to the cloud using IoT technology. Genetic algorithm is used to find the closest free slot for parking [16]. CoAP enables device management over the 6LoWPAN. It allows easy accessing of wireless resource-constrained network [17]. It is too much time consuming and a waste of resources to find out empty parking slot manually. A new idea where all the parking area is centralized and using android application a mobile user can reach into the empty parking area is described [19]. A real time parking empty slot detection using Convolution neural network (CNN) is proposed [18]. This technique is very efficient in most of the weather conditions like low intensity of light, shadow and rainy season. The setup is tested against many scenarios. An author has utilized open source dataset, which contains images of parking area taken in different light condition on different occasions with different views [18].

4. DETAILS OF IOT PROTOCOLS

IoT protocols are divided into four main streams: application, service related, infrastructure and other significant protocols. These all may not be used by every IoT application. The IoT Protocol Stack and its protocols are shown in Fig 4.1 and mentioned as follows [20]:

Physical and Data Link Layer: The most familiar embedded methods are [20]: GSM, 3G, LTE, 4G Ethernet, WiFi, Serial with point-to-point protocol.

Network Layer: It provides connection between different networks and between the other layers. It provides connection with an ubiquitous IP addressing.

Transport Layer: This layer is using two protocols such as TCP and UDP [20]. TCP is used for Web interaction: e-mail, browsing etc. Use of TCP for an embedded system is overloaded.

UDP is having a new space in far-off control applications and for real-time data applications. In IoT applications gateway is used for connection of IoT devices to the Internet.

Application layer Protocols [20]:

- 1) HTTPS [20] and Websockets are used in the payload to deliver XML/JavaScript Object Notation.
- 2) HTTP: It is for client-server communication. HTTP is preferred when only one client is associated with IoT device.
- 3) WebSocket: It is utilized for full-duplex communication. It uses HTML 5 specifications. This standard handles complication in bi-directional communication and connection management of the device.
- 4) XMPP [20]: Jabber open source community developed it. It supports a decentralized messaging

system which is open, secure, spam free. Its main purpose is for instant messaging.

- 5) CoAP[20]: Tiny protocol, for limited power and controlled network by IETF. It is a best choice for low power devices.
- 6) MQTT- MQ Telemetry Transport (MQTT) [20]: It is basically for minimum resource devices, high-latency and low- bandwidth networks. Embedded devices and network links using publish-subscribe communication to applications or middleware layer is handled by this protocol.

Application protocol		CoAP	AMQP	MQTT	MQTT. SN	XMPP	HTTP	Reset
Service Discovery		mDNS			DNS-SD			
Infrastructure Protocol	Routing protocol	RPL						
	Network Layer	6LoWPAN				IPv4/IPv6		
	Link Layer	IEEE 802.15.4						
	Physical / Device Layer	LTE-A	EPC Global	IEEE 802.15.4	Z-WAVE			
Influential Protocol		IEEE 188.3.IPSec				IEEE 1905.1		

Fig 4.1 IoT Protocols Stack

Discovery protocols:

- 1) Multicast Domain Name System(mDNS) [20]: Name Resolution service is needed for IoT. mDNS is used to handle multicast as well as unicast DNS server. mDNS is very flexible and best for embedded Internet-based devices.
- 2) Domain Name System Service Discovery(DNS-SD)[20]:It is a distinct arrangement of essential facilities by clients using mDNS. Clients can determine a set of preferred services of DNS messages with the help of this protocol.

Infrastructure Protocols:

- 1) RPL[20]: The IETF-ROLL standardized a routing protocol for IPv6 and minimal resource devices called RPL. RPL is very efficient to build a dominant topology above lossy links. It supports point, point-to-multipoint and multipoint-to-point communication.
- 2) 6LowPAN[20]: It has some features like limited packet size, different address lengths, and minimum bandwidth.IPv6 packets fits to the IEEE 802.15.4 specifications. It is fulfilled by 6LowPAN protocols.
- 3) IEEE 802.15.4[20]: For low-rate wireless private area networks (LR-WPAN) two sub-layers physical layer (PHY) and Medium Access Control (MAC) are mentioned.It is having features such as: low power consumption, less data rate, minimum cost, and high throughput.Due to this it is highly recommended for its use in IoT, M2M, and WSN[20].
- 4) Bluetooth Low Energy: It usages is for a small range,less power and long time radio operation (even for years) beside classic version. All characteristics, makes it more useful in IoT applications.
- 5) EPC-Electronic Product Code [20]: It is a distinctive identity reside on an RFID tag. Supply chain management also uses it for identification. EPC global body is playing an important role in the development of EPC, RFID technology and standards.
- 6) LTE-A (Long Term Evolution-advanced) [20]: For IoT and Machine-Type -Communications

(MTC) , it is a set of cellular communication protocols [20]. It outperforms for service cost and scalability.

- 7) Z-Wave: It is a low-power wireless communication protocol, especially for Home Automation Networks (HAN)[20]. This is improved by Z-Wave Alliance and is made for tiny data transmission applications such as home appliance control, smart energy, fire detection and wearable health.

As IoT is a collection of technologies, and each technology leads the IoT world. LoRaWAN and SigFox [29] are the two major networking landscapes and players in long-range, low-power networks (LPWAN). LPWAN technologies are well-suited for future communications aspects, some of them are listed as follows[29][30]:

- 1) NB-IoT and LTE-M: LTE-M is LPWAN alternatives based on standard LTE connectivity[30]. NB-IoT is another 3GPP build , challenging the Sigfox as well as LoRa. It is distinct from LTE-M as it operates different than LTE. NB-IoT utilizes less power with minimum cost, best suited for smart meters. LTE-M has benefits in roaming applications such as vehicles or drones. LTE-M has higher data rates.
- 2) LoRa[31]: It is developed by an open source association. It is a wireless technology for low-power low-data-rate and long-range applications need. Most of the functionality is similar to SigFox[31]. Different frequency channels, different data rates and encoded packets are used to distribute information.
- 3) SigFox: It is used for extended range using slow modulation rate. Due to this unique design, SigFox is an excellent option for small data requirement applications. Less data, less energy consumption, hence longer battery life. Some of the applications are as: parking sensors, water meters, or smart garbage cans. Its design fulfills the requirements of these applications such as: low device cost and high network plus battery capacity.
- 4) Weightless: It is a special interest group for various techniques such as: Weightless-W, N, N/NWave, P Weightless-N/NWave: Like SigFox, it's best for sensor-based networks, temperature readings, tank level monitoring, smart metering, and other such applications.

5. INTRODUCTION OF PLATFORMS

5.1 CONTIKI AND COOJA

Contiki [25][26]:

Contiki: Contiki is an open source, extremely portable, multi-tasking operating system. It is an embedded system, which is memory efficient. It is designed for microcontroller with very less amounts of memory (2 KB RAM, 40 KB ROM). It follows an IETF standard protocol stack. A group of developers from industry and academia have developed it. It has been extensively used in the industry.

This OS is used in numerous commercial and non-commercial applications: street light network, electrical power meter network, energy meter, many monitoring applications: industrial, radiation, remote etc. It is providing three network stacks: IPv4, IPv6, and Rime. The uIP TCP/IP stack is for IPv4 and IPv6 networking [25]. Rime stack used for low-power wireless networks using a lightweight protocol stack.

It has organized network modules in one stack: Network Protocol stack (NETSTACK) [25]. It handles the functioning of traditional OSI Layers. The mapping of layers with this Network

Protocol Stack is as follows [25]:

1. Network - NETSTACK_NETWORK
2. MAC -NETSTACK_MAC
3. RDC (Radio Duty Cycling) - NETSTACK_RDC
4. Radio layer - NETSTACK_RADIO

1. Network & Routing Layer: Contiki forms a wireless IPv6 network using routing protocol RPL. The network layer is divided into two sub layers, the upper IPv6 layer and the lower adaption layer for its deployment. The Adaptation Layer offers IPv6 and UDP header fragmentation and compression techniques.

2. MAC Layer[25]: There exist two types of MAC protocols in ContikiOS: which are named as nullmac and csma. The first one simply calls the appropriate Radio Duty Cycle (RDC) functions[25]. The second one, implements addressing, sequence number and retransmissions. Carrier-Sense Medium Access (CSMA), the deign in Contiki OS does not rely on the carrier sensing, as the medium access is handled by RDC protocol.

3. RDC (Radio Duty Cycling) layer [25]: It is the simplest layer in the IoT/IP stack. It mainly handles wireless media communication. It avoids collisions or back-off, if there is traffic. Contiki has three different duty cycling mechanisms, such as ContikiMAC, X-MAC and Low-Power Probing (LPP) [25]. ContikiMAC with low-power listening technique gives better power efficiency [25]. An enhanced version Contiki X-MAC, is used to reduce power consumption and to retain good network conditions. Contiki's LPP protocol improves power consumption for broadcast data communication within a network.

4. *Radio Layer*: Last layer in the Contiki Netstack. The data arrived is controlled by interrupt handlers. The packetbuf process is polled for this.

Cooja [25][26]:

It is a network simulator explicitly designed for WSN. Cooja, is provided along with a Contiki which is an OS for the network of memory-constrained systems for less power wireless IoT devices. On top of the Contiki operating system, Java based simulator, cooja is used to run the sensor network. It is a very useful tool for Contiki related development, as it permits developers to test their code. Developers regularly do new simulations to validate the performance of their systems. In spite of poor documentation it is a best tool for IoT protocols simulation. It can be used to create network layouts, compile motes, and examine results with the help of Collect plugin. You can write various scripts to produce more filtered results. Cooja is selected for designing because of the following advantages;

1. It provides a shared library. The library is loaded in Java using JAVA Native Interfaces.
2. It works with the contiki operating system via various tasks such as handle an event, system memory fetching for analysis.

The Contiki website provides downloads for an Instant Contiki, Zip/tar file for Contiki installation. The majority of the support is available within Internet discussion groups.

5.2 OPEN WSN

The OpenWSN project [21][27][28] specifically for new IEEE802.15.4e, time synchronized channel hopping is used in the OpenWSN. Along with IoT standards such as 6LoWPAN, RPL and

CoAP, it permits mesh ultra-low-powered networks. The tools established include, visualization, debugging software, the Internet environment and a simulator to OpenWSN. It shows how, the IEEE802.15.4e can be applied on standard platforms, without dedicated hardware. The IEEE802.15.4e implementation [27] [28] is an autonomous from the OS presence on the mote. This can be easily portable to other OS. This enables an OpenWSN network to connect effortlessly to the IPv6 network. On top of 6LoWPAN, RPL is directing routing topology. RPL uses two types of routing: collection routing and source routing. OpenWSN defines different metrics for routing algorithms. CoAP, a protocol that empowers RESTful interaction within motes is supported by OpenWSN. A mote behaves as a web browser and server. The following protocol stack is adopted as the standard communication technology, and the Open WSN project is providing its implementations [21] [27] :

Application Level	CoAP, HTTP
Transport Level	UDP, TCP
IP/routing Level	RPL
Adaptation Level	6LoWPAN
Medium access Level	IEEE802.15.4e
Physical Level	IEEE802.15.4-2006

Fig 5.2.2 Open WSN Protocol Stack [27]

OPENSIM [27] :

It simulates an OpenWSN network without physical devices[27]. The simulated network behaves exactly same as a real network. OpenVisualizer, is used to join sensor nodes from the Internet along with Open Sim. OpenSim compiles mote firmware and create an instance of the resulting class for each emulated mote. At the time of simulation these emulate motes connects with the OpenVisualizer via eventBus[27].

The OpenVisualizer does not know whether it is emulated or real motes.

6. RESULT ANALYSIS ON BOTH PLATFORMS

Cooja([26]:

In cooja, a simulated contiki mote is simply a copy of compiled and executed contiki system. The system is handled and evaluated by cooja. Cooja can be loaded using command “sudo ant run-bigmem”. A new simulation can be generated in Cooja window. Cooja window is occupied by the main simulating tools. The functionality of each tool is given below:

- Network-It defines the location of each node in the network. It visualizes the status of node, including LED's, mote IDs, addresses, log outputs etc. After some time this window will occupy sensors information.
- Simulation Control- This panel is used to handle the simulation parameters such as: Pause, Start, load execution time and simulation speed. It means that the events can run faster than actual execution [2].
- Notes-This is a simple notepad for making notes of the simulation.

- Mote output-Output of all the nodes is in one window. If required, separate window for each node in the simulation is also possible.
- Timeline-Here messages and events such as channel variation, log output, LEDs changes are displayed.

Mainly two protocols are analyzed in this paper using the Contiki operating system and Cooja simulator: RPL and CoAP[26].The reason behind selecting these two is, as simple as researchers mostly use them to develop a real time application.The researcher needs basic analysis details, so that they can extend their work for further application development.

Simulation is event based therefore it is more realistic to analyze any scenario on it. A sample network is designed in the Cooja simulator for 10 sender nodes and 1 sink node as a root. The network state is shown in Fig 6.1.

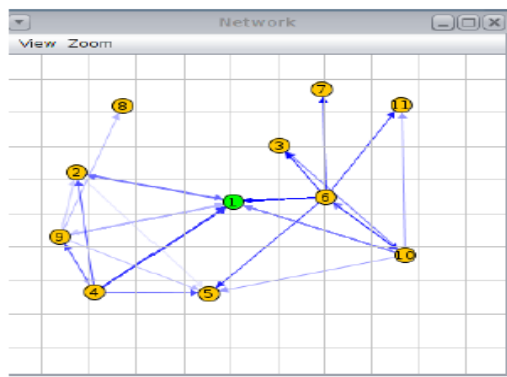


Fig. 6.1 Network scenario under Cooja simulators

The one sample application of sink- sender using UDP is used for analysis. Cooja plugin is used to create an output log file, which will be useful for analysis. In order to introduce lossyness, in wireless medium the Cooja Unit Disk Graph Medium (UDGM) model [26] is used.It presents lossy-ness with respect to radio medium relative distances of nodes.The environment parameters for the Simulation are mentioned in Table 6.1

Table 6.1 Environment Parameters

Parameters	Value
Radio Medium	Unit disk graph
Startup delay of mote	1000 ms
Simulation period	Half an Hour/ One Hour
No of nodes	10
RPL MOP	No DOWNWARD ROUTES
OF	MRHOF
DIO Min	12
DIO Doublings	8
Send Interval	4
RX Ratio	30-100%
TX Ratio	100%
TX and Interference Distance Range	50 m

Normally, the packet begins to transmit after small start delay. No Downward mode of RPL operation is used for analysis. Multipoint to point traffic is used for the evaluation. Contiki RPL default values are set for DIO Min and DIO Doublings[26]. The reception ratio (R_x) [26] defines lossyness of the radio medium. Results obtained in terms of different parameters, for various number of nodes are as shown below in Table 6.2.

Table 6.2 Analysis of RPLfor various attributes

No. of Nodes	Simulation time(min)	Average latency (ms)		Packets delivery ratio(%)	Average power consumption
		Sink	Sender		
10	30	20810	26100	82	0.24525
10	60	10016	15187	83	0.3565
50	30	30810	23400	81	0.3845
50	60	40828	34084	80	0.35675
100	30	410810	39400	78	0.7745
100	60	41449	54979	81	0.4695

CoAP:

The environment is same as mentioned in Table 6.3 for CoAP analysis. The sample application of client-server using UDP is used for analysis. A sample network in the Cooja simulator is having 10 client nodes and single sink/server node. The simulation time is 10 minutes. The packet delivery ratio is captured as the average of all nodes. Following Table 6.3 mention the PDR and latency delay. The Result shows that the with simple 10 CoAP client node PDR is 92.30%. In addition, the latency delay for simple CoAP in 205.27millisecond.

Table 6.3 PDR and Latency Delay for 10 nodes

For 10 Nodes using CoAP protocol	
Packet delivery ratio in %	92.30
Latency Delay in msec	205.27

OPEN WSN [27]:

The Open Visualizer along with Open Sim consists of core modules and different types of user interfaces as follows[27]:Graphical user interface (GUI),command-line interface (CLI) and web interface.Python Language is used for interaction.It includes python modules for operation such as :open Visualizer Gui, moteProbe etc.To get started with the visualizer, point your browser to the URL: <http://localhost:8080>. Simulation is event based so it is more realistic to analyze any scenario.RPL and CoAP protocols are analyzed with one very simple application.A small network is considered in the OpenWSN simulator,which includes 10 sender nodes and 1 sink node as a root. The network is shown in Fig 6.3.



Fig 6.3 Network Scenario in OpenWSN

A simulation of 10 min is done and the results are taken at the sink node side.The packet delivery ratio and latency delay are calculated and using Wireshark output.The values are as exhibited in Table 6.4

Table 6.4 Analysis of CoAP for 10 nodes for 10 min

For 10 Nodes using CoAP protocol	
Packet delivery ratio in %	80.92
Latency Delay in msec	221

7. CONCLUSION

The emergent idea of the IoT is improving our life style with smart devices, technologies, and smart applications. The paper has briefly described the need and use of two different platforms for IoT protocols. The primary analysis of two protocols RPL and CoAP using parameters PDR and Latency delay is done. The result indicates that both platforms are suitable for experimentation. The comparative study emphasizes that cooja platform is proper, as cooja log file is maintaining

all log records. These log records are very useful for analysis if we extend our simulation for various simulation parameters. This study can be further extended to solve real time problems.

REFERENCES

- [1] Balwinder Kaur Dr. Vijay Dhir, (2017) “Internet of things: Vision, Challenges and Future Scope”, In: International Journal of Advanced Research in Computer Science, vol. 8(4).
- [2] M. Chernyshev, Z. Baig, O. Bello, S. Zeadally, (2018) “Internet of Things (IoT): Research, Simulators, and Testbeds”, In: IEEE Internet of Things Journal, Vol: 5, Issue: 3, pp 1637 – 1647.
- [3] Julien Minaud, Oleksiy Mazhelis, Xiang Su, Sasu Tarkoma, (2016) “A gap analysis of Internet-of-Things platforms”, In: Elsevier Journal, Vol. 89–90(1), pp 5-16. [4] M. Chernyshev, Z. Baig, O. Bello, S. Zeadally, (2018) “Internet of Things (IoT): Research, Simulators, and Testbeds”, In: IEEE Internet of Things Journal, Vol: 5, Issue: 3, pp 1637 – 1647.
- [4] Solapure S.S, Kenchannavar Harish, (2016) “Internet of Things: A survey related to various recent Architectures and Platforms available”, In IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI) , DOI: 10.1109/ICACCI.2016.7732395.
- [5] Ruslan Kirichek, Andrei Vladyko, Maxim Zakharov, Andrey Koucheryavy, (2015) “Model Networks for Internet of Things and SDN”, In: IPSN '15, USA.
- [6] Gonçalo Marques, Nuno Garcia, Nuno Pombo, (2017) “A Survey on IoT: Architectures, Elements, Applications, QoS, Platforms and Security Concepts”, In: Advances in Mobile Cloud Computing and Big Data in the 5G Era. Studies in Big Data, Vol. 22, Springer, Cham.
- [7] Shikhar Verma, Yuichi Kawamoto, Zubair Md. Fadlullah, Hiroki Nishiyama, and Nei Kato, (2017) “A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues”, In: IEEE COMMUNICATIONS SURVEYS & TUTORIALS, Vol. 19(3).
- [8] Yahya Al Sawafi, Abderezak, Touzene Khaled Day, Nasser Alzeidi, (2018) “Toward hybrid RPL based IoT sensing for smart city”, In: International Conference on Information Networking (ICOIN), pp 599 – 604.
- [9] Sofia Biju, N. M. Shekhar, (2017) “Optimize the energy efficiency of RPL based 6LOWPAN by FL clustering”, In International Conference on Published In Big Data, IoT and Data Science (IEEE), pp 159 – 166.
- [10] Seyed Reza, Taghizadeh Hossein Bobarshad, Halima Elbiaze, (2018) “CLRPL: Context-Aware and Load Balancing RPL for IoT Networks under Heavy and Highly Dynamic load”, IEEE Access Journal, Vol: 6, pp 23277 – 23291.
- [11] Seelam Srinivasa Rao, N. Chenna Keshava Reddy, (2017) “An energy efficient improved RPL routing protocol”, In: International Conference on Intelligent Computing and Control (I2C2-IEEE), pp 1 – 7. Article history: 10.1109/I2C2.2017.8321954.
- [12] Ibtissem Zaatouri, Nouha Alyaoui, Awatef Benfradj Guiloufi, Abdennaceur Kachouri, (2017) “Performance evaluation of RPL objective functions for multi-sink”, Published in 18th International Conference.: Sciences and Techniques of Automatic Control and Computer Engineering (STA-IEEE), pp 661 – 665. Article history:10.1109/STA.2017.8314933.
- [13] Jinpeng Wang, (2017) “On the QoS routing with RPL”, In: International Conference Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), pp 1 – 5.

- [14] Seung-Chul Son, Nak-Woo Kim, Byung-Tak Lee, Chae Ho Cho, and Jo Woon Chong, (2016) “A Time Synchronization Technique for CoAP-based Home Automation Systems”, In: IEEE Transactions on Consumer Electronics , Vol: 62(1), pp 10 – 16.
- [15] Paridhika Kayal, Harry Perros, (2017) “A Comparison of IoT application layer protocols through a smart parking implementation”, In 20th Conference on Innovations in Clouds, Internet and Networks (ICIN-IEEE), pp 331 – 336.
- [16] Ilhan Aydin, Mehmet Karakose, Ebru Karakose, (2017) “A navigation and reservation based smart parking platform using genetic optimization for smart cities”, In: 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG-IEEE), pp 120 – 124.
- [17] Zhengguo Sheng, Hao Wang, Changchuan Yin, Xiping Hu, Shusen Yang, Victor C. M. Leung, (2015) “Lightweight Management of Resource-Constrained Sensor Devices in Internet of Things”, In: IEEE Internet of Things Journal, Vol: 2(5), pp 402 – 411.
- [18] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Claudio Vairo, (2016) “Car parking occupancy detection using smart camera networks and Deep Learning”, In: IEEE Symposium on Computers and Communication (ISCC), pp 1212 – 1217.
- [19] Nastaran Reza Nazar Zadeh, Jennifer C. Dela Cruz, (2016) “Smart urban parking detection system”, In 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE-IEEE), pp 370 – 373.
- [20] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, Moussa Ayyash, (2015) “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”, In: IEEE communication surveys & tutorials, Vol: 17(4), pp 2347 – 2376.
- [21] Thomas Watteyne, Xavier Vilajosana, Branko Kerkez, Fabien Chraim, Kevin Weekly, QinWang, Steven Glaser and Kris Pister, (2012) “OpenWSN: a standards-based low-power wireless development environment, Trans Emerging Telecommunications Technologies”, Vol:23(5), pp 480-493.
- [22] Industrial Low-Power Network, (2018) <https://tools.ietf.org/html/rfc5673>(accessed on 8.21.2018).
- [23] RPL, (2018) <https://tools.ietf.org/html/rfc6550>(accessed on 9.21.2018).
- [24] CoAP, (2018) <https://tools.ietf.org/html/rfc7252>(accessed on 7.1.2018).
- [25] ContikiOS1, (2018) <http://www.contiki-os.org/> (accessed on 8.21.2018).
- [26] Contiki OS2, (2018) http://anrg.usc.edu/contiki/index.php/Main_Page(accessed on 9.21.2018).
- [27] OpenWSN Project, (2018) <http://openwsn.org/>(accessed on 9.21.2018).
- [28] OpenMote Tech, (2018) <http://www.openmote.com/>(accessed 9.21.2018).
- [29] Dali Ismail, Mahbubur Rahman, Abusayeed Saifullah,(2018) “Low-Power Wide-Area Networks: Opportunities, Challenges,and Directions”in Workshops ICDCN '18, January 4–7, 2018, Varanasi, India , ISBN: 978-1-4503-6397-6.
- [30] <https://tools.ietf.org/html/draft-ietf-lpwan-overview-10>(accessed on 4.01.2019).
- [31] Aiju Thomas, N V Eldhose(2019)” Scalability concerns of chirp spread spectrum for PLWANapplications” in International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.10, No.1, February 2019 ,DOI : 10.5121/ijasuc.2019.10101.

AUTHOR :

Mrs.Sharwari S Solapure received her Masters of Technology in Computer Science and Engineering from CSE Department of Walchand College of Engineering, Sangli,Maharashtra, India.She is pursuing her research in the routing protocol of IoT at Research Center, CSE Department of GIT College of engineering,Belgavi,Karnataka, India. She is having 10 years of industrial experience and 13 years as faculty in CSE Department of Walchand College of engineering, Sangli,Maharashtra,India.Her field of interest includes Computer networks, IoT,wireless communication,Sensor network,SDN etc.

Dr.Harish Kenchanavar received his Ph.D. degree in CSE.As a research guide for VTU University, his research interests include computer network,wireless sensor networks, IoT etc.He is currently working as a Professor and HoD of IS Department in GIT College of engineering, Belgavi,Karnataka,India and guiding six research projects currently is having 15 years of experience. He has published a number of journal papers in reputed journals.He guided many UG,PG projects in the same area.