Gyancity Journal of Electronics and Computer Science,
Vol.4, No.1, pp. 14-24, March 2019
ISSN: 2446-2918 DOI: 10.21058/gjecs.2019.41003

Towards the implementation of Internet of Things

Pardeep Kumar* and Saleemullah Jamali
Quaid-e-Awam University of Engineering, Science & Technology Nawabshah, Pakistan
**Email: pardeep.kumar@quest.edu.pk*

Abstract

The Internet of Things (IoT) is a fast-expanding innovation that aims making several things / objects communicate with each other in a large heterogeneous environment. Lately, several ideas and schemes have been proposed by researchers in order to move further towards the realization of such a complex and challenging network. This paper discourses the overall requirements, challenges, merits, demerits and comparison of different operating systems, simulators, testbeds and architectures that have been proposed specifically for IoT. Additionally, a novel IoT architecture that intends to deal with standardization, interoperability, integration, security, etc., related issues of IoT has also been proposed in this paper.

Keywords—*IoT, simulation, operating systems, testbed, architecture, security, intelligence, management layer*

I. INTRODUCTION

The Internet of Things (IoT) intends to connect various devices and/or physical objects to the Internet in order to autonomously exchanging their data. These devices would sense the environment and collect data. The collected data can then be converted into a digital form to be shared by different devices / entities and accessed by end-users [1]. In this sense, IoT is an integral part of the future Internet that aims to have trillions of connected devices. This results in widely spreading out the application domain of IoT [2]. To name a few application areas [3] [4], IoT supports the medical tourism by enabling third-world patients to be virtually connected with doctors and medical resources all over the world, which would be a dream for them otherwise. IoT would enable us to find out how much ice, milk or yoghurt is left in the refrigerator, weather coffee is ready in the coffee machine, how much light, humidity and air is required in our room, when to alarm gardener in case a plant needs to be watered, when to send an auto SMS to the fire brigade if a building has caught fire, how does my car talk and get information about the traffic, accidents and weather conditions from the cars moving along on the road, what is the average number of cars per day crossing the bridge, when and in what quantity an item in the shop to be ordered/shipped, and many more. The various working groups are created by the IETF for the research and development of the standards for WSN i.e 6LoWPAN (IPv6 over Low-power Wireless Personal Area Networks), ROLL (Routing Over Low power and Lossy networks) and 6LO (IPv6 over Networks of Resource-constrained Nodes). As IoT faces numerous challenges and issues with the speedy advancement in technology, researchers are taking more interest towards the solutions of these challenges and issues [5]. Having a well-defined IoT architecture/standard that could deal with the complexity, scalability, energy, security, privacy, cost, co-existence, integration among the things/technologies and the socio-economic issues would be the genuine step to substantiating the IoT. The first step towards the realization of the IoT is related to its tangible implementation by

ISSN: 2446-2918 DOI: 10.21058/gjecs.2019.41003

considering some real-life networks. For that, a fair amount of research is being carried out to design and develop simulations tools, architectures, operating systems and testbeds for IoT. Since IoT is still at an infancy stage, its most of the proposed schemes need to be materialized. This paper critically analyses the working, merits, demerits and comparison of the several operating systems, simulators, testbeds and architectures that have been proposed for IoT and then proposes a novel IoT architecture to deal with the mentioned challenges of IoT. This paper is organized as follows. The next section highlights the challenges and requirements of IoT. Afterwards different operating systems, simulators and testbeds of IoT are briefly discussed in section 3 (due to the space limitation). Different IoT architectures including the proposed one are presented in section 4. Section 5 generally discusses the limitations of the conversed operating systems, simulators, testbeds and architectures, whereas the section 6 concludes the paper.

II. CHALLENGES AND REQUIREMENTS OF IOT

A highly scalable and heterogeneous network mostly containing small devices such as sensor nodes with limited resources would be the real shape of IoT. Hence, there are marginal chances that various operating systems, architectures, simulation tools and testbed that have been designed for the traditional communication networks can 'out-of-the-box' be used for IoT. The requirements for an IoT oriented operating system include the efficient handling of a microcontroller, limited memory, low computing, interaction with different integrated devices, security, light weight communication protocols, various hardware platforms, various programming languages, multi-threading, real timeliness and adaptation of 6LoWPAN, etc. The commonly used desktop operating systems such as windows, Linux and Unix are not suitable for IoT devices due to the high utilization of resources. Thus, novel operating systems are being designed and developed for IoT. Similarly, the existing network simulators such as NS-2/3, OMNeT++, OPNET, etc. are being transformed to support the basic requirements of IoT to facilitate, for example, time based or consecutive events, sensing and radio propagation models, protocols for different layers, diverse network topologies, communication modules, energy consumption models, mobility support, inter-operability, heterogeneity, scalability, etc. Moreover, in order to validate the simulation results, real implementation of IoT is necessary. For this purpose, different researchers and academicians are engaged deploying IoT testbeds. Most of the requirements of IoT simulators are also applicable for the IoT testbed along with the factors such as number of devices, high level architecture, users participation, applications support, resource discovery and registration, fault management, monitoring and analysis, resource reservation and scheduling, testbed reconfiguration, security etc.

III. IMPLEMENTING IOT

In this section we discuss different operating systems, simulators and testbeds that have been designing, developed and deployed for IoT.

A. Operating Systems for IoT

To support the general requirements of IoT, several operating systems have been designed and developed, few of them are discussed below. Table I compares different features of these operating systems.

1) TinyOS: TinyOS[1] is specially designed for microcontroller based devices that consume less energy. This embedded OS is developed by TinyOS alliance under the BSD license and is written in the nesC language. There are different active working groups that perform research activities to add new features, services, interfaces, etc to this OS [6]. TinyOS was initially developed for educational purposes but currently it is also being used in industries. This event driven operating system minimizes the resources usage such as energy by extending the sleep mode duration. The other characteristic of the TinyOS include software components, non-blocking, stack and threading, support for heterogeneous platforms, etc [7].

2) Google Brillo: Google has recently developed an android based OS for the IoT called as Google Brillo[2]. This OS is specially designed for the small and smart devices that consume less energy. Brillo uses the cross platform known as Weave for the integration purpose, where different devices can communicate with each other.

3) RIOT: Real-time IoT (RIOT)[3] is an embedded operating system specially designed to fulfill the IoT needs such as low memory, low energy and light weight communication. It also supports multi-threading, C/C++, modularity and real-timeliness [8]. RIOT is an open source OS that supports various platforms, architectures, boards, drivers, virtualization and testbeds.

4) Contiki: The C language based Contiki OS[4] is developed by Adam Dunkles and his team at Swedish Computer Science Institute. This open source OS runs on various platforms such as Embedded Sensor Board (ESB) and Modular Sensor Board (MSB), etc. It supports multi-tasking, multi-threading, and proto-threading [9]. Contiki supports less energy and less memory, i.e., RAM and ROM of 2KB and 40KB respectively. Contiki offers communication in IPv4 and IPv6 as well as uIP and Rime stacks.

TABLE I: Comparison of different OSs for IoT

Characteristic	**TinyOS**	**Contiki**	**RIOT**
Min RAM	1.5KB	10KB	≤ 1 KB
Min ROM	1.5KB	30KB	≤4 KB
Supporting Language(s)	C / C++	Standard C	nesC
Multi-Threading	Yes	Yes	Limited support
Real-Time	Yes	Yes	Yes
Memory allocation	Static / Dynamic	Static / Dynamic	-
Toolchains	gcc, gdb, valgrind	gcc, gdb	MSP430-gcc
License	GNU LPGL	3-clause BSD	BSD

B. Simulators for IoT

There are various simulators that support the implementation of IoT [10], a few of them are briefed below. Table II compares these simulators against several characteristics / parameters.

1 http://www.tinyos.net/
2 https://developers.google.com/brillo/
3 http://www.riot-os.org/
4 http://www.contiki-os.org/

Gyancity Journal of Electronics and Computer Science,
Vol.4, No.1, pp. 14-24, March 2019
ISSN: 2446-2918 DOI: 10.21058/gjecs.2019.41003

1) COOJA: Contiki OS Java (COOJA)[5] is a Contiki based network simulator and emulator specially designed for IoT [11]. In COOJA, devices are identified by its type such as COOJA emulated and Java based devices. Developers can test their small and large-scale sensor networks before its implementation on fleshly devices. Same code can be used for simulation as well as real deployment. Different Contiki libraries can be compiled and loaded in the same COOJA simulation, representing heterogeneous networks. COOJA uses some functions and plugins to control, analyze and monitor a Contiki system. 6lbr provides the routing, smart and transparent bridging facility consist of Contiki OS. It uses the IPv6 and 6LoWPAN for connecting the devices to the internet.

2) Castalia: Castalia[6] is an OMNeT++ based simulator that supports networks of low-powered wireless sensing devices and wearable devices. Castalia helps developers for testing and evaluation of the distributed algorithms, protocols with real wireless medium and radio models [12]. Castalia also supports various types of models such as different MAC protocols, radio models, mobility, climate monitoring, etc. for the IoT.

TABLE II: Comparison of different simulators for IoT

Characteristic	COOJA	NS3	Castalia	SImpleIoTSimulator
Level of detail	Cross level	Generic	Generic	Generic
Timing	Discrete Event	Discrete Event	Discrete Event	Discrete Event
Simulator Platform	Contiki, Unix, Windows	Linux, SunOS, Windows	Unix, Windows	Linux
Heterogeneous	Yes	Yes	Yes	Yes
Interoperability	Yes	Yes	—	—
GUI Support	Yes	Yes	Yes	Yes
Software license	Yes	GNU, GPU	Academic Public	Simple soft
Energy model	Yes	Yes	Yes	Yes

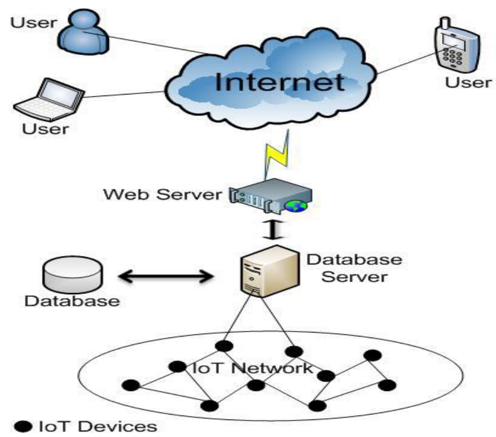
3) NS3: Network Simulator-3 (NS3)[7] is developed in C++, but optionally it also supports the Python language. This is an open source and discrete-event simulator particularly developed for the research and academic purposes. Researchers can use this simulator for testing the behavior of their designed IoT based networks before deployment. Various modules such as CSMA, LET, NS3WiFi and API's are available in NS3 [13]. The other features of NS3 include logging, tracing, real time scheduling, helper API, etc.

4) SimpleIoTSimulator: The SimpleIoTSimulator[8] is purely designed for IoT and Machine to Machine (M2M) communication. It is an easy to use platform that quickly creates networks of thousands of sensors and gateways. Currently this simulator can work with application layer protocols such as Constrained Application Protocol (CoAP), Message Queue Telemetry Transport (MQTT) and Hyper Text Transfer Protocol/ Secure (HTTP/s), etc. In some cases, network management simulator is available for the support of Simple Network Management Protocol (SNMP), Secure Shell and Telnet, etc.

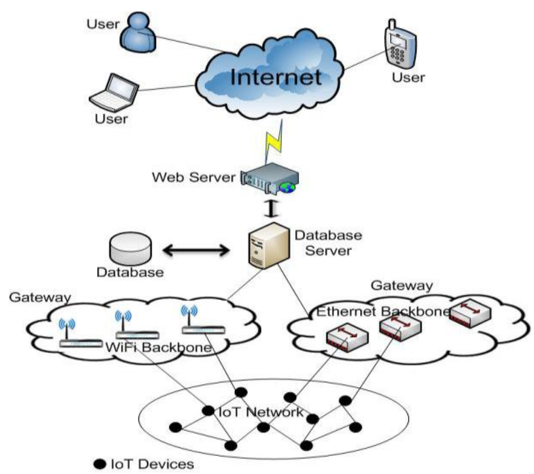
C. Testbeds for IoT

5 http://anrg.usc.edu/contiki/index.php/Cooja Simulator
6 https://castalia.forge.nicta.com.au/index.php/en/
7 https://www.nsnam.org/
8 http://www.smplsft.com/SimpleIoTSimulator.html

Testbeds for IoT can be either two-tiered or three-tiered as shown in Figure 1. The two-tiered architecture is suitable for small testbeds where only the limited numbers of devices are used [14]. In this architecture, IoT nodes can connect and communicate with server directly but in three-tiered architecture, nodes need gateway facility for interaction and communication with the server. This architecture is suitable for large scale testbeds, where huge numbers of devices are used. Some of the IoT testbeds deployed around the world are briefly stated below. Table III shows the characteristics and comparison of these IoT testbeds.



(a) 2-Tiered Testbed



(b) 3-Tiered Testbed

Fig. 1: Testbeds for IoT

1) SmartSantander: The purpose of Smartsantander project[9] is to integrate various IoT technologies, applications and perform experimental based testing for the smart city. This scalable project targets at the deployment of 20,000 sensors in Belgrade, Guildford, Luebeck and Santander while supporting a diverse range of IoT technologies [15]. The main features of SmartSantander are to validate and evaluate

9 http://www.smartsantander.eu/

the IoT architecture including IoT interaction & management protocols and services such as discovery, location, traveling, traffic, identity management, security and social acceptance for a smart city.

2) JOSE Testbed: Japan-wide Orchestrated Smart / Sensor Environment (JOSE)[10] is a testbed to perform experiments of WSN based IoT networks. A huge number of sensors are deployed on large area to collect the environmental data. The collected data can be processed and analyzed in real time using high-speed network.

TABLE III: Comparison of different testbeds for IoT

Characteristic	SmartSantander	FIT	JOSE	DES	Infinite
Heterogeneity	Yes	Yes	Yes	Yes	No
Scalability	Yes	Yes	Yes	No	No
Federation	Yes	Yes	No	Yes	Yes
Mobility	Yes	Yes	No	Yes	Yes
Repeatability	Yes	Yes	—	Yes	—
Interface	REST/JSON	Web Based/ Rest API	Open flow controller	DES-Protal / DES Cript	—
Locations	4	6	1 (5 data centers)	1	3
Total Number of nodes	20,000	2738	12,5000 VMs	120	N/A
Nodes types	Different sensors	WSN430, M3, A8	—	MSB-A2	—

3) DES Testbed: The Distributed Embedded Systems (DES)[11] testbed is deployed to perform experiments & research work on tiny wireless sensing and IoT devices on multihop networks [16]. The DES testbed is based on various components including nodes, configuration and management tools, auto evaluation and network visualization. This hybrid testbed plays an important role for the several EU projects such as WiMeshLab, OPNEX, G-Mesh-Lab and WISEBED.

4) INFINITE Testbed: The INternational Future INdustrial Internet Testbed (INFINITE)[12] testbed is developed for the industrial purposes. It is located at Ireland and is developed by the EMC Corporation with the approval of Industrial Internet Consortium. Other partners of the INFINITE testbed include Vodafone, Irish Government Networks, Cork Internet Exchange and Asavie. This IoT testbed connects the industries products to the Internet. This testbed is used in various industries such as manufacturing, healthcare, public sector, transportation, energy sector, etc.

5) FIT Testbed: The Future Internet of Things (FIT)[13] IoT Lab is particularly designed for the educational and industrial purposes and is deployed at six various sites in France [17]. It provides the platform to deploy a huge number of tiny wireless sensor nodes. The hardware of this testbed includes the nodes of WSN430, M3 and A8. It can support Contiki, RIOT, FreeRTOS, OpenWSN and TinyOS with the facility of online-based or CLI based utilities.

10 http://www.nict.go.jp/en/nrh/nwgn/jose.html

11 http://des-testbed.net

12 http://www.iotinfinite.org/

13 https://www.iot-lab.info/

D. Tools and services

Ubidots deals with clouding facility for IoT devices/sensors. Users can develop a program, deploy it and connect the selected devices to the Internet. The results of implemented applications can be viewed as in visual form, users can receive emails or SMS when any event triggers and cross the threshold value. The traffic of the IoT networks can be captured, monitored, analyzed and traced by the Wireshark. IoT networks also can be analyzed and diagnosed by Foren6 tool. Foren6 can capture the 6LoWPAN/IPv6 traffic, analysis routing misbehavior, debugging and live capturing of networks. The services for the IPv6 are provided by different service provider i.e Hurricane Electric and the services of Tunnel Setup Protocol (TSP) for the tunneling protocol provided by the gogo6. Copper (Cu) is a management tool for the smart IoT devices that can be integrated with the different web browsers like Chrome, Mozilla etc. It can use CoAP (Constrained Application Protocol) for interaction and web communication [21].

IV. IOT ARCHITECTURES

Currently, there is no any globally accepted architecture of the IoT, numerous IoT architectures have been proposed though. Researchers in [18] have proposed four layers and three sub-layers based IoT architecture. The layers include application, middle coordination and backbone layer whereas the sub-layers include access layer, edge technology layer and existed alone application system. This architecture generally considers the integration of various technologies; however, it does not look into security, QoS, scalability, etc. issues. A five layered IoT architecture has also been proposed in [19], however security is provided only at the network layer. Most of the other proposed IoT architectures do not included the power management and intelligence functions. Energy is crucial for the life time of IoT based networks and intelligence is also essential for the auto configuration or self-organization characteristics of the network. We propose an enhanced IoT architecture that intends to deal with above mentioned issues of such a huge and diverse network.

A. A Proposed IoT Architecture

To address the diverse challenges of IoT, a five-layered architecture of IoT is proposed as shown in Figure 2. The working of each layer is briefly stated below whereas the detailed working of this architecture is presented in [20].

1) Device layer: This layer identifies the things or objects of environment, obtains the data and collects information by sensing environmental properties. The information could be collected from RFID, sensors, 2D-Barcode or physical objects.

2) Communication layer: Communication layer offers transmission and connectivity functions. This layer works with different technologies and protocols such as Bluetooth, WiFi, Zigbee, sensor networks, 6LoWPAN, 3G, UMTS for communication, interoperability and interfacing for dissimilar type of devices and objects.

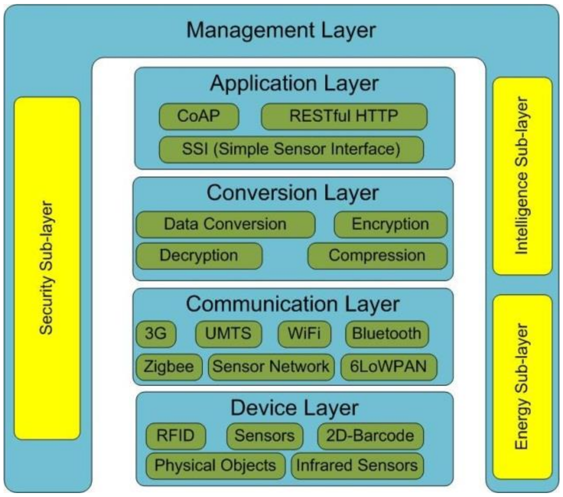


Fig. 2: The proposed IoT Architecture

3) Conversion layer: Conversion layer offers data conversion, compression, encryption and decryption services. This layer provides data transformation ability in an acceptable format for communication with application layer components.

4) Application layer: Application layer provides services to the applications, devices and end-users as per their needs by utilizing different protocols and interfaces. CoAP is considered a suitable protocol for IoT and provides integration between nodes and networks with a simple translation to HTTP. Simple Sensor Interface (SSI) is also application layer protocol with minimal overhead. It is also able to work with network layer protocols such as nanoIP. Computers and terminals use this protocol to exchange information with smart sensors.

5) Management layer: For the proposed IoT architecture, management layer controls the whole supervision of the IoT system, devices and their configuration to build a suitable relation. Management of big data is also the responsibility of this layer as it stores, evaluates and processes the data. Mobility management of IoT resources is also offered by this layer. Additionally, this layer is divided into three sub-layers namely security, intelligence and energy layers. The security sub-layer manages security concerns for the whole IoT system at each layer. The energy sub-layer attempts to minimize energy consumption during data collection and communication between devices and objects. On the basis of collected data, intelligence sub-layer takes smart decisions at application layer, especially for the auto-configuration purpose. An interface is used at intelligence sub-layer for connections with database, where applications take decision based on the received data.

B. Functional Design of the Proposed IoT Architecture

Figure 3 depicts the working flow of the proposed IoT architecture. After the deployment of an IoT network containing different devices such as sensors, they start collecting the data in a periodic or aperiodic fashion. The collected data is then sent to the gateway. The gateway first converts the data in an appropriate form and then forwards it to the storage server via Internet. So, all the collected data of IoT devices is stored on the storage server. The desired application monitors and analyses the data and also takes the smart decisions based on the user/application requirements. Alternatively, the application sends instructions or commands to devices through Internet. Devices receive those instructions via the gateway and then apply those instructions in order to behave smartly. Here we suppose that the IoT network contains N sensing devices, where n_i shows each of the sensing devices such that $i = \{1, 2, 3, .., N\}$. The D shows the collected data of the IoT network and d_i shows the collected data of the particular IoT device as shown in Equation 1.

$$D = \sum_{i-1}^{N} d_i \quad (1)$$

$$C_d = D_c\,(d_1 + d_2 + d_3 + \cdots + d_N) \quad (2)$$

$$C_d = D_c(D) \quad (3)$$

In Equation 2, D_C shows the data conversion function, whereas C_d shows the acceptable format of data in Equation 3. These functions help devices to take smart and intelligent decisions based on the application requirements.

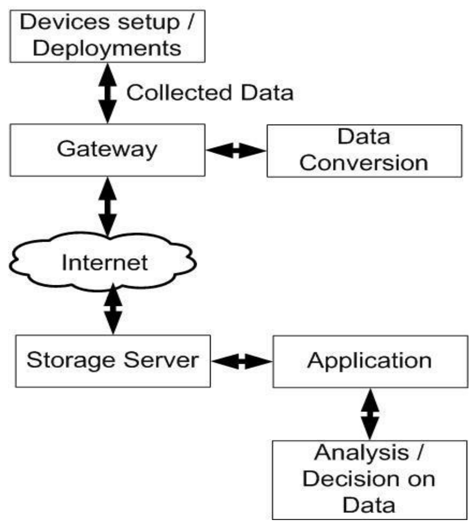


Fig. 3: Functional block diagram of IoT Architecture

V. DISCUSSION

Lately, the research on small sensing devices based smart networks has grown very rapidly. These networks spy our environment and autonomously monitor, collect, share and process the data collected with the help of different sensing devices. IoT envisions connecting physical and virtual objects around the world to the existing Internet for this purpose. Several operating systems, simulators, testbeds and architectures have been designed and deployed for IoT. Since these are still early days of IoT, most of the above discussed operating systems have limitations. Contiki and Tiny OS can only support some specific programming languages and provide limited features of multi-threading, real-timeliness, modularity and MAC & Radio modules. Their APIs also need to be friendlier. Moreover, these OSs use FIFO strategy, so long processes consume more time for execution. RIOT OS can fulfill most of the IoT requirements but due the limited ROM size it cannot support high level languages such as Java. IoT architectures and protocols are usually pre-tested through simulators. The IoT supporting simulators that have been discussed in this paper are generic type except COOJA, which has specifically been designed for IoT.

However, COOJA's Java Native Interfaces (JNI) depends on the outer resources such as compilers and linkers. OMNet++ supports the limited number of protocols and is having the issues such as compatibility with different models and debugging. NS3 is a popular network simulator but unfortunately it has various issues and limitations such as credibility, validation, scalability, no support for IPv6, Cygwin's no support for python and simulating the performance of upper layers. As discussed, the proposed architectures for IoT do not completely fulfill the requirements of IoT and are facing issues such as integration, interoperability, security, intelligence, etc. Therefore, an enhanced IoT architecture has been proposed in this.

VI. CONCLUSION

This paper provides a survey of various operating systems, simulators, testbeds and architectures that support the implementation and experimentation of IoT technologies and networks. This paper intends to help researchers and academicians in selecting a suitable OS, simulator and/or testbed for implementation and evaluation of an application specific IoT network. Lastly, a novel layered-based IoT architecture has been proposed in this paper that aims to support a large number of heterogeneous devices with proper integration, interoperability, management, security, energy, intelligence, etc. related functionality. The future work includes the implementation and evaluation of the proposed IoT architecture.

REFERENCES

[1] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," International Journal of Ad Hoc Networks, vol. 10, pp. 1497–1516, 2012.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," in IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.

[3] R. Petrolo, N. Mitton, J. Soldatos, M. Hauswirth, and G. Schiele, "Inte- grating Wireless Sensor Networks within a City Cloud," in International IEEE SECON Workshop on Self-Organizing Wireless Access Networks for Smart City, Singapore, June 2014.

[4] M. Jevtic and N. Zogovic, "Evaluation of Wireless Sensor Network Sim- ulators," in 17th Telecommunications forum TELFOR, Serbia, Belgrade, November 24-26 2009.

[5] Zeinab K A Mohammed and Elmustafa S A Ahmed, "Internet of Things Applications, Challenges and related future technologies", World Scientific News, Vol. 67, no. 2, pp. 126-148,2017.

[6] T. Reusing, "Comparison of Operating Systems TinyOS and Contiki," in Seminar: Sensor Nodes Operation, Network and Application (SN), Summer Semester 2012, Munich, Germany, August 2012, pp. 7–14, volume NET-2012-08-2 of Network Architectures and Services (NET).

[7] T. Borgohain, U. Kumar, and S. Sanyal, "Survey of operating systems for the iot environment," International Journal of Advanced Networking and Applications, vol. 6, no. 5, pp. 2479–2483, 2015.

[8] E. Baccelli, O. Hahm, M. Wahlisch, M. Guenes, and T. Schmidt, "RIOT: One OS to Rule Them All in the IoT," [Research Report] RR-8176, 2012.

[9] T. Paul and G. S. Kumar, "Safe Contiki OS: Type and Memory Safety for Contiki OS," in International Conference on Advances in Recent Technologies in Communication and Computing, October 2009, pp. 169–171.

[10] S. Rampfl, "Network Simulation and its Limitations," in Seminars FI / IITM / ACN SS2013, Network Architectures and Services, August 2013, pp. 57–63.

[11] B. Sobhan babu, P. Lakshmi Padmaja, T. Ramanjaneyulu et.al, "Role of COOJA Simulator in IoT", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Vol. 6, no. 2, March – April, pp. 139-143, 2017.

[12] C. Tapparello, H. Ayatollahi, and W. Heinzelman, "Energy Harvesting Framework for Network Simulator 3 (ns-3)," in 2nd International Workshop on Energy Neutral Sensing Systems (ENSsys14), Memphis, TN, USA, November 06 2014, pp. 37–42.

[13] D. Pediaditakis, Y. Tselishchev, and A. Boulis, "Performance and Scala- bility Evaluation of the Castalia Wireless Sensor Network Simulator," in 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools10), Brussels, Belgium, 2010.

[14] A.-S. Tonneau, N. Mitton, and J. Vandaele, "A Survey on (mobile) wireless sensor network experimentation testbeds," in DCOSS - IEEE International Conference on Distributed Computing in Sensor Systems, Marina Del Rey, California, United States, May 2014.

[15] L. Sanchez, L. Munoz, and J. A. Galachel, "SmartSantander: IoT experimentation over a smart city testbed," International Journal of Computer Networks, vol. 61, pp. 217–238, March 2014.

[16] M. Guenes, O. Hahm, and K. Schleiser, "DES-Testbed A Wireless Multi- Hop Network Testbed for future mobile networks," 2010.

[17] O. Fambon, E. Fleury, G. Harter, R. Pissard-Gibollet, and F. Saint- Marcel, "FIT IoT-LAB tutorial: hands-on practice with a very large scale testbed tool for the Internet of Things," in 10mes journes francophones Mobilit et Ubiquit (UbiMob), Sophia-Antipolis, 5-6 June 2014.

[18] L. Tan and N. Wang, "Future Internet: The Internet of Things," in 3rd In- ternational Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, August 2010, pp. 376–380.

[19] T. Fan and Y. Chen, "A scheme of data management in the Internet of Things," in 2nd International Conference on Network Infrastructure and Digital Content, Beijing, China, September 2010, pp. 110–114.

[20] S. Jamali, P. Kumar, and U. Ali, "Internet of Things: Architecture and Integration with Other Networks," in 1st International Conference on Modern Communication and Computing Technologies, MCCT14, Nawabshah, Pakistan, 26-28 February 2014.

[21] Antonio Linan Colina, A Vives, M Zennaro, A Bagula, E Pietrosemoli, "Internet of Things in five days", IPv6 WSN Book v1.1, pp. 1-227, June 2016.