International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-8 Issue-6C, April 2019

Analysis of Various Light Weight Protocols in Internet of Things-A Comparative Study

Ankit Khare, Rashmi Sharma, Neelu Jyoti Ahuja

Abstract*: Internet of Things (IoT) incorporates the physical world with computing devices enabling them to be operate from remote region. Sensors and actuators with the assistance of communication protocols (MQTT, CoAP, HTTP, and REST) exchange the data, enabling the smart devices to interact with each other. These sort of devices having restricted limit, the protocols are intended to be handle low data transfer capacity, communication issues and high latency rate. In this paper, existing lightweight protocols in IoT are analyze based on eight different parameters(Architecture, Need of broker, Transport protocol, Security protocol, Scope, Design Methodology, Message size, Service levels) and these protocols can be used based on their application area.

Keywords: COAP, *IoT*, MQTT, Web Services, REST, SOAP

I. INTRODUCTION

Internet of Things (IoT) is a connection of smart objects like actuators, sensors, mobile phones, Radio-Frequency IDentification (RFID) tags etc.[1]. These mentioned objects are called as 'Things'. And the term 'Internet' is defined as a communication medium i.e. used to interact between two or more physical devices, without knowing their geographical location [28]. In another words, IoT is a network of smart objects that recognize and interact with each other by unique addressing schemes [2].

IoT has become integral part of human life in the modern era, especially in commercial and domestic fields [2][3]. Along with above said area e-health, e-learning, intelligent transportation, logistics and industrial manufacturing are also few IoT based applications areas [23][27].

In all above mentioned applications one object (thing) interacts with other. It is not necessary that all these objects have same platform/architecture/operating systems[29].

A. Architecture of IoT

As discussed in figure 1, IoT's architecture comprises of following three layers that are Application layer, Network layer and Perceptual layer [30]. Fundamentally, the below mentioned layers explain the basic structure or working of IoT

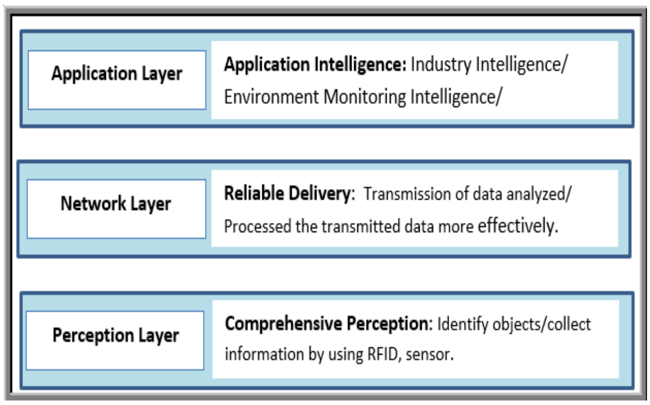


Figure. 1: Basic Architecture of IoT [30]

A. Perception Layer

It is responsible for sensing, reading and identification of objects, RFID tags, sensors, bar code readers GPS, actuators come under perception layer.

B. Network Layer

It is the brain of IoT. It is responsible to transmit the data after reading/collecting from perception layer. Along with, data transmission, the information processing is also the responsibility of this layer. Hence, Network layer is the junction point for communication network, network of internet and management & information center with intelligent processing center.

C. Application Layer

Application layer is a kind of bridge in-between IoT and industries. The creation of this layer is due to incorporation of social division of IoT and industry to understand the broad level of computations.

B. Communication in IoT

Communication in IoT takes place through either by using broker architecture or web services. Broker architecture based on the principles of Machine-to-Machine(M2M) communication by transmitting the binary data. To overcome the shortage of broker architecture, the web services are used in IoT to enhanced communication.

Revised Manuscript Received on December 22, 2018.
Ankit Khare, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India.
Rashmi Sharma, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India.
Neelu Jyoti Ahuja, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India.

A. Web Services

Web Services are combination of two words Web and Services, where services will be provided by web or World Wide Web (WWW)[34]. Web services are of two types as discussed below:

B. Representational State Transfer (REST)

REST is an architecture i.e. use to design networked applications. These web services based on stateless operation and/or client server architecture that always uses the HTTP protocol [18].

C. Arbitrary web services

In these services, arbitrary set of operations and SOAP messages are used.

C. Web Service Message Formats

Web service and client communicate through message exchange. Client sends the request message, which is responded by web service via response message. This working is similar to the working of HTTP where web server replies to the request of web browser with an HTTP response. Message format of REST web service, which came after SOAP, uses HTTP and XML [36]. This was followed by the REST that used JavaScript Object Notation(JSON) as message format.

A. SOAP

Simple Object Application Protocol (SOAP)It is an application of the XML specification. Figure 2 shows the SOAP architecture that provides an envelope based service over the Internet for sending Web Services messages [37]. It is a variant of JavaScript Object Notation (JSON) and REST

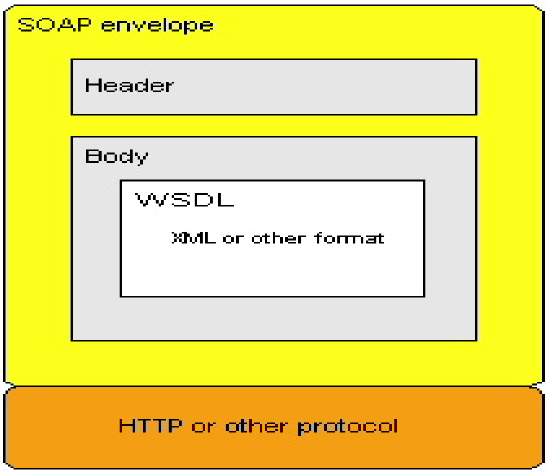


Figure 2: SOAP Architecture [37]

```
<? xml version="V.0"?>
<soap: Envelope
xmlns: soap="http://......soap-envelope"
soap: encodingStyle="http://.....soap-
encoding">

    <soap: Header>
    </soap: Header>

     <soap: Body>
        ... message data ...
        <soap: Fault>
        </soap: Fault>
    </soap: Body>
</soap: Envelope>
```

SOAP message consists of:

Envelope → Header, Body → Message Data, Fault

This structure of SOAP message is utilized to send the request and response among client and web services. SOAP is not used to determine the way a message is delivered to the web service from a client, in spite of the fact that the most widely recognized situation is by means of HTTP.

B. REST + XML

REST style work somewhat different in relation to SOAP web services. In REST request, no XML request is send. This request is similar to the request sent by web browser to web server in HTTP request. A REST response is normally a XML record sent in a standard HTTP response, similarly as though a program had asked for it. REST focuses more on resources as compared to services [39] [40]. Similar to an HTML page in a site, a resource is also identified with a given URL.A client profile in a social networking application can be an example of resource. E.g. http://abc.social.com/profiles/u_id3251.

The following syntax depicts how an XML document returned may look:

```
<profile>
    <Name> Ankit </Name>
    <Address>
        <City> Dehradun </City>
        <State> Uttrakhand </State>
        <pin code> 654321 </Pin code>
    </Address>
</profile>
```

REST normally supports accumulations of resources. For example, this URL may show a list of all open client profiles on public domain: http://social.abc.com/profiles/

C. RESTful Web Services

REST architecture is the base of RESTful Web Services where everything considers as a resource. Such services are maintainable, scalable and lightweight [35] [38].

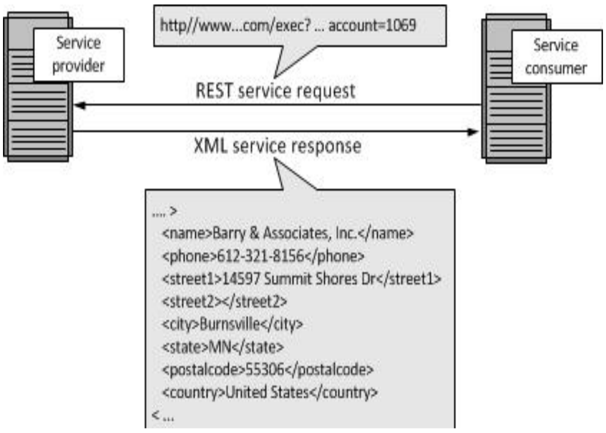
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Additionally, for web-based applications, creation of APIs comes under these web services. RESTful based application has following characteristics:

i. Functionality of the application are distributed into various resources

ii. Minimal and uniform set of commands are used to uniquely address the sender/receiver along with type of message. In general, commands of HTTP such as GET, POST, PUT, or DELETE are used.

iii. The protocol supports client/server, stateless, layered, and caching as well.



D. Lightweight Protocols in IoT

The Internet Engineering Task Force (IETF) defines and standardizes IP-based IoT[6]. Some of the working groups (WG) that participate in this task [8], are discussed here:

A. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)

Collaboration of IEEE 802.15.4 with IPv6 is the primary concern of WG because IPv6 addressing of IPv6 is contemplated for large number of hosts that may be connected in IoT [8].

B. Constrained Application Protocol

Constrained Application Protocol (CoAP) interactive model is similar to HTTP's client/server model. With the help of two layers, message and request/response layer, CoAP are used to understand the CoAP [5][7]:

i. Message layer is the lower layer that deals with UDP and asynchronous switching. CON (confirmable), NON (non-confirmable), ACK (Acknowledgement), RST (Reset) are four messages that plays a vital in message layer.

ii. Request/response is the upper layer that helps to communicate and deal with lower layer.

CoAP is responsible for reliability, duplication of messages and communication. Responsibility of CoAP is divided into two parts where reliability and duplication of messages is handled by message layer and communication part is handled by request/response layer [5][7].

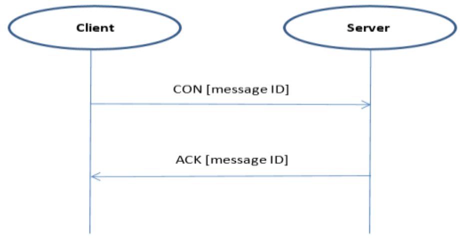


Figure 4 : Reliable Message Transport in CoAP[7]

C. Message Queue Telemetry Transport Protocol (MQTT)

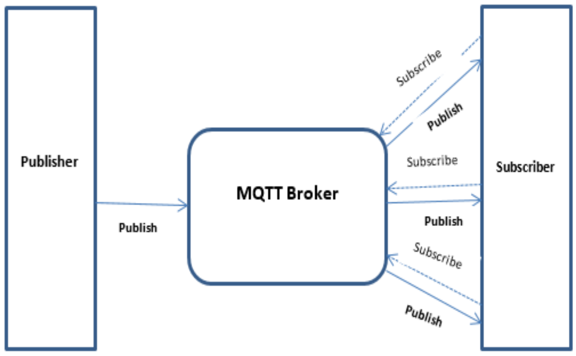


Figure 5: MQTT Architecture [21]

In figure 5, MQTT architecture has three components-publisher, subscriber and broker [21][22]. It is basically following a publish/subscribe architecture. If we think from IoT perspective, one component is responsible to connect with broker to transfer data and return to sleep state when promising, called publishers. One of the main important features of publisher is that they are lightweight sensors [21]. In order to be informed about the reception of new data, Subscribers needs to connect with broker.

II. LITERATURE SURVEY

There are four different kind of connections in IoT architecture for data transfer:

i. Device to Device(D2D)
ii. Device to Gateway(D2G)
iii. Gateway to Data System(G2D)
iv. Between Data Systems.

There are some protocols, which are used for D2D communication like CoAP, DDS. For device to gateway and reverse communication, we have two protocols named MQTT, RESTful.

A. Comparison of SOAP and RESTful Web Services

A Web service is a way to communicate between two applications or electronic devices over the internet. We have two categories of web services: Simple Object Access Protocol (SOAP) and Representational State Transfer (REST).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 1: Comparative Study on Web Services Protocol [30][34][35]

S.No	SOAP	REST
Full Form	Simple Object Access Protocol.	Representational State Transfer
Protocol	Protocol	Architectural Style
Data Format	XML data format only	Plain text, JSON, HTML, XML, etc.
Usability	Can't use REST	Can use SOAP web services.
Standards	Strictly followed.	Loosely followed
Bandwidth Requirement	More Resources and bandwidth	Less bandwidth and resource
Security	Describes its own security.	Inherits security measures from the underlying transport.
Preference	less preferred	More preferred

After comparison we can conclude that no protocol dominate one over other. Every protocol has its own Pros and cons. Although, one cons of SOAP protocol is that it is heavy weight protocol but one of its pros is provision of security for communication in tightly coupled environment [44] [45]. Whereas, REST is lightweight with some security issues. Therefore, considering security, we can say that SOAP is better than REST.

B. Comparison between CoAP and MQTT Protocol

We have some standard protocols that are used in IOT. MQTT and CoAP protocols are two most effective protocols for resource constrained devices [5][21]. The features of MQTT and CoAP protocols are:

i. Both of them are open standards.
ii. More suitable for constrained conditions in comparison to HTTP.
iii. They support asynchronous communication.
iv. Both protocols run on IP.

Along with some common features of CoAP and MQTT protocols, following are few basic differences between them as described in Table 2:-

Table 2: Comparison between CoAP and MQTT *[5][6][7][17][21][22]*

S.N	Parameter	CoAP	MQTT
1	Mode of Communication	Direct M2M communication	M2M communication with central broker
2	Functional communication protocol	One to one communication protocol	Many to many communication protocol
3	Architecture	REST architecture	Broker Architecture
4	Communication protocol	UDP	TCP
5	Model	Request/Response Model	Publish/Subscribe Model
6	Header size	4 Bytes	2 Bytes
7	QoS	2 level	3 level
8	Security	Datagram Transport Layer Security	Transport Layer Security/ Secure Socket Layer
9	Messaging	Both (Asynchronous & Synchronous Model)	Asynchronous Communication Model
10	No of Message Type	4	16
11	Type of Communication	Local Communication	Remote Communication
12	Speed of Transmit Cycle	Faster	Slower
13	Resource Discovery	Stable resource discovery mechanism	Flexible topic subscription

3) Comparison between various light weight protocol in IOT

Table 3: Comparison between various Lightweight protocols in IoT[5][6][7][21]

	MQTT	MQTT-SN	CoAP	LWM2M	REST	HTTP	SoAP
Full Form	Message Queue Telemetry Transport	**MQTT** for Sensor Networks	Constrained Application Protocol	LightweightM2M	Representational State Transfer	Hyper Text Transfer Protocol	Simple Object Access Protocol
Architecture	Publish/Subscribe	Request/Response	Request-Response, Publish-Subscribe	Request/Response	Request/Response	request/response	request/response
Need of broker	required, send devices communicate via broker	Required,send and receive messages.	No broker required.	Context broker	not required, end devices direct communicate	Hybrid broker can be used.	WebSphere Message Broker
Transport protocol	TCP/IP	UDP	UDP, TCP	UDP	TCP/IP	UDP	HTTP and SMTPetc
Security protocol	TLS	DTLS	IPSEC or DTLS	DTLS	HTTPS	HTTPS	It is already secure.
Scope	device to cloud cloud to cloud	client inside network, the broker is outside on Internet	Device to Gateway Gateway to Device	Device to device Device to gateway	device to cloud cloud to cloud	Device to device Device to gateway	Device to cloud Cloud to cloud.
Design Methodology	Protocol is data centric.	protocols suited for sensors network like ZigBee, Z-Wave	Generic web protocol for special requirements	Protocol is data centric.	Protocol is data centric	Protocol is document centric.	protocol is document centric
Message size	Small, binary with 2Byte header.	2 or 4 bytes Header, smallest 2 bytes, Largest 65535 bytes.	4 Bytes	XML document	Small , in different formats	Large, ASCII format.	XML document
Service levels	3	5	2	-	-	1	-
Implementation	to remotely perform service enablement and application management for “internet of things” embedded devices and connected appliances	implementation on low-cost, battery-operated devices	smart energy and building automation.”	enables **device management** and service enablement for **M2M**	**REST** can be **used** over nearly any protocol, it usually takes advantage of HTTP when **used for** Web **APIs**.	Used by the World Wide Web	protocol that allows Windows and Linux to communicate using Hypertext Transfer Protocol (**HTTP**) and its Extensible Markup Language (XML).

III. Discussion

There is no best protocol as such but their performance can be judge based on their following features:

A. Secure Protocols

While REST and HTTP use HTTPS as their security protocol, MQTT-SN, CoAP, and LWM2M use DTLS. However, CoAP uses IPSEC as well. Being slightly different from its derivate MQTT-SN, MQTT uses TLS. Above all, SoAP is already supposed to be a secure protocol [41]. So, in a way SoAP is better in terms of security as we do not have to extend another hand for security protocol.

B. Data Size

MQTT and MQTT-SN have the smallest data size as they have a 2 byte header file for both but, the size of the header file may vary for MQTT-SN to 4 bytes too. Likely, CoAP has 4 bytes header file. LWM2M and SoAP have XML documents as their data which are intended to be larger than that of others. Unparallel to the REST which has small message data size in different formats [42], HTTP has large message data size in ASCII format.

C. Application

MQTT-SN implemented on low cost battery devices, this focuses of more efficient power management when compared to its ancestor MQTT. Parallel to this CoAP works on smart management of energy and in the field of automation. LWM2M develops a connection between machine to machine [43]. Being lighter this protocols has better implementations on IoT and related services. However can be used over all other protocols as it is efficient because of the use of HTTP. Above all, SoAP being secure it establishes a communication gateway between operating systems.

IV. Conclusion

MQTT does not support the labelling of messages with their nature of type and it does not carry metadata to help the clients to understand the message. Before sending any MQTT messages, every client must be aware with its message formats, only after that communication can be started. Whereas CoAP allow devices to enquire each other to determine how to exchange data and negotiate for content [11].

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

There are some advantages and disadvantages of both the protocols but we need to take care while selecting protocols and selection depends on the application.

REFERENCES

1. Kevin Ashton,"That 'Internet of Things' Thing", Jun, 2009, https://www.rfidjournal.com/articles/view?4986
2. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.
3. Garcia-Morchon, O., Falck, T., Heer, T., & Wehrle, K. (2009, July). Security for pervasive medical sensor networks. In Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009. MobiQuitous' 09. 6th Annual International(pp. 1-10). IEEE.
4. Wu, M., Lu, T. J., Ling, F. Y., Sun, J., & Du, H. Y. (2010, August). Research on the architecture of Internet of Things. In Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on (Vol. 5, pp. V5-484). IEEE.
5. Brachmann, M., Garcia-Morchon, O., & Kirsche, M. (2011). Security for practical coap applications: Issues and solution approaches. GI/ITG KuVS Fachgesprch Sensornetze (FGSN). Universitt Stuttgart.
6. Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., Mccann, J., & Leung, K. (2013). A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. IEEE Wireless Communications, 20(6), 91-98.
7. Chen, X. (2014). Constrained application protocol for internet of things. URL: https://www. cse. wustl. edu/~ jain/cse574-14/ftp/coap.
8. Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., & Razafindralambo, T. (2011). A survey on facilities for experimental internet of things research. IEEE Communications Magazine, 49(11), 58-67.
9. Luo, H., Ci, S., Wu, D., Stergiou, N., & Siu, K. C. (2010). A remote markerless human gait tracking for e-healthcare based on content-aware wireless multimedia communications. IEEE Wireless Communications, 17(1).
10. Nussbaum, G. (2006, July). People with disabilities: assistive homes and environments. In International Conference on Computers for Handicapped Persons (pp. 457-460). Springer, Berlin, Heidelberg.
11. Alkar, A. Z., & Buhur, U. (2005). An Internet based wireless home automation system for multifunctional devices. IEEE Transactions on Consumer Electronics, 51(4), 1169-1174.
12. Darianian, M., & Michael, M. P. (2008, December). Smart home mobile RFID-based Internet-of-Things systems and services. In Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on (pp. 116-120). IEEE.
13. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.
14. Yun, M., & Yuxin, B. (2010, June). Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. In Advances in Energy Engineering (ICAEE), 2010 International Conference on (pp. 69-72). IEEE.
15. Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. Computer networks, 51(4), 921-960.
16. Liqiang, Z., Shouyi, Y., Leibo, L., Zhen, Z., & Shaojun, W. (2011). A crop monitoring system based on wireless sensor network. Procedia Environmental Sciences, 11, 558-565.
17. Moritz, G., Golatowski, F., & Timmermann, D. (2011, October). A lightweight SOAP over CoAP transport binding for resource constraint networks. In Mobil\e Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on (pp. 861-866). IEEE.
18. Guinard, D., Ion, I., & Mayer, S. (2011, December). In search of an internet of things service architecture: REST or WS-*? A developers' perspective. In International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (pp. 326-337). Springer, Berlin, Heidelberg.
19. Klas, G., Rodermund, F., Shelby, Z., Akhouri, S., & Hoeller, J. Lightweight M2M: Enabling Device Management and Applications for the Internet of Things. 2014.
20. Weber, R. H. (2010). Internet of Things–New security and privacy challenges. Computer law & security review, 26(1), 23-30.
21. Hunkeler, U., Truong, H. L., & Stanford-Clark, A. (2008, January). MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on (pp. 791-798). IEEE.
22. Lee, S., Kim, H., Hong, D. K., & Ju, H. (2013, January). Correlation analysis of MQTT loss and delay according to QoS level. In Information Networking (ICOIN), 2013 International Conference on (pp. 714-717). IEEE.
23. Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. Ad hoc networks, 10(7), 1497-1516.
24. Karakostas, B., & Bessis, N. (2016, March). Intelligent Brokers in an Internet of Things for Logistics. In Proceedings of the International Conference on Internet of things and Cloud Computing (p. 5). ACM.
25. Stankovic, J. A. (2014). Research directions for the internet of things. IEEE Internet of Things Journal, 1(1), 3-9.
26. Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. IEEE Internet of Things journal, 1(1), 22-32.
27. Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. IEEE Transactions on industrial informatics, 10(4), 2233-2243.
28. Leiner, Barry M., et al. "A brief history of the Internet." ACM SIGCOMM Computer Communication Review 39.5 (2009): 22-31.
29. Oen, H. M. (2015). Interoperability at the Application Layer in the Internet of Things (Master's thesis, NTNU).
30. Song, Y. (2013). Security in Internet of Things.
31. Wang, W. (2011). The research and development of the Internet of Things technology. Information network security, 3, 53-56.
32. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.
33. Zhang, B., Zou, Z., & Liu, M. (2011, May). Evaluation on security system of internet of things based on fuzzy-AHP method. In E-Business and E-Government (ICEE), 2011 International Conference on (pp. 1-5). IEEE.
34. Lee, C. (2017). A survey of the World Wide Web evolution with respect to security issues (No. e2793v1). PeerJ Preprints.
35. Hammami, R., Kacem, Y. H., Souissi, S., Bellaaj, H., & Kacem, A. H. (2017). Weighted Priority Queuing: A New Scheduling Strategy for Web Services. Information Technology and Computer Science, 2, 11-17.
36. Haghi, M., Thurow, K., & Stoll, R. (2017). Wearable devices in medical internet of things: scientific research and commercially available devices. Healthcare informatics research, 23(1), 4-15.
37. Weerawarana, S., Curbera, F., Leymann, F., Storey, T., & Ferguson, D. F. (2005). Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more. Prentice Hall PTR.
38. Sutaria, R., & Govindachari, R. (2013). Making sense of interoperability: Protocols and Standardization initiatives in IOT. In 2nd International Workshop on Computing and Networking for Internet of Things.
39. Colitti, W., Steenhaut, K., & De Caro, N. (2011). Integrating wireless sensor networks with the web. Extending the Internet to Low power and Lossy Networks (IP+ SN 2011).
40. Mitton, N., Chaouchi, H., Noel, T., Gabillon, T. W. A., & Capolsini, P. (2016). Interoperability, safety and security in IoT. In Second international conference, InterIoT 2016 and third international conference, SaSeIoT.
41. Antunes, M., Gomes, D., & Aguiar, R. L. (2018). Towards IoT data classification through semantic features. Future Generation Computer Systems, 86, 792-798.
42. Paulraj, G. J. L., Francis, S. A. J., Peter, J. D., & Jebadurai, I. J. (2018). Resource-aware virtual machine migration in IoT cloud. Future Generation Computer Systems, 85, 173-183.
43. Woo, M. W., Lee, J., & Park, K. (2018). A reliable IoT system for personal healthcare devices. Future Generation Computer Systems, 78, 626-640.
44. Nastic, S., Truong, H. L., & Dustdar, S. (2015). Sdg-pro: a programming framework for software-defined iot cloud gateways. Journal of Internet Services and Applications, 6(1), 21.
45. Batalla, J. M., & Gonciarz, F. Deployment of smart home management system at the edge: mechanisms and protocols. Neural Computing and Applications, 1-15.