ISSN (Online): 2583 648X

www.ijirid.in

IJIRID

International Journal of Ingenious Research, Invention and Development

Volume 1 | Issue 4 | June 2023

Scientific Journal Impact Factor (SJIF 2023): 3.647

DOI: 10.5281/zenodo.7878056

Analysis of Lightweight Approaches for IoT Protocols

Vijay Gulhane[1], Sagar Padiya[2]

[1] *Professor, Sipna College of Engineering and Technology, Amravati, India*

[2] *Research Scholar, Sipna College of Engineering and Technology, Amravati, India*

Abstract: Current IoT systems are characterized by using resource-constrained electronic devices therefore new lightweight protocols should be designed to reduce communications costs, computational costs, and storage costs. For such requirements, many researchers have already proposed various novel methodologies here we have analysed all of these. This research paper includes a survey of lightweight approaches for IoT protocols.

Keywords: Internet of Things, IoT, Lightweight Protocols, Security Protocol.

I. INTRODUCTION TO IOT

IoT has already enabled more effective monitoring, enhanced decision-making, and effectiveness in sectors such as eHealth, smart cities, defence and high-security applications, precision agriculture, transportation, or Industry 4.0. Current IoT systems are characterized by using resource-constrained electronic devices with reduced memory, limited communication capabilities, low computing power, and a high dependence on batteries. Therefore, they cannot support the implementation of complex security schemes [1-3]. These restrictions also apply to beacons and wearables, whose design represents a challenge for the IoT developer community. Some researchers [4] described ideal IoT devices as self-sufficient and energy-efficient computing elements that provide uninterrupted performance, QoS and long battery life. To create such ideal devices, the solutions devised must focus on three main areas: energy efficiency, scheduling optimization and lightweight protocols. Thus, new lightweight protocols must be designed to reduce communications costs (i.e., to minimize the number of exchanged messages), computational costs (i.e., performing lightweight operations) and storage costs [5].

II. LIGHTWEIGHT APPROACHES FOR IOT PROTOCOLS

The IoT has already enabled more effective monitoring, enhanced decision-making, and effectiveness in sectors such as eHealth, smart cities, defence and high-security applications, precision agriculture, transportation, or Industry 4.0. Current IoT systems are characterized by using resource-constrained electronic devices with reduced memory, limited communication capabilities, low computing power, and a high dependence on batteries, therefore they cannot support the implementation of complex security schemes. These restrictions also apply to beacons and wearable devices, whose design represents a challenge for the IoT developer community.

The [4] has been described ideal IoT devices as self-sufficient and energy-efficient computing elements that provide uninterrupted performance, QoS and long battery life. To create such ideal devices, the solutions devised must focus on three main areas: energy efficiency, scheduling optimization and lightweight protocols. Therefore, new lightweight protocols should be designed to reduce communications costs (i.e., to minimize the number of exchanged messages), computational costs (i.e., performing lightweight operations) and storage costs. For such requirements of lightweight wireless protocols, many researchers have proposed various novel designs.

Internet Engineering Task Force (IETF) set up several working groups, such as IPv6 over Low power Wireless Personal Area Networks (6LoWPAN), Routing over Lossy and Low-power Networks (RoLL) and Constrained

Restful Environment (CoRE), to carry out the research of low power IPv6 network and standardize communication protocols for resource-constrained devices such as 6LoWPAN, Routing Protocol for Low power and Lossy Networks (RPL), and Constrained Application Protocol (CoAP), etc. The following literature focused to review various lightweight protocols available.

1.1) Lightweight WSN MAC Protocol for Smart Home Networking Applications

In [6] authors focused on the unreliability problem of the IEEE 802.15.4 WSN MAC, which is caused by the contention-based MAC protocol used for channel access. This problem results in a low packet delivery ratio, particularly in a smart home network with only a few sensor nodes. This paper proposed a lightweight WSN protocol for a smart home, thus replacing the IEEE 802.15.4 protocol, which is highly complex and has a low packet delivery ratio. Subsequently, we describe the development of a discrete event system model for the WSN by using a GRAFCET and propose a development platform based on a reconfigurable FPGA for reducing fabrication cost and time. Finally, a prototype WSN controller ASIC chip without an extra CPU and with a proposed lightweight MAC was developed and tested.

Compared with an existing sensor hardware filter the proposed WSN MAC hardware controller possesses a unique feature: a wireless communication protocol based on a lightweight MAC protocol. The lightweight WSN MAC protocol can replace the IEEE 802.15.4 protocol, which is highly complex and has a low packet delivery ratio, for smart homes and intelligent buildings. The measurement results showed that the WSN hardware controller of the proposed solution can increase the packet delivery ratio by up to 100%, unlike solutions based on other MCUs. The obtained results attest to the efficiency of the proposed approach compared with other circuits designed for similar purposes.

1.2) Lightweight Security Protocol using Contiki OS

The deployment of IoT is advancing at a very fast pace, and relying on modified versions of the TCP/IP protocol suits. This rapid growth of the field is leaving several critical issues such as quality of service (QoS) and security of the delivered data. In [7] authors first attempted to develop and improve the performance of IoT networks with the use of IPsec protocol written inside Contiki OS. The authors tackle issues regarding QoS and the security of data by proposing a data delivery scheme that improves the QoS of classified data. The author proposed a solution that relies on differentiating the priority of the delivered data and giving preferences to secured and user-defined high-priority traffic with the purpose to test the possibility of improving the quality of services provided by the data link layer to the IoT applications.

The author proposed a solution that is denoted as Secured Traffic Priority Differentiation (STPD), which is made to support any application and is implemented at the MAC sub-layer. The proposed solution was tested in a virtual environment that simulates real scenarios using the Contiki operating system, using the Cooja simulator. At the simulation three major network performance metrics were tested; channel utilization, network latency, and packet delivery ratio. As a result, the proposed model successfully achieves the goals of the research by improving packet prioritization, enhancing resource utilization, upgrading QoS support, and improving the performance of (the IPsec) security protocol. Also outperforms the standard IEEE 802.15.4 MAC protocol regardless of the number of intermediate nodes that exist between the sender and receiver.

The simulation results demonstrated a significant improvement of the proposed solution over the Carrier Sense Multiple Access Collision Avoidance, (CSMA/CA), by 20%. The proposed solution worked to improve the channel utilization, data reliability and decreased latency of high-priority traffic, and low-priority traffic.

1.3) Datagram Transport Layer Security Version (DTLS)

The [8] specifies version 1.2 of the Datagram Transport Layer Security (DTLS) protocol. The DTLS protocol provides communications privacy for datagram protocols. It allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. The DTLS protocol is based on the TLS protocol and provides equivalent security guarantees. Datagram semantics of the underlying transport are preserved by the DTLS protocol.

TLS is the most widely deployed protocol for securing network traffic. It is widely used for protecting web traffic and for e-mail protocols such as IMAP and POP. The primary advantage is that it provides a transparent connection-oriented channel. Thus, it is easy to secure an application protocol by inserting TLS between the application layer and the transport layer. However, TLS must run over a reliable transport channel typically TCP. Therefore, it cannot be used to secure unreliable datagram traffic.

An increasing number of application layer protocols have been designed that use UDP transport. Protocols such as the Session Initiation Protocol (SIP and electronic gaming protocols are increasingly popular. The basic design philosophy of DTLS is to construct "TLS over datagram transport". The reason that TLS cannot be used directly in datagram environments is simply that packets may be lost or reordered. Unreliability creates problems for TLS at two levels:

1. TLS does not allow independent decryption of individual records.
2. The TLS handshake layer assumes that handshake messages are delivered reliably and break if those messages are lost.

1.4) IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)

The [9] defined low-power wireless personal area networks (LoWPANs) to comprise devices that conform to the IEEE 802.15.4-2003 standard by IEEE802.15.4. IEEE 802.15.4 devices are characterized by short range, low bit rate, low power, and low cost. Many of the devices employing IEEE 802.15.4 radios will be limited in their computational power, memory, and/or energy availability.

[10] gives an overview of LoWPANs and describes how they benefit from IP and IPv6 networking. It describes LoWPAN requirements regarding the IP layer and the above and spells out the underlying assumptions of IP for LoWPANs. Finally, it describes problems associated with enabling IP communication with devices in a LoWPAN, and defines goals to address these in a prioritized manner. Admittedly, not all items on this list may be necessarily appropriate tasks for the IETF. Nevertheless, they are documented here to give a general overview of the larger problem. A LoWPAN is a simple low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements. A LoWPAN typically includes devices that work together to connect the physical environment to real-world applications, e.g., wireless sensors. LoWPANs conform to the IEEE 802.15.4-2003 standard [IEEE802.15.4].

Some of the characteristics of LoWPANs are as follows:

1. Small packet size. Given that the maximum physical layer packet is 127 bytes, the resulting maximum frame size at the media access control layer is 102 octets. Link-layer security imposes further overhead, which in the maximum case (21 octets of overhead in the AES-CCM-128 case, versus 9 and 13 for AES-CCM-32 and AES-CCM-64, respectively), leaves 81 octets for data packets.
2. Support for both 16-bit short or IEEE 64-bit extended MAC addresses.
3. Low bandwidth. Data rates of 250 kbps, 40 kbps, and 20 kbps for each of the currently defined physical layers (2.4 GHz, 915 MHz, and 868 MHz, respectively).
4. Topologies include star and mesh operation.
5. Low power, typically, some or all devices are battery operated.
6. Low cost, typically devices are associated with sensors, switches, etc. This drives some of the other characteristics such as low processing, low memory, etc. Numerical values for "low" are elided on purpose as costs changes over time.

7. Many devices are expected to be deployed during the lifetime of the technology. This number is expected to dwarf the number of deployed personal computers, for example.

8. Location of the devices is typically not predefined, as they tend to be deployed in an ad-hoc fashion. Furthermore, sometimes the location of these devices may not be easily accessible. Additionally, these devices may move to new locations.

9. Devices within LoWPANs tend to be unreliable due to a variety of reasons: uncertain radio connectivity, battery drain, device lockups, physical tampering, etc.

10. In many environments, devices connected to a LoWPAN may sleep for long periods to conserve energy, and are unable to communicate during these sleep periods.

1.5) The Constrained Application Protocol (CoAP)

The application layer is responsible for data formatting and presentation. The application layer on the Internet is typically based on HTTP. However, HTTP is not suitable in resource-constrained environments because it is verbose in nature and thus incurs a large parsing overhead. The Constrained Application Protocol (CoAP) is presented in [11] as a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks. The nodes often have 8-bit microcontrollers with small amounts of ROM and RAM, while constrained networks such as IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) often have high packet error rates and a typical throughput of 10s of kbit/s. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.

CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types. CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments. The interaction model of CoAP is like the client/server model of HTTP. However, machine-to-machine interactions typically result in a CoAP implementation acting in both client and server roles. A CoAP request is equivalent to that of HTTP and is sent by a client to request an action (using a Method Code) on a resource (identified by a URI) on a server. The server then sends a response with a Response Code; this response may include a resource representation.

CoAP has many features such as web protocol fulfilling M2M requirements in constrained Environments, UDP binding with optional reliability supporting unicast and multicast requests, asynchronous message exchanges, low header overhead and parsing complexity, URI and content-type support, simple proxy and caching capabilities, a stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP and security binding to Datagram Transport Layer Security (DTLS).

1.6) Lithe: Lightweight Secure CoAP for the IoT

The IoT enables a wide range of applications that have to perform potentially critical actuating and sensing tasks. For communication at the application layer, resource-constrained devices can employ the CoAP that is currently being standardized at the Internet Engineering Task Force. CoAP-enabled hosts will be an integral part of the IoT. Furthermore, real-world deployments of CoAP-enabled devices require security solutions. To protect the transmission of sensitive information, secure CoAP mandates the use of datagram transport layer security (DTLS) as the underlying security protocol for authenticated and confidential communication. DTLS, however, was originally designed for comparably powerful devices that are interconnected via reliable, high-bandwidth links. DTLS is the standard protocol to enable secure CoAP (CoAPs).

In [12] authors presented Lithe: Lightweight Secure CoAP for the IoT, an integration of DTLS and CoAP for the IoT. It additionally proposed a novel DTLS header compression scheme that aims to significantly reduce energy consumption by leveraging the 6LoWPAN standard. Most importantly, the proposed DTLS header compression scheme does not compromise the end-to-end security properties provided by DTLS. Simultaneously, it considerably reduces the number of transmitted bytes while maintaining DTLS standard compliance. The authors evaluated an approach based on a DTLS implementation for the Contiki operating system, the results show significant gains in terms of packet size, energy consumption, processing time, and network-wide response times when compressed DTLS is enabled.

1.7) Cluster-based CoAP for Message Queueing in IoT Networks

The IoT must exchange information between devices. For constrained IoT network environments, lightweight application protocols have recently been proposed, which include CoAP and MQTT. However, those protocols still have scalability problems in the network with many sensors. In MQTT, each subscriber must communicate with the broker even though a lot of subscribers have the same interests in topics. It may cause the traffic overhead problem and the energy-wasting of constrained devices. In the meantime, CoAP is a request-response model that is also difficult to manage for many devices. However, the CoAP has also a similar scalability problem to MQTT. In [13] authors proposed a simple extension of CoAP using a clustering approach, in which a set of CoAP sensors are grouped into a cluster, and a cluster head is used for message aggregation and transmission for the sensors associated with the cluster. The cluster head will manage many sensors. By using this clustering approach, it is easy to transfer messages containing topics.

In this scheme, the cluster head is used to reduce the amount of traffic concentrated on the server. This scheme can also be used for CoAP-based many-to-many communication. By implementation and experimentation, the author shows that the proposed cluster-based CoAP provides performance gains over the existing CoAP and MQTT protocols in terms of bandwidth consumption and transmission time.

1.8) CoAP-based Constrained Web Things

A smart home system is usually constructed by integrating various sensors, actuators, and software. Meanwhile, the concept of IoT becomes popular so that the concept of the Web of Things (WoT) also emerges out of the application layer. In a WoT network, all things can be accessed and operated via HTTP so that the development of smart home applications is simplified. The Web Things is described based on the Web Things Model (WTM), which has been submitted to W3C. In [14] authors proposed a solution for dealing with the WoT findability issues by combining the WTM and CoAP.

Different IoT devices have different protocols which creates a burden for the developers to learn various protocols and technologies. The concept of WoT is a relief of burden for developers, as all Web Things are programmable using RESTful Web API. However, up to now the ecosystem of programming IoT via Web API is still under development. One of the core issues is that there still lacks a standard way to describe the object model of Web Things. As a result, even though all Web Things are programmable, the ways to find out the services these things provide and the locations of these services are mostly proprietary. Given the above problem, the authors designed a tool to deal with the findability issues of a CoAP network, and adopt the WTM as the standard object model for describing CoAP-based Web Things. Next, inspired by SSDP, the authors proposed a lightweight service discovery mechanism for the CoAP-based Web Things and finally, designed a tool called the WTM Browser that helps WoT developers to be aware of the capabilities and locations of CoAP-based Web Things.

1.9) Lightweight M2M (LWM2M)

LWM2M is an emerging Open Mobile Alliance standard that defines a fast deployable client-server specification to provide various machine-to-machine services. It acts as an OMA device management (OMA-DM) successor for use in M2M and provides efficient device management as well as security workflow for IoT applications using the same protocol, thus offering enhanced simplicity. It provides both efficient device management as well a security workflow for IoT applications, making it especially suitable for use in constrained networks. However, most of the ongoing research activities on this topic focus on the server domain of LWM2M. Enabling relevant LWM2M functionalities on the client side is critical and challenging as well since these end nodes are invariably resource-constrained.

The [15] addresses those issues by the client-side architecture for LWM2M and its complete implementation framework carried out over Contiki-based IoT nodes. It presented a lightweight IoT protocol stack that incorporates the LWM2M client engine architecture and its interfaces. The widely used CoAP provides in-built binding for LWM2M, thus making it particularly appealing for the IoT. The implementation is based on the recently released OMA LWM2M v1.0 specification and supports OMA, IPSO as well as third-party objects. It employed a real-world application scenario to validate its usability and effectiveness. The results obtained indicated that the memory footprint overheads incurred due to the introduction of LWM2M into the client-side IoT protocol stack are around 6-9%, thus making this implementation framework very appealing to constrained devices.

1.10) Message Queue Telemetry Transport (MQTT)

MQTT is a published protocol that runs over TCP. It was developed by IBM primarily as a client/server protocol. The clients are publishers/subscribers and the server acts as a broker to which clients connect through TCP. Clients can publish or subscribe to a topic. This communication takes place through the broker whose job is to coordinate subscriptions and authenticate the client for security. MQTT is a lightweight protocol, which makes it suitable for IoT applications. But because it runs over TCP, it cannot be used with all types of IoT applications. Moreover, it uses text for topic names, which increases its overhead [16].

MQTT-S/MQTT-SN is an extension of MQTT, which is designed for low-power and low-cost devices. It is based on MQTT but has some optimizations for WSNs as follows. The topic names are replaced by topic IDs, which reduce the overheads of transmission. Topics do not need registration as they are pre-registered. Messages are also split so that only the necessary information is sent. Further, for power conservation, there is an offline procedure for clients who are in a sleep state. Messages can be buffered and later read by clients when they wake up. Clients connect to the broker through a gateway device, which resides within the sensor network and connects to the broker. In MQTT, each subscriber must communicate with the broker even though a lot of subscribers have the same interests in topics. It may cause traffic overhead and the energy-wasting of constrained devices.

1.11) XMPP (Extensible Messaging and Presence Protocol)

An open technology “Jabber” was launched in 1999 for instant messaging and presence, the IETF published work done by the Instant Messaging and Presence Protocol (IMPP) working group in 2000, Jabber protocol with the Internet standard process is known as the Extensible Messaging and Presence Protocol (XMPP) and finally in 2004 the IETF publishes RFC 3920 and RFC 3921 which defines the core functionality of XMPP. Application-layer wise XMPP has similarities with Simple Mail Transfer Protocol (SMTP). However, the main difference between XMPP and SMTP is the usage. Namely, in XMPP clients can send messages in chat (unicast), group chat (multicast) and headline (broadcast) ways [17]. XMPP is an open standard and hence provides various open-source implementations, standard, proven, decentralized, secure, extensible, flexible, and diverse services. The XMPP is not dependent on any specific type of network architecture, it is usually implemented on client-server architecture. The architecture of the XMPP network is like email, anyone can run their XMPP server and there is no central master server. A wide range of applications including instant messaging, presence, lots of private chat, voice and video calls, collaboration, light middleware, content, and generalized routing of XML data. It performs person-to-person communication using TCP. It uses XML text format as a native type.

1.12) A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices

The XMPP is a perfect protocol to solve the interoperability issues between heterogeneous networks hence XMPP is extended to the IoT. Initially, the XMPP protocol was designed for the Internet where the equipment was rich in resources, but now by considering the IoT characteristics, the XMPP protocol needs to be optimized to meet the IoT requirements. In [18], by staying true to the IoT visions, the authors proposed a lightweight XMPP publish scheme for resource-constrained IoT devices to perform data exchange either periodically or upon any value change. According to the subscriber’s needs, the publisher can adjust the data information published to the server. Inherit the merits of XEP-0060, the server maintains and manages the publish/subscribe relationships with multiple subscribers, as well as distributes the received data sent by the publisher to all subscribers, yielding less complexity of publication and energy efficiency.

In [18] inheriting the advantages of XEP-0060, the server maintains and manages the publish/subscribe relationships with multiple subscribers and forwards data to clients, so that most of the system complexities reside on the server’s side. Besides, considering the practical application in resource-constrained networks, the publisher does not periodically or conditionally publish all objects and their attributes, but triggers the services based on the publication type, events and attributes subscribed by subscribers to save energy for resource-constrained devices. Furthermore, the achieved test results and performance evaluations validate the objective of improving the XMPP publish/subscribe scheme in IoT. The experimental results demonstrated that, optimizing and improving XMPP publish/subscribe scheme for resource-constrained IoT devices and show the simplicity and efficiency of this scheme.

V. REVIEWS OF LIGHTWEIGHT PROTOCOLS

The lightweight protocol must have the following three features:

1. Low storage cost (does not occupy much memory to make it possible for upload to the resource-constrained beacons),
2. Low computational cost (short frames help to speed up the computational tasks at both the communications link ends) and Low communication cost or low energy consumption (minimum number of frames to transmit).

Many lightweight protocols were proposed at the different layers of communication to reduce overhead in data transmission in the lower layers of the communications stack. Various lightweight protocols are available but generally at the Physical Layer, Network Layer, or MAC Layer hence these are not sufficient for effective communication. Lightweight protocols at the application layer are also needed to reduce power consumption while transmitting the necessary messages.

The first proposal was for a lightweight “WSN MAC” protocol that replaced “IEEE 802.15.4” (complex and has a low packet-delivery ratio) and attempted the packet-delivery ratio up to 100%, another proposal was regarding the use of CoAP in 6LoWPAN for the connection of Bluetooth Smart devices. One was for a network layer to emphasize the role of 6LoWPAN” to include the use of REST services and Tiny OS-based nodes, the next was in which network and security layer protocols are combined, integrating DTLS with 6LoWPAN called 6LoWPAN-NHC, which encodes different DTLS headers.

Recently lightweight protocols at the application layer, among all the available have had a major impact on the growth of the IoT, the major contribution is MQTT, CoAP, XMPP, and AMQP. Several researchers also proposed protocols for the application and the presentation layers, for example, W3C has proposed and implemented the WTM for the WoT in CoAP-based prototypes to discover and describe the different elements of a smart home environment, and LWM2M protocol supported by the OMA.

VIII. CONCLUSION

The development of lightweight protocols for IoT is still a challenge for the IoT community, as it shall achieve a balance between the high-security level, efficiency, computational and communication cost. Given the diversity of IoT beacons currently available, which differ in terms of hardware constraints and communication technologies, it was necessary to create a protocol that allows for fulfilling the following requirements: i) Low storage cost or low footprint. The developed firmware cannot occupy much memory and it should be possible to upload it to resource-constrained beacons. ii) Low computational cost. Frames must be short, which allows for speeding up computational tasks at both ends of the communications link. iii) Low communication cost. The number of frames to be transmitted must be minimized as much as possible to reduce communications costs in terms of energy consumption.

REFERENCES

[1] Suárez-Albela, M.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. A Practical Evaluation of a High-Security Energy-Efficient Gateway for IoT Fog Computing Applications. Sensors 2017, 17, 1978, doi:10.3390/s17091978.

[2] Seneviratne, S.; Hu, Y.; Nguyen, T.; Lan, G.; Khalifa, S.; Thilakarathna, K.; Hassan, M.; Seneviratne, A. A Survey of Wearable Devices and Challenges. IEEE Commun. Surv. Tutor. 2017, 19, 2573–2620

[3] Georgiou, K.; Xavier-de-Souza, S.; Eder, K. The IoT energy challenge: A software perspective. IEEE Embed. Syst. Lett. 2017, doi:10.1109/LES.2017.2741419.

[4] Ejaz, W.; Naeem, M.; Shahid, A.; Anpalagan, A.; Jo, M. Efficient Energy Management for the Internet of Things in Smart Cities. IEEE Commun. Mag. 2017, 55, 84–91.

[5] Das, A.K.; Zeadally, S.; Wazid, M. Lightweight authentication protocols for wearable devices. Comput. Electr. Eng. 2017, 63, 196–208.

[6] Chen, C.H.; Lin, M.Y.; Lin, W.H. Designing and Implementing a Lightweight WSN MAC Protocol for Smart Home Networking Applications. J. Circuits Syst. Comput2017

[7] Contiki. Available online: http://www.contiki-os.org/ (accessed on 20 November 2017).

[8] Datagram Transport Layer Security Version 1.2. Available online: https://tools.ietf.org/html/rfc6347/ (accessed on 20 November 2017).

[9] IETF, IPv6 over Low Power WPAM (6LoWPAN). Available online: https://datatracker.ietf.org/wg/ 6lowpan/charter/ (accessed on 20 November 2017).

[10] Castellani, A.P.; Bui, N.; Casari, P.; Rossi, M.; Shelby, Z.; Zorzi, M. Architecture and protocols for the Internet of Things: A case study. In Proceedings of the 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2010), Mannheim, Germany, 29 March–2 April 2010; 678–683.

[11] The Constrained Application Protocol (CoAP). Available online: https://tools.ietf.org/html/rfc7252 (accessed on 20 November 2017).

[12] Raza, S.; Shafagh, H.; Hewage, K.; Hummen, R.; Voigt, T. Lithe: Lightweight Secure CoAP for the Internet of Things. IEEE Sens. J. 2013, 13, 3711–3720.

[13] Choi, D.K.; Jung, J.H.; Kang, H.W.; Koh, S.J. Cluster-based CoAP for message queueing in Internet-of-Things networks. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), Phoenix Park, PyeongChang, Korea, 19–22 February 2017; pp. 584–588.

[14] Cheah,W.T.; Liao, C.F. On Findability Issues of ConstrainedWeb of Things in a Smart Home Environment. In Proceedings of the 2017 International Conference on Platform Technology and Service (PlatCon), Busan, Korea, 13–15 February 2017; pp. 1–6.

[15] Rao, S.; Chendanda, D.; Deshpande, C.; Lakkundi, V. Implementing LWM2M in constrained IoT devices. In Proceedings of the 2015 IEEE Conference on Wireless Sensors (ICWiSe), Melaka, Malaysia, 24–26 August 2015; pp. 52–57.

[16] Neven Nikolov, Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems, Proc. XXIX International Scientific Conference Electronics - ET2020, September 16 - 18, 2020, Sozopol, Bulgaria.

[17] Oral Gurel and Mehmet Ulas Çakir, Promising XMPP Based Applications for Military and Defense Systems, 2013 IEEE 37th Annual Computer Software and Applications Conference.

[18] Heng Wang, Daijin Xiong, Ping Wang, and Yuqiang Liu, A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices, IEEE Access (Volume: 5), Electronic ISSN: 2169-3536, Page(s): 16393 – 16405, 2017

[19] Padiya, S. D., & Gulhane, V. S. (2020). IoT and BLE Beacons: Demand, Challenges, Requirements, and Research Opportunities-Planning-Strategy. 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 125-129). Gwalior, India: IEEE Xplore.