DIFFERENT TYPE OF PROTOCOL AT IOT APPLICATION LAYER

[1]Yudhvir Singh, [2]Jogender Singh
[1] Professor & Dean, [2]Assistant Professor
[1]MDU, ROHTAK, INDIA
[2]Computer Science, GCW, ROHTAK, INDIA

Abstract: There are tremendous amount of work happening in the field of IoT. All the smart device (microcontroller and MCU) work together with internet create a new smart world that include machine to machine and human to machine communication. For this communication different type of computer or IoT specified protocols are used. Application layer is higher layer in IoT on the client side. This proposed paper discusses almost all common application layer protocols that are used in IoT like MQTT, MQTT-SN, SMQTT, CoAP, XMPP, AMQP, DDS, SMCP, RESTful HTTP, HTTP/2 etc.

IndexTerms - IoT, MCU, MQTT, MQTT-SN, SMQTT, CoAP, XMPP, AMQP, DDS, SMCP, RESTful HTTP, HTTP/2.

1. Introduction

A smart vision of the world involves computer science, computer engineering, and electrical engineering. There is a greater need of interactions among these communities by which a speedy progress in smart world can be achieved. The outcome of these interactions between technologies is IoT.

1.1 Internet of Things (IoT)

The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and Internet connectivity, which enables these objects to collect and exchange data [26]. IoT allow things (Computer, Mobile Phone, embedded Microcontroller and MCU etc.) act smartly and collaborative make a decisions that are beneficial to a certain applications. They allow things to hear, see, think or act by allowing them to communicate and coordinate with others in order to make decisions that can be as critical as saving lives or buildings. It transforms "things" from passive node to active node that can take critical decision.

The electronic technology of ubiquitous computing, embedded sensors, light communication and internet protocols allow IoT to provide it's significant. Every smart world thing that we are used in life, are using the IoT like Smart Home, Smart Cities, Wearable, Smart Grids, Industrial Internet, Connected Car, Connected Healthcare, Smart Retail, Smart Supply Chain, Smart Farming.

IoT protocol stack is different than OSI model and TCP/IP protocol stack.

Table1.1

Application Layer Protocol - CoAP, MQTT, XMPP, AMQP, RESTFUL
Transport Layer Protocol - UDP,DTLS
Internet Layer Protocol - IPv6/IP Routing,6LoWPAN
Link layer/Network Layer Protocol - IEEE 802.15

IoT protocol model Stack shown in Table 1.1 where different layers of model like Application layer, Transport, Internet layer and Physical/Link. The integration of embedded devices with the Internet also introduces several challenge and issue because many of the existing Internet technologies and protocols were not designed to provide service for IoT. From past decades there many efforts has done to enable the extension of Internet technologies to IoT. Starting from proprietary protocols and architectures now IETF and many other standard organizations moving towards standardized IP-based protocols.

In this survey a introduction of different application layer protocols in details, where mainly a replacement of TCP/IP application layer protocols in IoT framework. The second section in this proposed paper discusses almost all common application layer protocols that are used in IoT. At last there are some suggestions about future work and enhancements in IoT application layer protocol.

2. APPLICATION LAYER PROTOCOL IN IoT

The application layer in the computer Internet is typically based on HTTP protocol. However, HTTP is not suitable in lightweight node like embedded microprocessor, MCU because a large parsing overhead and need heavy processing and memory space. So, there are many alternate proprieties and public standard protocols have been developed for IOT environments. Some of the

popular IOT application layer protocols are as follow - MQTT-SN, SMQTT, CoAP, XMPP, AMQP, DDS, SMCP, RESTful HTTP, HTTP/2 etc.
Application layer protocols are basically differentiate into 2 categories.
1. Request-Response Model
2. Publish-Subscribe Model
Traditional computer system used Request-Response model where client sent a request to the server, server process on the request and sent the response to the client. Whereas in IoT a new alternate of protocol called Publish-Subscribe Model is popular where all the client are subscribe for a specific service from the publisher server. That provides a better performance in IOT because any changes in data value get better response from machine to machine or human to machine communication. Different Application Layer protocols are following.

2.1 MESSAGE QUEUE TELEMETRY TRANSPORT (MQTT)

Message queue telemetry transport (MQTT) protocol is based on Publish-Subscribe Model. It's similar to the client-server model. It was design by IBM and now standardized by OASIS (MQTT v3.3.3 and MQTT 5.0).This protocol made for IoT environment having low power, limited computation capability and memory, and limited bandwidth. MQTT protocol use TCP/IP protocol of traditional internet with a aims of reduce bandwidth requirement (Small Message size from 2 byte).
IANA reserved TCP/IP port 1883 for MQTT protocol and port 8889 for MQTT over SSL. From v3.1 MQTT also provide the support of username and password functionality. For the reliable message delivery, MQTT three option of Quality of Service (QoS):

- At Most Once - Here message sent only once believe on the best effort of the network and does not wait for an acknowledgment. It is the least level of quality for service.
- At least once - Here message sends at least once or resent till an acknowledgment message is received.
- Exactly Once - Here 2-way handshaking is done before sent a message once and only once. It is the highest level of quality for service.

A client using the MQTT protocol may be a publisher of information or a subscriber of the information. A broker in between publisher and subscriber that take the information from client as data and transfer the data to clients that are subscribed for that specific information. MQTT use four type of message.

1. Connect- For Connection establishment and Authentication
2. Communication- For communication between M2M or H2M after connection establishment
3. Disconnect-For disconnect the session like TCP.
4. Publish - To give the message to subscriber immediately.

The popularity of MQTT protocol is due to its simplicity, availability with many devices and the no need of high CPU and memory usage.

2.2 CONSTRAINED APPLICATION PROTOCOL (COAP)

Constrained application protocol (CoAP) is based on request/response model design by IETF for constrained (limited) environment having low capability in RAM or CPU, and constrained network, such as lower power using wireless personal area network (WPAN).The main interest of IETF to design this protocol is machine to machine (m2m) applications and the automation of systems to reduce overhead, enhance packet delivery, and to increase the simplicity of work using UDP over IP. CoAP also supports publisher/subscribe architecture, using a Universal Resource Identifier (URI), where architecture provides multicast communications and the publisher sends the message to multi-subscribers can get the message and takes appropriate actions. CoAP use Asynchronous communication like UDP in TCP/IP Model. It use two layers, messaging layer to achieve reliability based on UDP and request/response layer to act the interactions and communication. CoAP protocol message size start from 4 Byte in length and uses different types of massages like Conformable Message, Non-conformable Message, Acknowledgement Message, Reset Message, Piggybacked Response, Separate Response, and Empty Message. It is simple and has lower consumption of CPU and memory but bad packet delivery because of UDP, and its inability to be used on complex data type.

2.3 EXTENSIBLE MESSAGING AND PRESENCE PROTOCOL (XMPP)

XMPP is a based on XML which is a markup language for encoding documents that is readable by both human and machine. Extensible Messaging and Presence Protocol (XMPP) developed by Jabber open source community and after that standardized by IETF. XMPP is an open decentralized instant messaging protocol that supports small messages with low latency. It is open, free, public and easy to understand that why multiple implementations of server, client, and server component and code libraries are available on internet that's making this protocol popular and a good choice for IoT communications and messaging. It supports both request/response and publish/subscribe models. XMPP provided high scalability by decentralized architecture because thousand of XMPP server running on internet and millions of user of instant messaging like Google Talk use it.
In IoT XMPP allows real-time and scalable networking between devices or things.XML tag may cause network traffic overhead, and needs high consumption of bandwidth and high CPU usage, no guarantee of QoS give some restriction to use XMPP is IoT.

2.4 REPRESENTATIONAL STATE TRANSFER (RESTFUL SERVICES)

The Representational State Transfer (REST) architectural is a distributed hypermedia systems which use the state of the distributed application and assigning it to the different components (i.e., clients and servers). Clients can navigate through URI and modify state to achieve their goals is defined through hypermedia controls like links and forms. In REST, IoT node gets the current state, modifying the state on the client side, and transferring the new state to the server in the form of new representations. REST use different representations such

- Text/Plain for simple UTF-8 text
- Application/octet-stream for arbitrary binary data
- Application/json for the JSON
- Application/senml+json for Sensor Markup Language (SenML) formatted data
- Application/cbor" for CBOR
- Application/exi for EXI.

Currently HTTP and CoAP both are using REST architecture and uses the same methods GET, PUT, DELETE, POST, and OPTION, to the request or response of resource usage. POST method is used to create a resource at server side and GET function are use to retrieve resources at client node, while PUT method is to update the state of resource after change resource state at client node and DELETE method is to remove the resource from server. RESTFUL provide a better services for M2M and H2M communications in the IoT .In 2018 IETF also proved a guidance for RESTful design for IoT system. RESTful HTTP over TCP is mainly used in connected consumer's premises device like home energy management.

2.5 ADVANCED MESSAGE QUEUING PROTOCOL (AMQP)

Advanced Message Queuing Protocol (AMQP) is a open standard publisher/subscribe model protocol for message oriented middleware with peer to peer communication functionality. It was standardized by OASIS in 2012 and ISO certified in 2014. It was design by a organization of many banking and computer corporate to use in business and commercial platforms. AMQP protocol can connect across different organization application, different application technology platform, different time and different distance poor network. It provides an open, reliable, standard, interpretable, reliable and secure messaging.

AMQP small message size is 8-bytes in length. It uses TCP at transport layer to ensure reliability and TLS/SSL and SASL for security purpose. AMQP message along with a header is passed by the client to a broker or exchange. There is a single queue to which the message is passed by a producer node. From the exchange or broker, the message can be passed to unicast or multicast queues. Message header contains routing information and information about each byte of the message. It is the broker or exchange responsibility to read headers and receive, route and deliver messages to the client applications. The communication in this protocol remains point to point.

For providing so much service it has some restriction for constrained environment and real-time applications.

2.6 DATA DISTRIBUTION SERVICE (DDS)

Data Distribution Service (DDS) is a publish/subscribe model protocol design by OMG (Object Management Group) to support IoT applications and M2M communication without broker unlike MQTT and CoAP protocols. The reason in which of this protocol is a good choice in M2M and IoT refers to its ability to achieve QoS , reliability and deployed from low footprint devices to cloud.

DDS nodes produce information by create a topic like temperature, location, pressure etc and publish samples for related topic and delivered the samples to subscribers those are interested in that topic. DDS simulate the different application those need real time data exchange like air traffic control, small grid, robotics etc.

2.7 MQTT-SENSOR NETWORK (MQTT-SN)

MQTT-SN is a light-weight publisher /subscription protocol designed by IBM specifically for ZigBee based wireless sensor network or BLE 4.0. It is used to address specific wireless sensor network device which does not have TCP/IP stack on the top of it. It is very close to MQTT protocol with some update for wireless sensor communication environment having low bandwidth, high link failures, and short message length. MQTT is also optimized for the implementation on low-cost, battery-operated devices with limited processing and storage resources.

In MQTT-SN topic name is changed with topic ID so no need of repeated registration of Topic name. It also support sleep node with buffering where client can read the message when it wake up.

MQTT-SN architecture use three kinds of MQTT-SN components those are MQTT-SN clients, MQTT-SN gateways (GW), and MQTT-SN forwarders. MQTT-SN clients connect to a MQTT broker server via a MQTT-SN Gateway using the MQTT-SN protocol that work as a translator between MQTT-SN to MQTT and vice-versa.

MQTT-SN has 2 type of message header of size 1 byte and 3 byte. It also provide a lightweight security in messaging Because MQTT-SN is an update in MQTT for sensor network so it is fast, reliable and lightweight protocol for IoT.

2.8 STREAM TEXT ORIENTED MESSAGING PROTOCOL (STOMP)

It is the Simple or stream Text Orientated Messaging Protocol provides an interoperable wire format by which clients can communicate with any STOMP message broker to provide easy and widespread messaging interoperability among many languages, platforms and brokers. STOMP is a very simple and easy to implement protocol, coming from the HTTP school of design; the server side may be hard to implement well, but it is very easy to write a client to get you connected. For example you can use Telnet to login to any STOMP broker and interact with it!
Many developers have told us that they have managed to write a STOMP client in a couple of hours to in their particular language, runtime or platform into the STOMP network. So if your favored language/runtime of choice does not offer a good enough STOMP client don't be afraid to write one [53].

2.9 SECURE MESSAGE QUEUE TELEMENTRY TRANSPORT (SMQTT)

It is the extension of MQTT and MQTT-SN for security in machine to machine communication. This protocol use Key or Cipher text messaging. It is using lightweight Attribute Based Encryption over Elliptic Curve Cryptograph. It is better and lighter then MQTT because MQTT using SSL/TLS for secure messaging that need high processing and bandwidth. SMQTT use for stage in the session setup, encryption, publish, decryption. Client node or subscriber node register with the MQTT broker and resister with the Key. So when topic is publishing in encryption form then subscriber use his key to decrypt the message.

2.10 HTTP-2

It is new version of HTTP protocol of the internet design by IETF that has a compress header which is efficient and take less memory suitable for IoT. It support parallel connection with multiple node so a faster transmissions and low battery power communicate will be provide to IoT device.

4. CONCLUSION AND FUTURE WORK

In this paper we give the brief introduction of different application layer protocol. With different scenario of IoT environment IoT protocol give its signification. MQTT is best of messing in battery operated IoT environment. MQTT-SN best for ZigBee and BLE 4.0 sensor environment. CoAP is best when there is limited hardware and instant messaging.DDS and XMPP give their best when they are used in real-time data analysis. RESTful HTTP is better with limited hardware and battery operated environment. STOMP used for cross platform data communication.HTTP-2 also work fast with limited hardware as compare to http. So according to environment, available device, power source, cost, reliability and security different application layer protocol are used. Many of protocol are implemented and many of them are in implementation stage with different language and different platform of IoT device. There is a better scope in IoT application protocol that support different hardware device and cloud platform on internet. There are needs of application layer protocol that support AD-Hoc network and edge technology for IoT.

REFERENCES

1. About XMPP. (n.d.). Retrieved from https://xmpp.org/about/
2. Advanced Message Queuing Protocol. (n.d.). Retrieved from https://en.wikipedia.org/wiki/ Advanced_Message_Queuing_ Protocol
3. Ahmed, N., De, D., & Hussain, I. (2018). Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas. *IEEE Internet of Things Journal*, *5*(6), 4890-4899. doi:10.1109/jiot.2018.2879579
4. Al-Sabbagh, A. (2016). *Internet of Things and Big Data Analysis: Recent Trends and Challenges*.
5. Al-Sarawi, S., Anbar, M., Alieyan, K., & Alzubaidi, M. (2017). Internet of Things (IoT) communication protocols: Review. *2017 8th International Conference on Information Technology (ICIT)*. doi:10.1109/icitech.2017.8079928
6. AMQP is the Internet Protocol for Business Messaging. (n.d.). Retrieved from https://www.amqp.org/about/what
7. Application Layer Protocols for IOT : IOT Part 11. (n.d.). Retrieved from https://www.engineersgarage.com/Articles/IoT-Application-Layer-Protocols?PageSpeed=noscript
8. Architecture. (n.d.). Retrieved from https://www.amqp.org/product/architecture
9. Asghar, M. H., Mohammadzadeh, N., Negi, A., & Kazerouni, T. (2015). Principal ingredients and framework of Internet of Things (IoT). *2015 Twelfth International Conference on Wireless and Optical Communications Networks (WOCN)*. doi:10.1109/wocn.2015.8064520
10. Atmoko, R. A., Riantini, R., & Hasin, M. K. (2017). IoT real time data acquisition using MQTT protocol. *Journal of Physics: Conference Series*, *853*, 012003. doi:10.1088/1742-6596/853/1/012003
11. Chernyshev, M., Baig, Z., Bello, O., & Zeadally, S. (2018). Internet of Things (IoT): Research, Simulators, and Testbeds. *IEEE Internet of Things Journal*, *5*(3), 1637-1647. doi:10.1109/jiot.2017.2786639
12. Chiang, M., & Zhang, T. (2016). Fog and IoT: An Overview of Research Opportunities. *IEEE Internet of Things Journal*, *3*(6), 854-864. doi:10.1109/jiot.2016.2584538
13. CoAP. (April 3). Retrieved from https://coap.technology/
14. Constrained Application Protocol. (2011, August 22). Retrieved from https://en.wikipedia.org/wiki/ Constrained_Application_Protocol
15. DDS Protocol Architecture basics | DDS Protocol in IoT. (n.d.). Retrieved from http://www.rfwireless-world.com/Terminology/DDS-protocol-architecture.html

16. Donnal, J. (2018). Joule: A Real-Time Framework for Decentralized Sensor Networks. *IEEE Internet of Things Journal*, *5*(5), 3615-3623. doi:10.1109/jiot.2018.2815432
17. Dunko, G., Misra, J., Robertson, J., & Snyder, T. (2017). *A Reference Guide to the Internet of Things*.
18. Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., & Hindia, M. N. (2018). An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges. *IEEE Internet of Things Journal*, *5*(5), 3758-3773. doi:10.1109/jiot.2018.2844296
19. Frustaci, M., Pace, P., Aloi, G., & Fortino, G. (2018). Evaluating Critical Security Issues of the IoT World: Present and Future Challenges. *IEEE Internet of Things Journal*, *5*(4), 2483-2495. doi:10.1109/jiot.2017.2767291
20. Fysarakis, K., Askoxylakis, I., Soultatos, O., Papaefstathiou, I., Manifavas, C., & Katos, V. (2016). Which IoT Protocol? Comparing Standardized Approaches over a Common M2M Application. *2016 IEEE Global Communications Conference (GLOBECOM)*. doi:10.1109/glocom.2016.7842383
21. H. Amaran, M., S. Rohmad, M., H. Adnan, L., N. Mohamed, N., & Hashim, H. (2018). Lightweight Security for MQTT-SN. *International Journal of Engineering & Technology*, *7*(4.11), 223-226. doi:10.14419/ijet.v7i4.11.20811
22. Hedi, I., Speh, I., & Sarabok, A. (2017). IoT network protocols comparison for the purpose of IoT constrained networks. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. doi:10.23919/mipro.2017.7973477
23. Home | AMQP. (n.d.). Retrieved from https://www.amqp.org/
24. Hong, S., Kim, D., Ha, M., Bae, S., Park, S., Jung, W., & Kim, J. (2010). SNAIL: an IP-based wireless sensor network approach to the internet of things. *IEEE Wireless Communications*, *17*(6), 34-42. doi:10.1109/mwc.2010.5675776
25. Internet of Things (IoT). (n.d.). Retrieved from https://xmpp.org/uses/internet-of-things.html
26. Internet of things. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Internet_of_things
27. Jiang, Y., Zheng, X. S., & Huang, R. (2014). Application and Research for IOT Based on REST Style. *Applied Mechanics and Materials*, *556-562*, 5910-5913. doi:10.4028/www.scientific.net/amm.556-562.5910
28. Jiang, Y., Zheng, X. S., & Huang, R. (2014). Application and Research for IOT Based on REST Style. *Applied Mechanics and Materials*, *556-562*, 5910-5913. doi:10.4028/www.scientific.net/amm.556-562.5910
29. Kianoush, S., Raja, M., Savazzi, S., & Sigg, S. (2018). A Cloud-IoT Platform for Passive Radio Sensing: Challenges and Application Case Studies. *IEEE Internet of Things Journal*, *5*(5), 3624-3636. doi:10.1109/jiot.2018.2834530
30. Kim, H., & Choi, D. (2016). *Information Security Applications: 16th International Workshop, WISA 2015, Jeju Island, Korea, August 20-22, 2015, Revised Selected Papers*. Basingstoke, England: Springer.
31. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal*, *4*(5), 1125-1142. doi:10.1109/jiot.2017.2683200
32. Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., & Manzoni, P. (2016). Improving MQTT Data Delivery in Mobile Scenarios: Results from a Realistic Testbed. *Mobile Information Systems*, *2016*, 1-11. doi:10.1155/2016/4015625
33. Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., & Manzoni, P. (2016). Improving MQTT Data Delivery in Mobile Scenarios: Results from a Realistic Testbed. *Mobile Information Systems*, *2016*, 1-11. doi:10.1155/2016/4015625
34. Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., & Manzoni, P. (2016). Improving MQTT Data Delivery in Mobile Scenarios: Results from a Realistic Testbed. *Mobile Information Systems*, *2016*, 1-11. doi:10.1155/2016/4015625
35. Manohar, H. L., & Reuban Gnana Asir, T. (2017). Data Consumption Pattern of MQTT Protocol for IoT Applications. *Communications in Computer and Information Science*, 12-22. doi:10.1007/978-981-10-7635-0_2
36. MQTT Format, MQTT Tutorial, MQTT Protocol Format, MQTT Packet Format. (2019, April 2). Retrieved from https://www.bevywise.com/developing-mqtt-clients/
37. MQTT. (2011, August 9). Retrieved from https://en.wikipedia.org/wiki/MQTT
38. MQTT. (April 2). Retrieved from http://mqtt.org/
39. MQTT-SN Specification by IBM. (n.d.). Retrieved from www.mqtt.org/MQTTs_Specification_V1.0.pdf
40. Mrs.Kiran Patil, Prof. Dr. S.L. Lahudka2. (2016). A Survey of MAC Layer Issues and Application layer Protocols for Machine-to-Machine Communications. *IRJET*, *3*(2), 742-747.
41. Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Systems Engineering Symposium (ISSE)*. doi:10.1109/syseng.2017.8088251
42. Novo, O. (2018). Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Internet of Things Journal*, *5*(2), 1184-1195. doi:10.1109/jiot.2018.2812239
43. Palattella, M. R., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L. A., Boggia, G., & Dohler, M. (2013). Standardized Protocol Stack for the Internet of (Important) Things. *IEEE Communications Surveys & Tutorials*, *15*(3), 1389-1406. doi:10.1109/surv.2012.111412.00158
44. Premsankar, G., Di Francesco, M., & Taleb, T. (2018). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275-1284. doi:10.1109/jiot.2018.2805263
45. Rayes, A., & Salam, S. (2016). IoT Protocol Stack: A Layered View. *Internet of Things From Hype to Reality*, 93-138. doi:10.1007/978-3-319-44860-2_5
46. Rayes, A., & Salam, S. (2016). IoT Protocol Stack: A Layered View. *Internet of Things From Hype to Reality*, 93-138. doi:10.1007/978-3-319-44860-2_5

47. RESTful Design for Internet of Things Systems. (n.d.). Retrieved from https://tools.ietf.org/id/draft-keranen-t2trg-rest-iot-05.html
48. Salman, T., & Jain, R. (2016). NETWORKING PROTOCOLS AND STANDARDS FOR INTERNET OF THINGS. *Internet of Things and Data Analytics Handbook*, 215-238. doi:10.1002/9781119173601.ch13
49. Serpanos, D., & Wolf, M. (2017). *Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies*. Basingstoke, England: Springer.
50. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, *3*(5), 637-646. doi:10.1109/jiot.2016.2579198
51. Siegel, J. E., Kumar, S., & Sarma, S. E. (2018). The Future Internet of Things: Secure, Efficient, and Model-Based. *IEEE Internet of Things Journal*, *5*(4), 2386-2398. doi:10.1109/jiot.2017.2755620
52. Singh, M., Rajan, M., Shivraj, V., & Balamuralidhar, P. (2015). Secure MQTT for Internet of Things (IoT). *2015 Fifth International Conference on Communication Systems and Network Technologies*. doi:10.1109/csnt.2015.16
53. SMCP PROTOCOL. (n.d.). Retrieved from https://github.com/darconeous/old-coap-archived/tree/smcp-0.6
54. Specification, S. (n.d.). STOMP. Retrieved from https://stomp.github.io/
55. Stankovic, J. A. (2014). Research Directions for the Internet of Things. *IEEE Internet of Things Journal*, *1*(1), 3-9. doi:10.1109/jiot.2014.2312291
56. Stevens, W. R. (n.d.). *TCP/IP Illustrated Vol. I: The Protocols*. Delhi, India: Pearson Education India.
57. Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, Jesus Alonso-Zarate. (2015). A Survey on Application Layer Protocols for the Internet of Things. *Transaction on IoT and Cloud Computing*, 1-8.
58. What is XMPP Protocol in IoT | XMPP Server | XMPP Client. (n.d.). Retrieved from http://www.rfwireless-world.com/IoT/XMPP-protocol.html
59. XMPP. (2018, November 2). Retrieved from https://en.wikipedia.org/wiki/XMPP
60. Yassein, M. B., Shatnawi, M. Q., & Al-zoubi, D. (2016). Application layer protocols for the Internet of Things: A survey. *2016 International Conference on Engineering & MIS (ICEMIS)*. doi:10.1109/icemis.2016.7745303
61. 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. doi:10.1109/jiot.2014.2306328