International Journal of Pure and Applied Mathematics
Volume 115 No. 6 2017, 37-47
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

PERFORMANCE ANALYSIS OF DATA PROTOCOLS OF INTERNET OF THINGS: A QUALITATIVE REVIEW

Anusha.M[1], Suresh Babu.E[2], Sai Mahesh Reddy.L[3], Vamsi Krishna.A[4], Bhagyasree.B[5]
[1,3,4,5] Department of Electronics and Communication Engineering, K L University, A.P, India.
[1]kris.ambati143@gmail.com
[2]Department of Computer Science and Engineering, K L University, A.P, India.

Abstract: Recently, IoT emerged as the most popular advancement of Internet and became a trending technology that incorporates M2M communication. This communication makes use of Internet protocols and various devices such as smart sensors, actuators, LTE, WLAN etc. These devices are self-configurable that collaborate with each other for exchanging the data without the inclusion of human beings. The custom of IoT through M2M communication provides unique characteristics that expand to various applications such as smart home, smart supply chain, wearable's, smart retail, military, smart farming, smart city, industrial internet, connected car, smart grids and connected health etc., which makes human lives simpler. Like network architecture, the IoT architecture also comprises three tired architecture-application tier, network tier and context-aware tier. This paper review only the application layer protocols of network tier of IoT. While the application tier, other layer of network tier and context-aware tier are out of the scope of this paper. Specifically, this paper reviews MQTT, MQTT-SN, AMQP, CoAP, XMPP, and DDS data protocols of IoT and compared these protocols with the challenging issues such as security, caching, resource discovery, support to QoS etc. Finally, we analysed the performance of these protocols with various metrics such as network packet loss rate, message size, bandwidth consumption and latency.

Keywords: Internet of Things (IoT), M2M, Data protocols, XMPP, MQTT, CoAP.

1. INTRODUCTION:

From the last three decades, there is a tremendous development and usage that had taken place on the internet for effective communication. Today, this communication progressed to connect numerous smart devices to the Internet, characterized as Internet of Things (IoT), which is a most popular and trending technology that incorporate Machine to Machine communication(M2M). This M2M communication devices includes embedded sensors, RFID, Wi-Fi, data networks, actuators, LTE, WLAN etc. These devices process itself and exchange the data without the inclusion of human beings that empowered the physical world into a computerized network for greater accuracy and efficiency. Moreover, IoT provides more attractive characteristics such as correspondence, union,unification, Green living, Preventative maintenance, temperature control, dynamic nature, connectivity, enormous scale, heterogeneity, sensing, energy and safety etc., which attract various applications-smart home, smart supply chain, wearable's, smart retail, military, smart farming, smart city, industrial internet, connected car, smart grids and connected health etc., that makes human lives simpler. However, this technology possesses various challenges such as sensing, connectivity, power, security and makes use of cloud services as shown in Table 1.

Like network architectures, Internet of things (IoT) also comprises three-tiered architecture-Application tier, Network tier and Context-aware tier. The application tier contains applications, which includes environment monitor, medical applications authentication, service management, information management, technical management; Intelligent computer technology-SOA, Platform Enhanced Technology. Cloud services. While Network tier is the backbone of IoT technology that consists stack of protocols-application protocols (MQTT, CoAP, XMPP, AMQP, DDS), transport protocols (TCP/UDP), network protocols (RPL, CORPL, IPv6 and 6LoWPAN) and data link protocols (WLAN technologies). Finally, Context-aware tier consists of various sensors devices sensors, actuators, RFID etc. that collects the data. Moreover, context-aware tier

Table 1. Challenging Issues of Data Protocols

Protocol	Support to QoS	Security	Bandwidth needed	Caching	Resource discovery	Type of service
XMPP	No	Yes(SSL)	Low	Yes	Yes	TCP
CoAP	Yes	Yes (DTLS)	Low	Yes	Yes	UDP
AMQP	Yes	Yes (SSL)	High	Yes	No	TCP
MQTT	Yes	Yes (SSL)	Low	Yes	No	TCP
DDS	Yes	Yes (SSL, DTLS)	Low	Yes	Yes	UDP
MQTT-SN	Yes	Yes (SSL)	Low	Yes	No	TCP

connects to the network tier through gateways to provide the better service, which is depicted in Fig.1

This paper reviews the various data protocols- MQTT, MQTT-SN, CoAP, XMPP, DDS, AMQP etc. that comprises in network tier of IoT a broker to establish the connections of higher bandwidth. But this protocol performs better in higher bandwidths which is the limitation of this protocol. Next, the CoAP (Constrained Application Protocol) is a UDP-based protocol that supports both one to one and many to many communications. This protocol is mainly used for lightweight applications such as smart city development, smart grid and building automation, group communications and transport logistics. While XMPP (Extensible Messaging and Presence Protocol) is a standardized one for instant messaging services that is established through TCP. Finally, DDS (Data Distribution Service) known for the machine to machine communication, designed by Object Management Group, which is a UDP based connection between the publisher and subscriber.

2. Related Work

This section presents related work of application layer protocols of IoT for qualitative analyses proposed by various researchers.

In Muneer Bani Yassein et.al. has surveyed on application layer protocols in IoT because for any application to connect protocols are the key factor and also provides many services for message transmission [5]. This survey provides the reliable protocol for certain applications. Further, he discussed communication model, security, and quality of service of each protocol and provided information about how to choose an application layer protocol for any application.

Mohamed H. Elgazzar has proposed a complete analysis of the various protocols used for M2M communication and Device Controlling. He discussed the pros and cons of each of the protocols and recognized their open issues [1]. With this information, we managed to differ various protocols concerning their application. He compared protocols based on network overhead, supported functions, network reliability and security while emphasizing the protocol architectures.

Vasileios Karagiannis has proposed “A Survey on Application Layer Protocols for the Internet of Things”. He mainly discussed various data protocols like CoAP, AMQP, MQTT, HTML; used to connect multiple devices without the involvement of human beings and noticed their reliability, security, and energy consumption by comparing them [11] . He showed an underlying IoT architecture in which various application protocols are compared to demonstrate their use in the future. He proved CoAP is best among them as battery consumption is not taken into consideration.

Sven Bendel has proposed “A Service Infrastructure for the Internet of Things based on XMPP”; which mainly focuses on the integration of real world objects in IoT using extensible messaging and presence protocol(XMPP) [12]. In addition to that, it tells about how XMPP is helpful in remote robot control and service improvement in the e-mobility domain. This protocol is useful for minute message footprint and low message transfer. It provides highly scalable and most capable communication platform among the building blocks.

Yuang Chen has proposed “Performance Evaluation of IoT Protocols under a Constrained Wireless Access Network”; in thispaper, he quantitatively compared performance of various protocols such as MQTT, AMQP, DDS, CoAP and a custom UDPprotocol. By comparing he anticipated DDS is better for medical purposes for its performance with regard to latency and reliability [3]

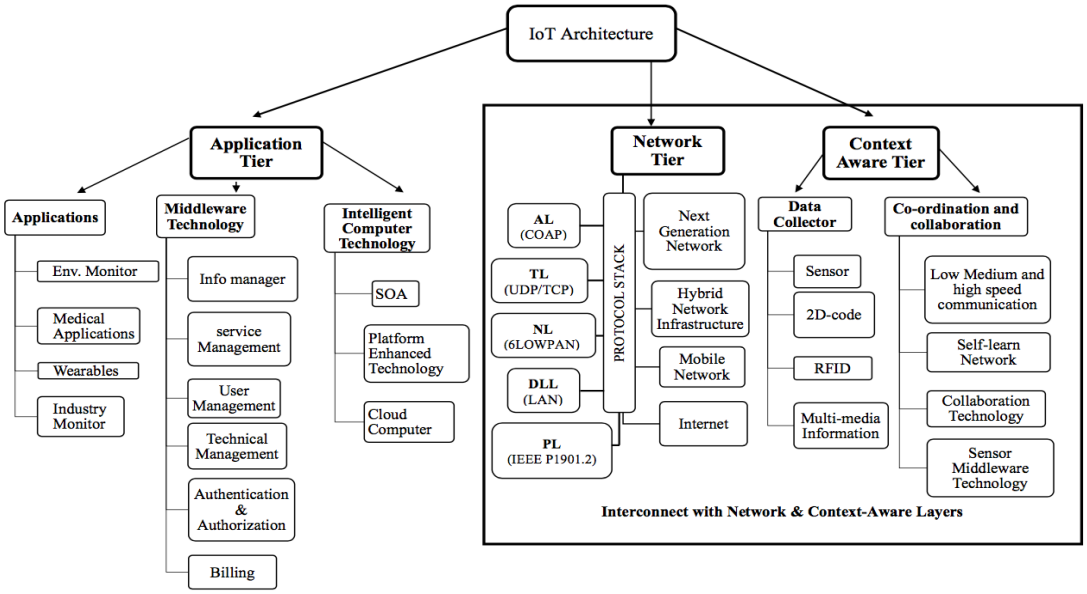

Figure1. Architectural diagram of IoT

3. Data Protocols in IOT

This section presents MQTT, MQTT-SN, CoAP, XMPP, DDS and AMQP data protocols, which is the part of IoT network tier. The main purpose of this survey is to know the functionality of each data protocol, comparing with the other data protocols based on various performance metrics- Latency, bandwidth consumption, loss rate of the packet and workload. The comparison helps us to identify which data protocol is more appropriate and suitable for numerous IoT applications that are not properly addressed in the literature. Particularly, we studied each data protocol in detail with pros, cons and application used for better communication to transmit the data from one application to another application. Finally, the summary of all these protocols are depicted in the table 2.

3.1 Extensible Messaging and Presence Protocol (XMPP)

The XMPP is one of the popular protocol known as "Jabber", which is an open source for instant messaging that connects different people using text messaging. Specifically, this protocol is composed of XML service that make use of TCP protocol for reliable communication. Moreover, XMPP had built-in security and adapts the current and future applications, which lacks in other core protocols [13]. The built-in security feature of XMPP make use of both SASL and TLS mechanism forproviding the data integrity, authentication and secure communications. The data security and authentication can be achieved using Simple Authentication and Security Layer (SASL) technique. To securely communicate between application to application the Transport Layer Security (TLS) mechanism is used.

Based on various features such as Instant Messaging and security authentication the XMPP architecture is shown in Fig.2. The components available in XMPP architecture are mainly servers and clients. Routing capability is provided by servers for data transfer from one unique client to another unique client and also for foreign domains via gateways [13]. XMPP gateway permits the expiry of client-to-server session and the beginning of a new session to the target endpoint protocol. Thus, XMPP is an ideal support protocol to offer universal connectivity between different protocols. By using XML stanzas,XMPP connects the client to server. An XML stanza denotes a section of code which is divided into three modules such as message, presence, and Iq (information/query) as shown in Fig. 3. A message stanza consists of message title and its contents. Presence stanza shows customer status and status updates for the authorized members. Moreover, Iq stanza combines message senders and receivers.

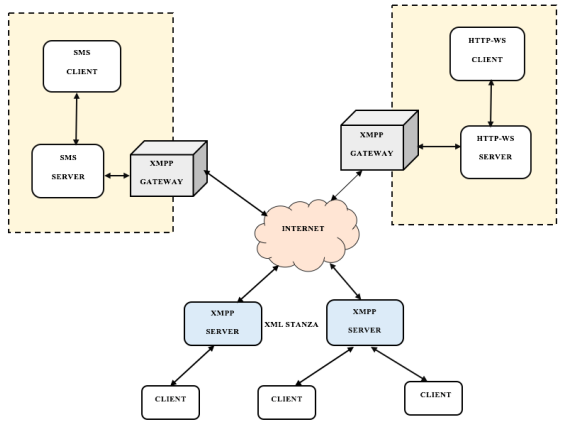

Figure 2. XMPP architecture

```
<stream>
<presence>
    <show/>
</Presence>
<message to = 'x'>
    <body/>
</message>
<iq to = "y">
    <query/>
</iq>
</stream>
```

Figure 3. Structure of XMPP stanza

Pros:

1. Channel encryption: Channel encryption helps in building secure applications which provide connection between client and server.
2. Authentication: The communication over the network will be authenticated first by the server which also helps in building secure applications.
3. Presence: With the support of this service we can know the availability of clients and servers where they were online or offline in the network and the presence information sharing based on presence subscription.
4. One-to-one messaging: By this service we can enable to send messages to an alternative entity, any two entities on a network can transfer messages they might be servers, XMPP-enabled web services, devices, or any supplementary XMPP entity.
5. Notifications: By this function, we can generate notifications instantly which will have used in our applications

Cons:

1. XMPP is only suitable for short messages
2. Quality of Service(QoS) was not supported by XMPP

Applications:

1. Group chat
2. Gaming
3. System control
4. Voice over IP (VoIP)

3.2 Message Queueing Telemetry Transport (MQTT)

MQTT is another data protocol that supports the M2M communication, which is yet to be standardized at OASIS. This protocol is a lightweight publish/subscribe messaging transport that is used to connect various remote locations with low space and minimal network bandwidth. Moreover, the MQTT will give flexibility to add the security feature for applications to encrypt the data that he sends and receives, in order to keep the protocol lightweight and simple. However, the MQTT does add significant network overhead, while addressing the security issue using secure socket layer (SSL), which is not a light weighted one [14]. MQTT mainly addresses three properties of Quality of Service (QoS) very effectively while message transport between the applications- QoS 0- is the first level of QoS with the term "At most once delivery", which is the least level. However, this level adds the significance to the application, where the user can send the message in the fastest way using MQTT protocol without waiting for the responses. QoS 1 – is the next level with the term "At least Once Delivery" in which the client or server should send at least one message irrespective of duplicate messages. QoS 2 – is the last one with highest level of QoS, known as "Exactly once delivery" in which messages are transferred only once without allowing duplicate messages. Further, MQTT protocol provides best communication in mobile applications because of its small size, efficient distribution of information to one or many receivers, minimized data packets and low power usage.

MQTT is mainly designed with three components namely server, broker and client as shown in the Fig. 4. It uses publish/subscribe mechanism to transmit data between devices for effective communication. Depending on the required functionalities client shifts between publisher and subscriber roles. A server can generate essential content and publishes

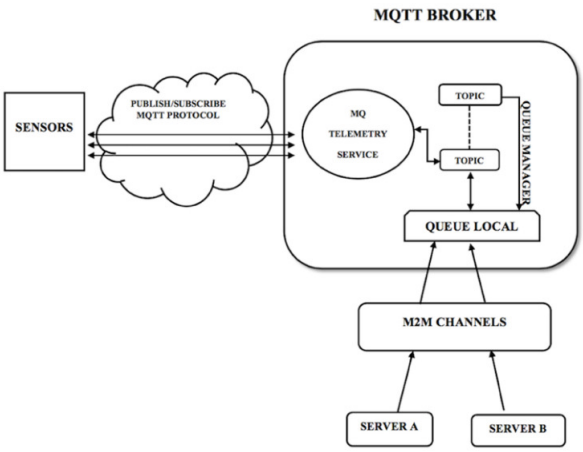

Figure 4. MQTT Architecture.

information to required client (subscriber) through the broker. Broker handles the circulation of information by producing limited security between clients and server. Moreover, the broker sent the information from its queue manager in the form of topics. Each topic is specified for a subscriber based on the publisher generation and subscriber can be connected by registering itself to the broker.

Pros:

1. MQTT is considered as a lightweight messaging protocol because all messages have small code footprintThis protocol is a bandwidth-efficient protocol that was data agnostic with support for multiple levels of QoS.
2. It also provides two-way communication over unreliable networks.
3. MQTT has few methods (publish/subscribe/unsubscribe), quick to learn.
4. The smallest packet of size 2 bytes is possible for an MQTT message
5. This protocol distributes from one-to-none, one-to-one, one-to-N via the publish/subscribe mechanism.

Cons:

1. MQTT Version 3.x only supports the publish/subscribe.
2. MQTT has no advanced features such as flow control.
3. As all the message payloads are binary MQTT protocol lacks interoperability
4. Problems will arise in open networks because there will be no information about how they are encoded.

Applications:

1. Home automation: Gardening, lighting control, power monitoring, energy monitoring with the old style analog ammeter.
2. Constrained networks: Medical applications, Smart home.
3. Mobile software: Facebook Messenger
4. Enterprise level applications.

3.3 Advanced Messaging Queueing Protocol (AMQP)

AMQP is also an application layer protocol that uses the message-oriented middleware with significant features- routing, security, message orientation, reliability and queuing.Like MQTT, this protocol also uses publish/subscribe mechanism for transmission of data to provide reliable communication and guarantees message delivery [7]. In additional, AMQP supports various features-heterogeneity, interoperability, reliability and security. Particularly, AMQP also maintains the broker to provide better reliability through queues, it reduces overhead at client side using TCP connection and it provides authentication or encryption through TLS/SASL protocols for AMQP like MQTT.

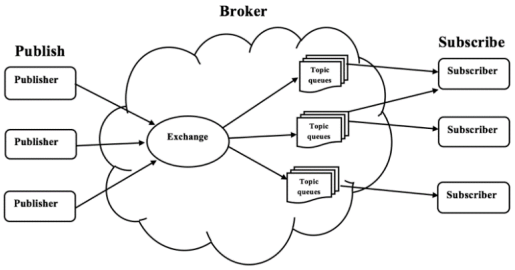

Figure 5. AMQP architecture

The Fig. 5 shows the AMQP architecture that contains the three components- a publisher, a broker and a receiver. The communication between the publisher and subscriber will be exchanges with the help of broker. The publisher is efficient of producing and forwarding messages to broker. The Broker provides two services-one is exchanges and other is to maintain the queues. Exchanges are used to forward messages to specific queues by following pre-defined rules. Whereas, Topic queues can store messages and transmit them to receiver. Receiver will store the messages.

Pros:

1. Store-and-forward feature in AMQP ensures reliability even after network disruptions.
2. This AMQP protocol is an open standard and interoperable messaging protocol.

3. AMQP provides reliable Quality of Service like at-most-once, at-least-once, exactly once.
4. AMQP is a secured protocol that is handled by SASL/TLS (authentication and security layer) in application layer.

Cons:

1. AMQP is not reliable for lower bandwidths but can improve reliability with increase in bandwidth.
2. This protocol is not constrained and light-weighted protocol.
3. It does not support an automation discovery mechanism.

Applications:

1. It is used by many famous banks like JP Morgan and The Deutsche Borse for heavier data transmissions.
2. AMQP is used in UIDAI, government of India for collection and maintaining data of 1.2 billion people.
3. It is used in cloud computing services of NASA for nebula and RED HAT LINUX for their internal communications.
4. National science foundation is using AMQP for transmitting data from ships to off shore

3.4 Data distribution service (DDS)

This data protocol was designed by object management group (OMG), which is mainly used for M2M communication in IoT that runs over UDP protocol. In contrast to other protocols, it uses limited publish/subscribe mechanism. But it provides concurrent, scalable data exchange between the publisher and subscriber. In additional, it can transfer thousands of messages per unit time to several receivers very proficiently without loss of information. DDS provides exceptional Quality of Service and reliability by supporting 23 quality of service levels. Unlike other protocols, Data distribution service (DDS) has a broker-less architecture as shown in Fig. 6 That contains the data object system between the subscribers and publisher. These publishers and subscribers were connected over a network and topics are related to data writers and data readers. When the publisher transmits data, then where the data is passed on data writer. Whereas, data readers are capable of reading and storing data from different users.

Pros:

1. It needs the less bandwidth, and it also reduces the complexity of the network.
2. Provides reliability and flexibility.
3. It is used to provide the interoperability between the users.
4. Resource discovery, caching are the major advantages.
5. It provides security using both SSL and DTLS.

Cons:

1. It does not provide any scalability.
2. It does not have broker facility.

Applications:

1. Hospital Integration
2. Medical Imaging
3. Military Systems
4. Wind Farms

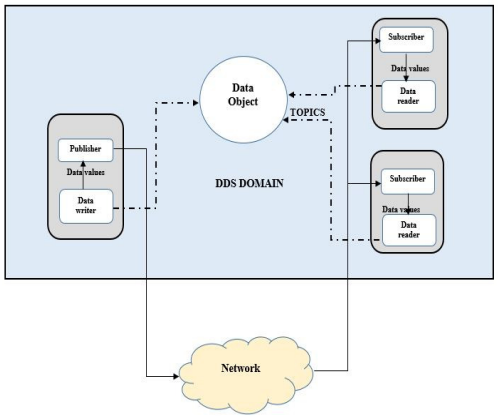

Figure 6. Architectural model of DDS.

3.5 Message Queueing Telemetry Transport-Sensor Network (MQTT-SN):

The MQTT-SN is an extension of publish/subscribe communication protocol, mainly designed for the low bandwidth range, low power devices- ZigBee or Tiny OS. ZigBee is an IEEE standard for wireless personal area networks, which provides security and interoperability of various products. However, MQTT- SN make use certain features of MQTT that allows communication between WSN with available infrastructure. Moreover, it can communicate with multiple clients that are connected over a wireless sensor network. Particularly, each client is connected to their gateways with MQTT-SN protocol and gateways connects the traditional MQTT broker for transmission of data.

Pros:

1. MQTT-SN is a many-to-many communication protocol.
2. MQTT-SN's credibility was supported by IBM, Eurotech, Cisco and Red Hat.

3. It can be used for wireless sensor networks.
4. This protocols another advantage is that it is an open source.

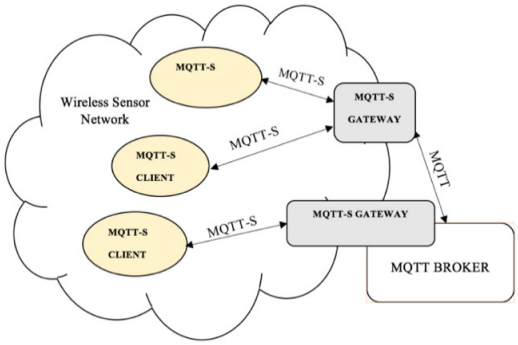

Figure 7. MQTT-SN architecture

Cons:

1. MQTT-SN lacks support for labelling messages which make it difficult to understand.
2. It has to be familiar with the message formats to enable communication.

Applications:

1. Mainly designed for the use in enterprise applications over low bandwidth wide area network links such as ISDN or GSM

3.6 Constrained Application Protocol (CoAP)

CoAP is a lightweight protocol that provides a communication channel and runs over UDP protocol with request/response message. This is one of the standard protocols for the interaction between various physical devices. To achieve the data transmission, CoAP keeps the message size as small as possible and supports stop-wait retransmission mechanism. For transmission of messages using CoAP, clients are directly connected to a server or client will connected to a proxy, which is linked to servers via HTTP.

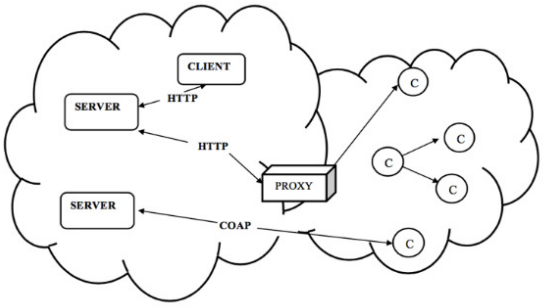

Figure 8. CoAP Architecture.

CoAP has four types of messages– confirmable, non-confirmable, acknowledgment and reset. Moreover, it supports M2M requirements, Datagram Transport Layer Security (DTLS), asynchronous message exchanging and Resource discovery, unicast and multicast communication. Like HTTP, CoAP has client/ server model with two layers. The bottom layer is the request/response layer and the upper layer is the application layer [5]. The Application layer handles different clients and external applications.While request/response layer is comprised of different interconnected servers.

Pros:

1. It supports multicast communication i.e. both one-one and many to many communications.
2. CoAP provides asynchronous communication and offers several security features like integrity, confidentiality, and authentication.
3. It gives datagram transport layer security over UDP format.
4. CoAP provides simple proxy and caching capabilities.

Cons:
It is less standard than the MQTT.
Applications:
1. Smart city development.
2. Smart grid and building automation
3. Group communications
4. Transport logistics

4. Performance Evaluation and Analysis

This section presents, the performance evaluation and analysis of various data protocols like XMPP, MQTT, AMQP, DDS, CoAP based on theoretical values and values obtained from contiki software. The graphical representation is drawn based on various metrics-network packet loss rate, message size, bandwidth consumption and latency.

The Fig. 9 shows the latency with varying bandwidth of different data protocols. It is observed from the figure that CoAP Protocol has lower latency compared to other data protocols. Also, DDS has constant telemetry latency varying with network bandwidth. Unlikely, MQTT, AMQP is showing decreasing latency with an increase in network bandwidth. While, XMPP latency increases up to a certain point and decreases with increase in network bandwidth.

The Fig. 10 shows the varying message size with number of produced messages per second for all the data protocols [4]. It is observed that MQTT provides better workload compared to other AMQP and CoAP protocol with fixed header size of 2

bytes. Whereas AMQP uses is 8 bytes and CoAP has 4-byte header. XMPP produces better reception than all others due to its XML stanza based transmitting, and light-weight carrying of messages.

The Fig. 11 shows the relation between network packet losses with bandwidth consumption of all data protocols. It is observed that CoAP maintains the consistent bandwidth throughout, because there is no re-transmission were involved. While DDS consumes enormous amounts of bandwidth compared to other data protocols. The XMPP, AMQP and MQTT are TCP based protocols shows an increased bandwidth consumption with increased packet loss rate due to their retransmission mechanism. However, it is observed that from the same Fig. 11, all the three data protocols, bandwidth consumption reduces with increased packet loss rate.

The Fig. 12 shows the network packet loss rate with actual telemetry loss rate. We had tested all the data protocols with the consistent packet loss from 0% to 25%. it is observed that the TCP-based XMPP, AMQP and MQTT protocols has diverse characteristics compared to UDP-based CoAP, DDS protocols. However, The CoAP and XMPP has very similar packet loss on produced network packet loss rate. On the other hand, MQTT, AMQP, and DDS provide no packet loss due to topic queues.

Table 2. Summary of all the Data protocols in IoT

Protocol	**Characteristics**	**Working**	**Advantages**	**Disadvantages**	**Applications**
MQTT	Low power usage, M-M communication	Pub-Sub based protocol, main aim to collect data and transport to IT infrastructure	Save power and memory, Low power usage	Long-lived TCP connection, topic names are long strings	Home automation, Enterprise level applications
XMPP	Channel encryption and presence checking	Allows internet users to send instant messages	Secure, Service discovery, Very Robust, powerful	Data flow is more than XMPP server, lack world wide support	Instant Messaging, Group chat, Gaming, Vehicle Tracking
AMQP	Message queueing and interoperable.	Designed to support wide variety of messaging and communication patterns	Highly reliable, Store & forward communication	Works at higher bandwidths only	Business Messaging, and in Banking Industry
DDS	Interoperable, data service with high performance	To connect one device to other device and also to share right data at the right place	Interoperable, saves bandwidth, flexible and reliable	Have no scalability	Medical Imaging, Military Systems, Hospital Integration and Wind Farms.
CoAP	Synchronous request response, 1-1 or M-M communication	Used in simple electronic devices that permits them to communicate interactively over a network.	1-1 Communication, M-M communication, Resource discovery	Less standard, not more mature and standard compared to MQTT.	Smart homes, smart grid and Building automations
MQTT-SN	Light weight and Publish subscribe messaging protocol	Has been adapted for better function of devices where low power device usage is a primary concern.	Open source, many-to-many Communication protocol	Lacks support in Labelling messages which makes it difficult	Enterprise applications

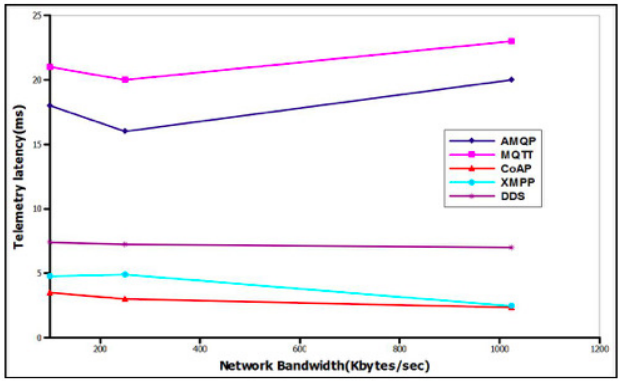

Figure 9 : Telemetry latency vs. Network Bandwidth.

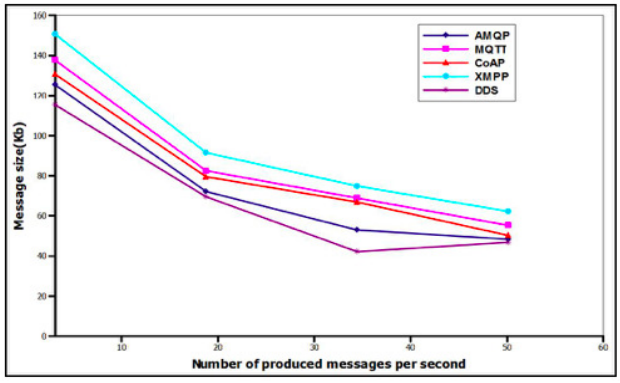

Figure 10: Message size vs. Number of produced messages per second.

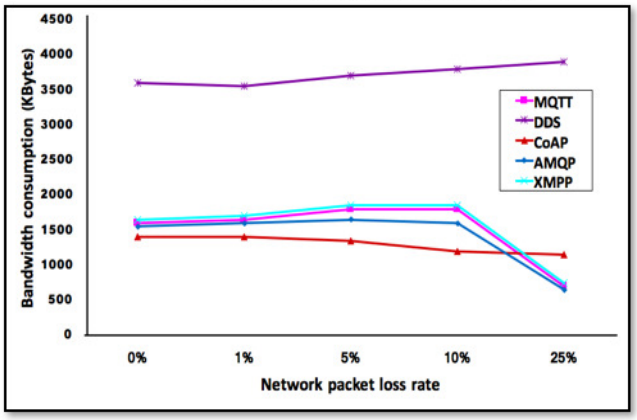

Figure 11: Bandwidth consumption vs. Network packet loss rate.

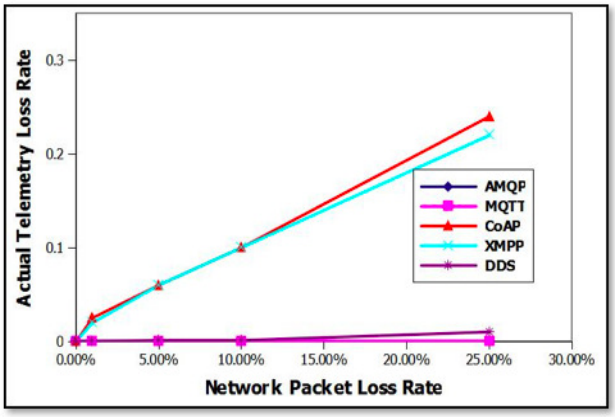

Figure 12: Actual Telemetry Loss rate vs. Network packet loss rate.

5. Conclusion

This paper review the MQTT,MQTT-SN, AMQP, CoAP, XMPP, and DDS data protocols of IoT. We compared these protocols with the challenging issues such as security, caching, resource discovery, support to QoS etc. Finally, we analysed the performance of these protocols with various metrics such as network packet loss rate, message size, bandwidth consumption and latency. We observed that after analysis of each protocol is better on its way depends upon its applications. However, for internet-based applications such as instant messaging, Systems controlling and Voice over IP (VoIP) for M2M communication, XMPP provides a better result, due to its XML stanza based transmitting, and light weight carrying of messages with minimal latency.

Further, it is recommendable to determine further evaluations of performance metrics and appropriate qualitative interpretations for additional M2M protocols that can be applied in IoT

References

[1] Mohamed H. Elgazzar. "Perspectives on M2Mprotocols" in 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15).

[2] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, and Moussa Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications" in IEEE communication surveys & tutorials, vol. 17, no. 4, fourth quarter 2015.

[3] Yuang Chen, Thomas Kunz "Performance Evaluation of IoT Protocols under Constrained Wireless Access Network" in 2016 International Conference on Selected Topics in Mobile & Wireless Networking.

[4] Jorge E. Luzuriaga, Miguel Perez, Pablo Boronat, Juan Carlos Cano, Carlos Calafate, Pietro Manzoni "A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks" in 2015 IEEE 12th Consumer Communications and Networking Conference (CCNC): CCNC 2015 Workshops – VENITS.

[5] Muneer Bani Yassein, Mohammed Q. Shatnawi, Dua' Al-zoubi. "Application Layer Protocols for the

Internet of Things: A survey".

[6] Wentao Shang, Yingdi Yu, Ralph Droms. "Challenges in IoT Networking via TCP/IP Architecture" in NDN Technical Report NDN-0038, 2016

[7] Xuandong Xiong, Jiandan Fu. "Active Status Certificate Publish and Subscribe Based on AMQP" in 2011 International Conference on Computational and Information Sciences IEEE.

[8] Md. Mahmud Hossain, Maziar Fotouhi, and Ragib Hasan. "Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things" in 2015 IEEE World Congress on Services.

[9] Joel L. Fernandes, Ivo C. Lopes, Joel J. P. C. Rodrigues, and Sana Ullah. "Performance Evaluation of RESTful Web Services and AMQP Protocol".

[10] Urs Hunkeler & Hong Linh Truong, Andy Stanford-Clark. "MQTT-S – A Publish/Subscribe Protocol for Wireless Sensor Networks".

[11] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, Jesus Alonso-Zarate. "A Survey on Application Layer Protocols for the Internet of Things" in Transaction on IoT and Cloud Computing 2015.

[12] Sven Bendel, Thomas Springer, Daniel Schuster, Alexander Schill. "A Service Infrastructure for the Internet of Things based on XMPP" in Work in Progress session at PerCom 2013, San Diego (19 March 2013).

[13] www.xmpp.org

[14] www.mqtt.org

[15] S. Ezhilvanji, S. Malarkodi, "An Efficient Water Distribution System for India using IoT", International Innovative Research Journal of Engineering and Technology, 2017.