IoT Architectures and Platforms

[1]Abhishek Pathak, [2] C. Kalaiarasan
[1]Department of Computer Engineering, St. Vincent Pallotti College of Engineering & Technology, Nagpur
[2]School of Engineering, Presidency University, Bengaluru
[1]abhishekpathak81@rediffmail.com, [2]kalaiarasan@presidencyuniversity.in

Abstract

The Internet of Things (IoT) is the newest evolutionary field in both current and future times. It is a concept of several, different gadgets that are linked together via wired and wireless technology and are available on any system, at any time. IoT is projected to bridge and synchronise the capabilities of new technologies with those of existing devices in the upcoming years as it emerges as a major technology. To create intelligent systems, IoT gadgets rely on large-scale processing power. By giving less complex network-connected objects semantic power, the Internet of Things' primary goal is to automate systems. We put a lot of attention on providing schematic details of IoT design approaches in this chapter, along with protocol specifications and design based on various types of infrastructure and capabilities. Applications, quality-of-services (QoS), parameters for effective communication and management, the impact of scalability, device interoperability, issues with device interoperability with the environment, various IoT standards or protocols, the IoT working environment, and related issues with understanding the standards are discussed along with the core functionalities of the publish and subscribe model and the Request/Response model.

Keyword: Publish/Subscribe, AMQP, MQTT, Interoperability, COAP & IoT protocols.

I. Introduction

The need for intelligent and user-focused Internet of Things (IoT) applications is driving up the availability of better and more advanced techniques for automating and changing the global environment. The paradigm for physical devices connected to networks for communication is changing thanks to the Internet of Things. Numerous applications use diverse heterogeneous devices that adhere to various standards. As a result, there are several obstacles to enhancing the quality of services (QOS), including those related to security, privacy, energy efficiency, scalability, interoperability, and protocol standardisation. A vast number of devices will contribute to the development of an automated and intelligent environment in the future years [2].

Additionally, improvements in IoT will improve the complex problems and services to applications such as transportation, e-health, home and industry automation, and agriculture, among others [3]. IOT is made up of numerous heterogeneous device combinations that communicate by sending data. Data interoperability is a barrier in data extraction since the data is generated from different types of devices in different ways. The most promising approach to overcoming issue is to incorporate semantics, as explained in [4]. [10][1] describes a process for semantic analysis and extraction.

Secure authentication is offered for various non-intelligent and intelligent devices or nodes in an IOT network that is categorised level by level. Many research projects have been developed and are now being worked on to address major issues facing various organisations and sectors working in IOT. Numerous researchers are helping to address important scalability and interoperability concerns. The design techniques were discussed, along with a brief overview of each methodology. The main objectives of architecture are outlined. The construction blocks and an explanation of each level are given in section III. The aspects taken into consideration when creating the architecture are explained in Section IV. Section V highlights several protocols based on the publish/subscriber and request/response categories and specifies the message format in detail.

Nomenclature	
IoT	Internet of Things
CoAP	Constrained Application Protocol
AMQP	Advanced Message Queuing Protocol
MQTT	Message Queue Telemetry Transport
DDS	Data Distribution Service
XMPP	Extensible Messaging and Presence Protocol

II. IoT Design Methodologies

Almost all of the world's application sectors have been impacted by IoT. Healthcare, transportation, agriculture, smart homes, and industrial applications are some of the major boosted areas where a significant amount of funding is going.

The IoT design and implementation techniques needed to implement the applications vary depending on the applications. The key approaches should, in a broader sense, define the specifications indicated in Fig. 1 and in accordance with the explanation provided in [22].

- Descriptions based on Objectives and purpose
- Descriptions based on Actions required
- Descriptions based on Modelling
- Descriptions based on Information
- Descriptions based on Functions
- Descriptions based on Networking and services

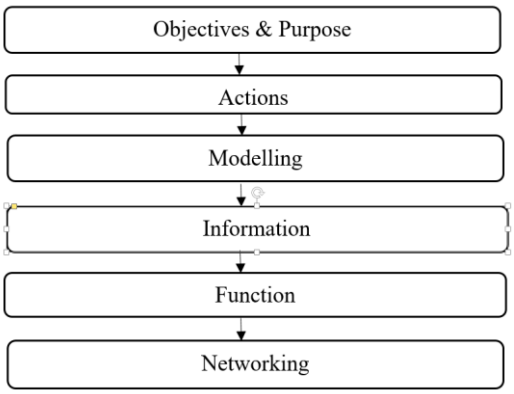

Fig. 1 Methodologies adopted in IoT

Description based on objectives: This is the first stage in learning about the system's execution behaviour, or the aspects that should be taken into account when collecting data, retrieving information, and using the system.

Description based on actions: This step specifies the manner of flow of actions, which are derived from the nature of system. The behavioural aspects of system which is specific to the application is identified.

Description based on modelling: In this step, the attributes related to the system devices are identified, the relational effect of devices on each other is identified. The concepts involved in between the devices are considered, this helps in predicting the behavioural aspects involved in the system.

Description based on information: We identifies the structure of information, the different entities involved in the system uses different structure on different levels which requires different parameters to be considered. This assist in receiving the details of the information in system.

Description based on function: This level defines the specifications required for the functional overview of IoT structure. The functions depends on device, services, management, and security.

Description based on networking: This step specifies the networking and interconnection parameters on device and component level. Based on an application the different devices and components utilized and will communicate with their specific tools and platforms. So deciding the factors supports in resolving the issues.

III. Architecture

IoT is a developing field that provides assistance and services to all technical and scientific fields. All system components in M2M, industries, and other organisations create their functional architectures depending on the fundamental needs present. An architecture must be created to comprehend how the various devices and components are connected in order to administer and support these systems. The system should be able to link a variety of nodes and gadgets. Future corporate, agricultural, and industrial growth and development will all depend on the Internet of Things (IoT), which will make it easier to provide services for all items used in daily life. In order to meet anticipated infrastructure needs for flexible system design. scalability, and interoperability, hence we need robust and flexible infrastructure.

3.1 The key goals of Architecture

In this section, we discuss the key goals for designing the architecture of IoT.

3.1.1 Ready to use

We must recognise the IoT's usages, applications, and technologies in order to fulfil its purpose and goals. With an expanding number of devices, future applications and usage will necessitate updates and improvements at all IoT levels. This results in an infrastructure that is ready for use, and both software and hardware should be able to be updated in response to changes in an application's requirements.

3.1.2 Modelling and Functionality

It is crucial to understand which device is sending the data from which sort of application because many devices are involved in com munication and data gathering. It's likely that the same type of devices will be linked to a network, making it necessary to distinguish between them in order to provide that application with specific functionality and services. As a result, each device is given a unique d evice identification.

3.1.3 Privacy and Security

As increasingly heterogeneous devices connect with one another, interoperability has become a significant problem in the IoT space, posing concerns to data privacy. Identification, Authentication, Confidentiality, Integrity, and Non-Repudiation capabilities should be provided by infrastructure.

3.1.4 Localization

The network's features of availability, dependability, and mobility should be supported by the infrastructure as well. Communication between mobile and mobile-to-mobile or mobile and non-mobile types of devices is possible if the node is mobile in nature. As a result, the configuration constantly varies depending on the location, making device/node availability the key determinant of communication. Infrastructure needs to be able to provide the functionalities when the location changes.

3.2 Building block of IoT Architecture

Various architectures are proposed according to the technologies. As per the services and management, the IoT architecture must consists the basic three layers Physical Layer, Networking layer, and Management layer as shown in Fig. 2.

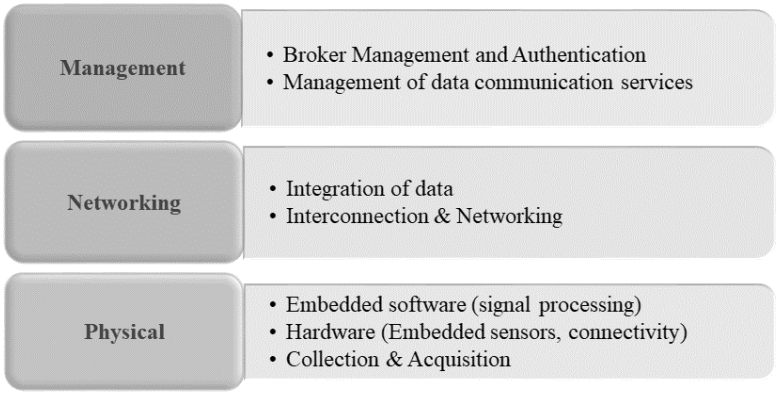

Fig. 2 IoT Architecture

3.2.1 Physical layer

The physical layer, where physical is defined as "devices," establishes the norms and method at the fundamental level of any interconnection in accordance with international networking standards. When a signal is sensed from physical objects, or in signal form, and then turned into digital signals, the Internet of Things (IoT) uses sensors and smart devices or hardware. Additionally, developers integrate their platforms into the hardware in accordance with the applications.
Example: Sensing devices like pulse sensors and heartbeat sensors are used in healthcare applications to detect the pulse and heartbeats, respectively. The signals are then translated to digital values and the values are aggregated in accordance with the needs of the application..

3.2.2 Networking

The most crucial component of the IoT infrastructure is networking as a process medium. It involves the wired and wireless routing topologies and techniques used in the Internet of Things. Basic structure is needed at this layer to connect end-to-end communication, and it varies depending on the kind. These relationships could be of the same, various, or hybrid forms. The network pays attention to the hybrid interconnection because it is more susceptible to modifications in the future. Some applications have technology that are built specifically for them, making them resistant to modify. The technology, scalability, and complexity of networking are the three main factors to take into account. The technological revolution could lead to modifications to the standards governing wired and wireless topologies, methodologies, and horizontal growth

3.3.3 Management

The most important process in the IoT ecosystem is data management. Managing the various forms of data created by the numerous devices used in the Internet of Things is a crucial responsibility. Data management is required to process this massive volume of data that is constantly being pulled or transmitted. The main difficulties in managing data are monitoring of data, services for service provision, processing, connectivity, etc. The handling of data involves various technological ideas such as;

i. Event processing
ii. Data collection and analytics
iii. Semantics related to networking.

Event processing: One of the new research fields in the intricate field of event processing is event processing in nodes, particularly in virtual sensors. The ability to forecast future behaviour is dependent on the understanding that situations and actions have provided. It may be based on computation or detection.

Data collection and analysis: One of the IoT's most revolutionary applications is data collection and gathering, where data is gathered and analysed to obtain precise information. It facilitates data interchange among many systems linked via a network.

Semantics related to networking: Today, practically all data is collected and accessed on clouds. To identify the features in the programme, this data is expanding. Because of the increased need for data-related applications, a mechanism is needed to identify the type of data depending on the complexity degree of applications. In order to facilitate the network's scalability and interoperability, data is interpreted and annotated.

IV. Elements of Internet of Things

As Internet of Things is a paradigm of heterogeneous element, so based on the functionality the elements of IoT differs. This sections specifies the important elements to be considered for functionality based on services.

i. Sensing and Identification
ii. Computation and Services
iii. Communication

Sensing and Identification: The most important process in IoT is to gather the data coming from various physical objects connected in the network. These devices or nodes sends the data through various links. The devices are assigned with specific device id’s, the purpose of id’s are for identification in a network. The sensing and identification also represented in Fig. 3.

Example: Electronic product codec (EPC) and Ubiquitous are methods available for this purpose. Sensing and identification is the basic step in the world of IoT, which is important for further steps of communication and computation.

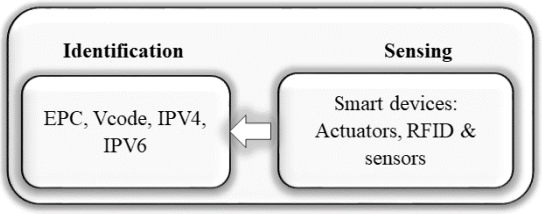

Fig. 3 Sensing and Identification

Computation and Services: IoT analysis involves removing valuable information from the data so that it can be processed further. Through embedded sensors, the data batches are sensed and gathered for analysis. These data are diverse and diversified. Understanding the pattern and its structure with the aid of the analysis helps to reduce maintenance, prevent failures, and enhance operations. The protocols in computation validate the data's data structures and make them available for processing. On the IoT's hardware and software levels, processing is carried out. Recent technologies like fog and cloud computing are applied at the software level. Devices connected to the Internet of Things (IoT) include those with terminology like actuators incorporated in them. These gadgets produce massive amounts of big data chunks, necessitating complicated calculations in order to extract knowledge. [8] provides examples on numerous IoT platforms for using hardware like the Arduino platform series, the Rasberry Pi, as well as more contemporary and specialised application platforms like Intel Galileo, BeagleBone, Gadgeteer, WiSense, Cubieboard, and T-Mote Sky. Researchers that have contributed to this topic have also highlighted the use cases for computing and smart device applications. These intelligently functioning devices run a variety of operating systems and applications, including TinyOS, RiotOS, LiteOS, Contiki Android, C, C++, and JAVA, among others. IOT is now able to use powerful computing capabilities because to recent developments in the fields of cloud and Edge computing. A cloud platform offers the ability to handle and analyse real-time data for intelligent analysis while storing data in the cloud.

Communication: In the Internet of Things, communication takes place between devices using M2M, D2S, or S2S protocols (S2S). The behaviour and operating system of these strategies vary. Since the goal of the Internet of Things is to connect everything that isn't intelligent and can't execute tasks. In such a setting, communication protocols and standards enable the interpretation and comprehension of data from alien objects. The complexity of interacting with the other end device is facilitated and made simpler. Information on various communication standards is displayed in Table 1.

Table1 Communication Standards at Different Levels[5]

Level	Communication Standard
Infrastructural level	Wireless Fidelity (WiFi) Bluetooth IEEE 802.15.4 Z-Wave Long Term Evolution(LTE)-Advanced Near Field Communication (NFC) Radio Frequency Identification(RFID)
Management Level	Advanced Message Queuing Protocol(AMQP) Constrained Application Protocol (COAP) Message Queuing Telemetry Transport (MQTT) Data Distribution Service (DDS) Extensible Messaging and Presence Protocol (XMPP)

Infrastructural Level: Range and power considerations help to construct the physical connections. As the primary factor determining the operating life of devices in a network, power consumption and device depletion. Examples of such communication protocols are IEEE 802.15.4, Z-Wave, and LTE-Advanced, among others. M2M communication was made necessary by some technologies, including RFID, NFC, and UWB. RFID uses radio links to assist communication; these radio links can identify an object within a range of 10 cm to 200 cm and can communicate at a frequency of 13.56 MHZ. The NFC protocol likewise uses the same high frequency of 13.56 MHZ for communication and enables a range of up to 10 cm.The most promising technology is WiFi which arises as a basic support for IoT infrastructure for communication and services within 100m range[6].

Management Level: Data management entails checking data for mistakes and exceptions, as described in Section 3.3. Routing logic is required to convey pulled data from the publisher to the subscriber. To manage the channel of communication between publishers and subscribers, various management protocols are implemented; this management is carried out at the application layer. The IOT ecosystem supports a number of management protocols, including AMQP, DDS, COAP, MQTT, and others.

V. Standardisation & Protocols

Standardization is a collaborative effort between the private sector, academic institutions, government agencies, and consumers that establishes a common set of guidelines for testing processes on certain platforms. In order to achieve quality of service (QoS) in the operation of devices and platforms, it is important to define the standard. IoT devices need a common platform to interpret communication because they are a collection of heterogeneous devices[5][7][9].
Standardisation is essential to set;

- Interoperability across applications and services
- Maintain operation across system, syntax, semantics, and domain knowledge
- Maintain regulations in economy between regulators, and developers
- Provide security and privacy of contents

Today, preserving performance interoperability, availability, and reliability is the main problem for standardisation. Numerous organisations are moving in this direction and making progress. The most influential groups include ETSI (European Telecommunications Standards Institute), IEEE (Institute of Electrical and Electronics Engineers), W3C (World Wide Web Consortium), IETF (Internet Engineering Task Force), and EPCglobal[23].

In IOT, protocols are the fundamental building elements of the communication process. The IOT services are made possible via a variety of protocols.On each layer of interconnection the specific protocols functions for task as explained in the Table 2.

Table 2 Protocols and Standards

Layers	Protocols and standards
Application layer	ONEM2M/ETSI, HTTP, COAP, SEP 2.0
Transport layer	TCP,UDP , ZigBee
Network layer	RPL, IPV6, 6LowPan
Data link layer	IEEE 802.15.4, Bluetooth, wifi, NFC, 3GPP
Physical layer	

The protocols in IOT can be categorized based on the functionality assigned and level of management. Protocols in IoT can be categorized according to application level, infrastructural level, and influential level. As shown in Fig. 4, the protocols at influential level are protocols used with wide scope of usability at system level.

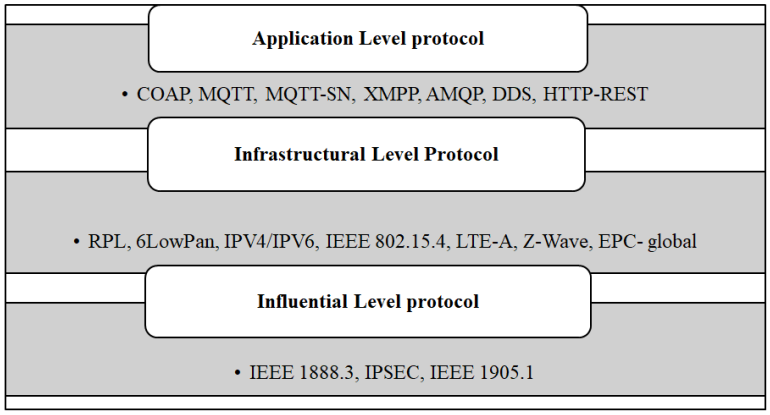

Fig. 4 IoT Protocols

5.1 IEEE 1905.1

Protocol defines the standard for both wired and wireless media, specially used as WiFi media in market. It supports connectivity to the heterogeneous types of devices with mobility benefits and used as an intermediate between network and data link layer. IEEE 1905.1 message frame consists of 8 octets and variable length list. The frame slots are as follows and the frame format is shown in Fig. 5.

- Message version - 1 octet
- Reserved - 1 octet
- Message type - 2 octet
- Message-Id - 2 octet
- Fragment-Id - 1 octet
- Last fragment indicator - 1 octet
- Relay indicator - 1 octet
- Reserved – 6 bits
- List of type length value (TLV)

Message version	Reserved	Message Type	Message Id	Fragment Id	Last Fragment Indicator	Relay Indicator	Reserved	Type Value Length

Fig. 5 IEEE 1905.1 Frame Format

5.2 IEEE 1888.3

Another protocol discussed in this category is IEEE 1888.3, a standard defined by ISO/IEC/IEEE. It is a protocol for security service enhancement and a standard proposed for Ubiquitous green community control network. It also help to avoid unintended data disclosure. The processing of infrastructural layer and application layer related to protocols is listed as below;

- At infrastructural level the protocols applied for synchronisation of standard. The popular protocol in this category are RPL, 6LowPan, IPV4/IPV6, IEEE 802.15.4, LTE-A, Z-Wave, EPC- global.

- Application layer protocols are responsible for data presentation and formatting. In constrained environments where heterogeneous nature of devices are connected the IEEE 1888.3 protocol supports the communication of such heterogeneous environment. Many protocols are developed by keeping the objective of constrained environment i.e. COAP, AMQP, and MQTT, etc., these are standard communication protocols. The basic functionality of these protocol is to provide services based on message transmission and supports the environment of Request/Response or Publish/Subscribe type. Maximum protocols in this category uses routing schemes such as round robin and message queuing mechanism.

5.3 Constrained Application Protocol (COAP)

Protocol designed especially for constrained application devices to meet the requirement of Internet and M2M communication, and supports multicasting. It is request/response type protocol and message length is four byte with fixed header, version type, token length, request/response code and message ID. In COAP, GET, PUT, POST, and DELETE are the operations to achieve Create, Retrieve, Update, and Delete (CRUD) operations. It works on four types of messages i.e. confirmable, non-confirmable, reset, and acknowledge[11][12][13].

The Fig. 6 explains the implementation view of COAP. The request from client is accessed by COAP server and is forwarded over the HTTP-REST to process the query on Internet storage. In Addition, Representational State Transfer Protocol (REST) is cacheable connection. The CRUD operations are indicated and type of value each operation hold is represented below[11][12][13];

- Request
  - 0: Confirmable: Acknowledgement message
  - 1: Non-confirmable: Does not expect a confirmation message.
- Response
  - 2 : Acknowledgement : Acknowledge a confirmable message
  - 3: Reset: Received a message and not processed.

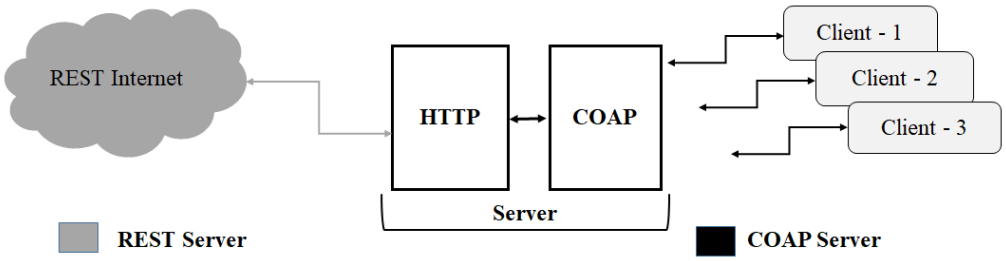

Fig. 6 Constrained Applocation Protocol

COAP is a specialized application protocol designed by RFC7252. It is designed in such a way that it easily get integerated with HTTP and UDP for seamless integeration and useful in environment where M2M is implemented. The frame format of COAP is presented in Fig. 7, COAP is low overhead protocol thus providing easy integeration and less outflows. The message formats consists of version, message ttype, Token length, rerquest/response code and message id[17].

Versio n	Typ e	Token Lengt h	Request /Response Code	Message ID

Fig. 7 COAP Frame Format

Queuing Mechanism:

The queuing mechanism protocols are more popular in use and attracted the researchers to investigate and resolve the issues related to interoperability and reliability. We enlist the protocol here and the detailed working is highlighted in Section 5.4 and 5.5 respectively.

- Message Queuing Telemetry Transport(MQTT)
- Advanced Message Queuing Protocol(AMQP)

5.4 Message Queuing Telemetry Transport protocol[

MQTT is a binary based lightweight protocol, bandwidth efficient and uses low battery consumption. It's an open source protocol designed on publish/subscribe scheme and introduced [24] and then modified to standardised by OASIS in 2013 [15]. It is built of TCP protocol. The packet length of MQTT protocol is computed using control header, length, protocol level, connect flags and payload, the same is represented by Eq. 1.

$$MQTT\ packet\ length = control\ header + length + protocol\ level + connect\ flags + payload \quad (1)$$

The protocol level defines the QoS level of delivery assurance. In this control header is fixed and is of 1 byte and packet length is of maximum 4 bytes. There are two versions of MQTT protocol; MQTT designed for TCP/IP protocol and MQTT-SN designed to work over UDP and ZigBee protocols. The architecture of MQTT is presented in Fig. 8.[26]

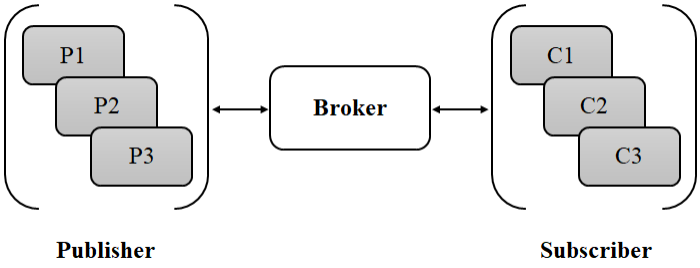

Fig. 8 MQTT Protocol

MQTT operations is categorized in two sub-operations i.e. from Publisher to broker and then from broker to consumer/client. As shown in Fig. 9, MQTT server as broker receives the messages from publishers and make them available to get subscribed to appropriate client. The publisher publishes the message and gets the response/acknowledgement to publish. The broker consists of queue, the maximum queue length in MQTT is of 260MB. The specific client access the specific message required. The client request to connect with the broker and broker provide the acknowledgment to connect[19].

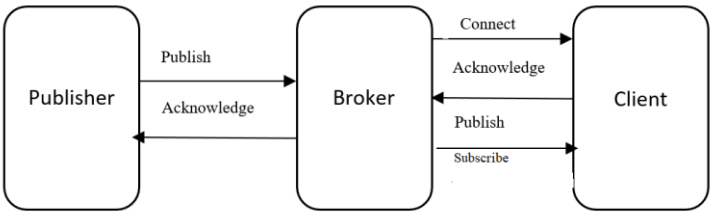

Fig. 9 MQTT operation

The Fig. 10 demonstrates the MQTT message format which is a binary based protocol. It consists of Message type of four bytes, UDP field of one byte, QoS level of two bytes and lastly one byte of retain field. The message format of MQTT consists of control header, packet length, variable length header and payload. The control element are in form of binary bytes not the text strings. It operates on request and response mechanism i.e. to each request, the acknowledgment is accepted[16].

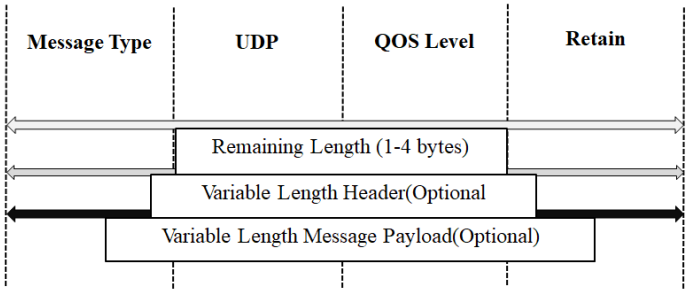

Fig 10. MQTT message format

5.5 Advanced Message Queuing Protocol (AMQP)

AMQP is an advanced message queuing protocol. As queuing mechanism targets the issues related to interoperability and reliability in IoT. In this concept multiple queues can be implemented. AMQP is layered architecture that defines[18];
Layer 1: Defines system and encoding process.
Layer 2: Defines transport layer and efficient, binary peer to peer protocol functioning.
Layer 3: Defines grouping for atomic transactions.
Layer 4: Defines security aspects
The popular AMQP versions are RabbitMQ, OpenMQ, StormMQ, ApacheQpid, and RedHat Enterprise MRG.

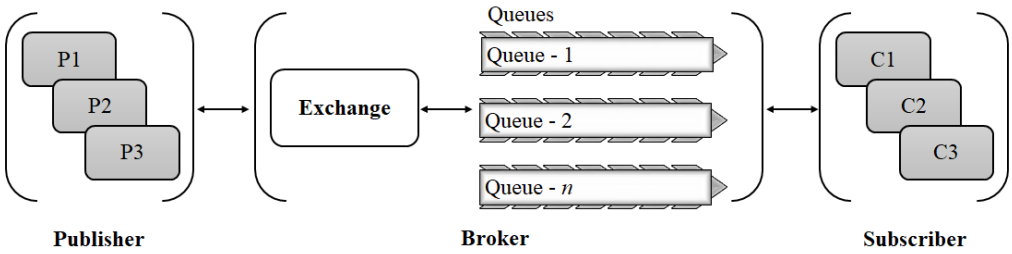

Fig. 11 AMQP Protocol

As shown in Fig. 11 the implementation of AMQP protocol consists of broker. The broker is an implementation of exchange and queues. The exchange is a software program that decides the assignment of messages to particular queue, the messages can be assigned based on Topic, Direct, Fanout and header, and the exchange implement routing algorithm. Exchange is used to bind the queues[20][25].

Direct Exchange: Direct exchange is used to deliver the messages to queues based on routing keys. It is well suited to unicast mode. Direct exchange is default exchange of AMQP protocol, and same has been shown in Fig. 12.

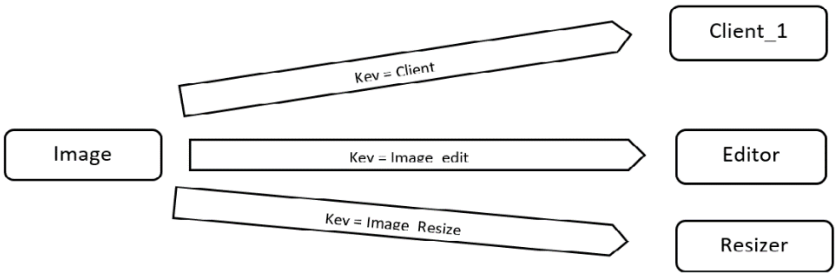

Fig. 12 Direct exchange

Fan-out Exchange: This exchange supports broadcasting type of communication. If 'N' queues are connected with the exchange, so messages will be published to all the queues. The function of Fan-out exchange is presented in Fig. 13.

- Online multiplayer gaming.
- Best suited to distributed environment concepts in which server sends configuration messages and updates to all the clients.

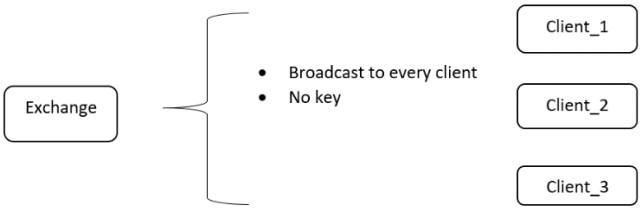

Fig. 13 Fan-out Exchange

Topic Exchange: This exchange type supports multicast. The routine is done to more queues simultaneously. The message gets delivered to queues based on matching key and pattern. Example, in Geographical cases, sending specific data to specific location user.

Header Exchange: In this type of exchange, the messages are published as per the type of headers assigned as key. The key can be multiple in number. Based on matching key with only one type or all type messages are published.

This exchange model accepts messages from the publisher and route them to queues according to the predefined criteria as shown in Fig. 11. It uses a routine and instances to examine the message and route it to proper queue by using key, which is actually a virtual address. In communication, the parameters used for message are:

- Queue current status
- Time to Live (TTL) for message expires
- Queue Length
- Message Type
- Message Identifier
- Message order

Queue status: The queues used in messaging holds the data according to the capacity of queue. The messages/data is pulled from messages as per the parameter i.e. links defined for the consumer.
Time to Live (TTL): It is a live for which the message will remain in the queue. The time to live decides the life of message in the queue.
Queue Length: The queue will decides the capacity of message to hold in communication by the queue.

Moreover, the features target in the communication are**:**

- Targeted Quality of service
- Persistence
- Delivery of messages to multiple consumers
- Possibility of ensuring multiple consumptions

RabbitMQ: It is basically a publish/subscribe model, a broker architecture. It uses queue exchange mechanism for publishing the message and subscribing the message from consumer. It is an advanced message queuing technology. The exchange implements routing logic to forward the message to the respective queue. The routing logic can be anything the user need but more preferably round robin type of routing is common for forwarding the messages. As explained by RabbitMQ pivotal, the message can be routed by using four different types of exchange mechanism i.e. Fan out, Direct, Topic, and header. The exchange mechanism using fan-out the message is forwarded to all queues i.e. messages gets consumed by all the consumers. Direct exchange mechanism binds the queues with consumer by using routing/binding keys.
The topic exchange mechanism provides same mechanism as direct only adding some wild characters to messages. The header type keys and values as header in the message as a parameter to publish and consume.

The basic mechanism on which RabbitMQ works is Queuing mechanism. The queue gets bind with exchange, exchange routes the messages produced by the producer, and consumed by the consumer from queue. These producers and consumers may be of different and varying configurations and applications, which gives rise to the issues related with Accuracy and interoperability i.e. mismatch between number of message packets generated per second by the producer (publisher) and message packets consumer by the consumer because of different configuration platforms.
Example, in a company with various departments, where all departments are connected centrally. Considering only three department's Human resource, Marketing, and Administration maintaining active directories. The same is presented in Fig. 14 and details are shown follows;

Actors: Human resource department – producer
Marketing department – consumer1
Administrative department – consumer2
RabbitMQ system – Broker (exchange & queue)

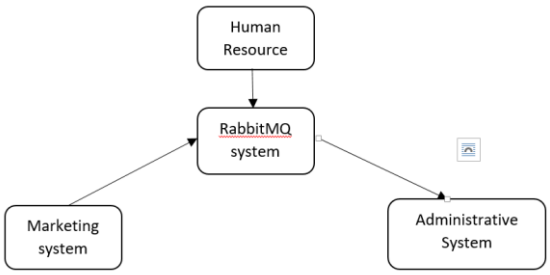

Fig. 14 Implementation example of RabbitMQ

RabbitMQ is more sophisticated implementation of message queuing techniques.it is more scalable as it provides the mechanism to increase the use of queues. The number of queues may vary as per the number of exchanges used. It also provides the mechanism named as dead letter exchange (DLX) where the rejection of messages is avoided. Still with some issues related with availability of queues may arise which results in decrease in performance related to queues ? Another more specific implementation of message queuing known as Kafka is attracting the attention of researchers. In Kafka the zookeeper concept is implemented to avoid the issues related with performance. Various research papers describes the working of Kafka with zookeeper implementation as well as examples are explained.

Now a days, researchers are focusing on issues related to communication in IoT. As heterogeneous devices are connected in IoT systems, hence we are facing continuous issues related to scalability and interoperability, so security and privacy becomes the most challenging task. MQTT protocol is light weighted protocol, consumes less power, requires less bandwidth and utilized in heterogeneous devices connected in different environment. Hence MQTT protocol is less secured because of only service of authentication process and no encryption capabilities as presented in [21] .Mechanism applied to overcome security issues can be implemented that focuses on authentication and authorization of devices at broker level. These mechanism are best suited for constrained devices but for resource constrained devices still needs further development. AMQP protocol offers more secure architecture as it provides more reliable connection oriented procedure. In addition, AMQP protocol facilitate diversity in application and use, becomes vulnerable to know threats to network. Table 3 presents protocols highlighted according to the security aspects.

Table 3. Protocol Connectivity and Security Aspects

Protocols/Connectivity	QOS	Transport	Security	Usage
AMQP	Yes	TCP	Yes	D2S/S2S
MQTT	3 level	TCP	Yes	D2S
HTTP	No	TCP	Yes	Web
DDS	20 levels	TCP/UDP	Yes	D2D
Web-socket	No	TCP	Yes	Web

Another important challenge is related to interoperability, it is becoming a key challenge in communication between D2S and S2S. To address this, researchers are providing the solution related to analysis of data based on syntax and semantics. Semantic extraction of data helps in classifying and categorizing the data, thus helping in future issues related to scalability and interoperability. In IoT, the challenge related to interoperability can be addressed in basic connectivity between devices. Semantics provides potential strength to data extraction, the tags or labels are utilized to extract and classify the data through which domain knowledge and context information can be matched for accurate data extraction.

VI. Summary and Future Direction

As predicted by various standard bodies and organisation that with the growth in number of devices and nodes in network. The challenges and threats related to the communication and synchronisation will increase. In coming years, the quality of services and performance of the system will depend on the networks support for scalability with efficient interoperability. The coming era will be independent of the platform and modelling of system will not depend on the specifications and requirements of functional modelling in the system. Various IOT platforms will get updated with the arising needs and requirements. It is clear that a single protocol and standard will not be sufficient to cope up with publish and subscriber synchronisation. The challenges will arise with quality of

services (QOS) in performance of protocols. In such varying environment the standards and protocols in IoT will play a vital role in seamless integration of devices with platforms to achieve objectives of IOT.

References:

1. ANDRIY MAZAYEV , JAIME A. MARTINS , AND NOÉLIA CORREIA. (2018). Interoperability in IoT Through the Semantic Profiling of Objects Center for Electronic, Optoelectronic and Telecommunications, University of Algarve, Faro 8005-139, Portuga.

2. Evans, D. (2011). The Internet of Things: How the Next Evolution of the Internet is Changing Everything. Cisco Internet Business Solutions Group: San Jose, CA, USA
.
3. Knaian, A. N., Paradiso, J. & Smith, A. C. (2000). A Wireless Sensor Network for Smart Roadbeds and Intelligent Transportation Systems. Mass. Internet Technol. https://dspace.mit.edu/handle/1721.1/9072 (accessed on 17 January 2018).

4. Shi, F., Li, Q., Zhu, T., Ning, H., et al. (2018). A Survey of Data Semantization in Internet of Things. Sensors, 18, 313.

5. Wazid, M., Das, A. K., Odelu, V., Neeraj Kumar, Conti, M., Jo, M., et al. (2018). Design of Secure User Authenticated Key Management Protocol for Generic IoT Networks. IEEE Internet of Things Journal, 5(1), 269-282.

6. Lin, J., Yuy, W., Zhangz, N., Yang, Z., Zhangx, H., Zhao, W., et al. (2017). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. IEEE Internet of Things Journal, 4(5), 1125-1142.

7. Yang, Y., Wu, L., Yin, G., Li, L., Zhao. H., et al. (2017). A Survey on Security and Privacy Issues in Internet-of-Things. IEEE Internet of Things Journal, 4(5), 1250-1258.

8. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M., et al. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communication Surveys and Tutorials, 17(4), 2347-2376.

9. Sethi, P. & Sarangi, S. R. (2017). Internet of Things: Architectures, Protocols, and Applications. Journal of Electrical and Computer Engineering, 2017, 25 pages.

10. Satyavrat Wagle., Semantic Data Extraction over MQTT for IoTcentric Wireless Sensor Networks 2016 International Conference on Internet of Things and Applications (IoTA) Maharashtra Institute of Technology, Pune,India 22 Jan - 24 Jan, 2016

11. Ludovici, A., Moreno, P. & Calveras, A. (2013). TinyCoAP: A Novel Constrained Application Protocol (CoAP) Implementation for Embedding RESTful Web Services in Wireless Sensor Networks Based on TinyOS. Journal of Sensors and Actuator Networks, 2, 288-315.

12. Shelby, Z., Hartke, K., Bormann, C., Frank, B., et al. (2013). Constrained Application Protocol (CoAP). Internet Engineering Task Force (IETF), Fremont, CA, USA.

13. Bormann, C., Castellani, A. P. & Shelby, Z. (2012). CoAP: An Application protocol for billions of tiny Internet nodes. IEEE Internet Computing, 16(2), 62-67.

14. Amaran, M. H., Noh, N. A. M., Rohmad, M. S., Hashim, H., et al. (2015). A Comparison of Lightweight Communication Protocols in Robotic Applications. IEEE International Symposium on Robotics and Intelligent Sensors, Procedia Computer Science, 76, 400-405.

15. Oasis. (2014). MQTT Version 3.1.1 Plus Errata 01. http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf.Cohn, R. (2012). Comparison of AMQP and MQTT. White Paper. https://lists.oasisopen.org/archives/amqp/201202/msg00086/ StormMQ_WhitePaper_- Comparison_of_ AMQP_ and_ MQTT. Pdf.

16. Clark, A. S. & Troung, H. (2013). MQTT for sensor networks (MQTT-SN) protocol specification. http://mqtt.org//new/wpcontent/uploads/2009/06/MQTT-SN/_spec_v1.2.pdf.

17. Madhumitha, P., Johnsema, B. & Manivannan, D. (2014). Domination of Constrained Application Protocol: A Requirement Approach for Optimization of Internet of Things in Wireless Sensor Networks. Indian Journal of Science and Technology, 7(3), 296-300.

18. Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., Manzoni, P., et al. (2015). A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks. IEEE 12th Consumer Communications and Networking Conference, Las Vegas, NV, USA

19. Grgić, K., Špeh, I & Heđi, I. (2016), A web-based IoT solution for monitoring data using MQTT protocol. International Conference on Smart Systems and Technologies (SST), Osijek, Croatia.

20. Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., Manzoni, P., et al. (2014). Testing AMQP Protocol on Unstable and Mobile Networks. Internet and Distributed Computing Systems. IDCS 2014. Lecture Notes in Computer Science, vol 8729. Springer, Cham.

21. Syaiful Andy1, a, Budi Rahardjo2, b, Bagus Hanindhito3, c. Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System, Proc. EECSI 2017, Yogyakarta, Indonesia, 19-21 Sept.

22. Arshdeep Bahga & Vijay Madisetti (2015), Internet of Things, a hands on approach. ISBN : 978-0996025515.

23. Arpan Pal, Hemant Kumar Rath, Samar Shailendra and Abhijan Bhattacharyya., et al. (2018). IoT Standardization: The Road Ahead.

24. Dr Andy Stanford-Clark of IBM, and Arlen Nipper of Arcom (now Eurotech), in 1999https://www.ibm.com/support/pages/mqtt

25. Gregory Marsh, Ajay P. Sampat, Sreeram Potluri, and Dhabaleswar K. Panda et.al Scaling Advanced Message Queuing Protocol (AMQP) Architecture with Broker Federation and InfiniBand.