IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-7, July 2019 E-ISSN: 2347-2693

IoT Service Search using MQTT Protocol

Nihaala Najeeb[1*], Shelbi Joseph[2]

[1,2] Department of Information Technology, School of Engineering, CUSAT, Kochi, India

Corresponding Author: nihaalanajeeb@gmail.com

Available online at: www.ijcseonline.org

DOI: https://doi.org/10.26438/ijcse/v7i7.292297 | Available online at: www.ijcseonline.org

Accepted: 12/Jul/2019, Published: 31/Jul/2019

***Abstract*—** The Internet of Things (IoT) visualize seamlessly connecting the physical objects with the Internet. This in turn combines the physical world into the digital world. There are several mechanisms available that reduces the need for external human intervention for maintaining and configuring deployed. They must be scalable and efficient, since the number of deployed objects is expected to grow exponentially in the next few years. Resource discovery is a problem of finding useful devices and their services in order to accomplish a task, and it constitutes a major challenge for IoT. In this paper a survey on various service discovery mechanisms is performed and an MQTT protocol based IoT service search has been proposed. The proposed work uses a clustering and optimal distribution mechanism on the IoT services to find the optimal distribution value and return the services. Experimental results have shown significant improvement in service discovery time.

***Keywords*—**Internet of Things, MQTT protocol, Resource discovery, Services.

I. INTRODUCTION

The Internet of Things (IoT) concept is derived from the current set of existing devices on the market that have the ability to connect to the internet. As the numbers of such devices are growing every day, its diversity also increases [11]. In late 20th century, IoT received its first attention. The term was coined in 1998 and later defined as "The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/ network and Any service" [1].

IoT brings together the ever growing devices, also called smart objects, by connecting them in an Internet-like structure, allowing them to communicate and exchange information and to form new types of interaction among things and people [2].

IoT devices are provided with a microcontroller, a radio interface for communication, sensors and/or actuators. Consumer centric applications are created with services in various domains like intelligent home control (smart home), eHealth, intelligent transportation system, environmental monitoring etc. using these trends. [3]. All these application areas are being now aggregated into the concept of Smart Cities. IoT became more popular as it is supported with an advanced microcomputer and wireless connectivity technologies [4]. The importance of connecting devices and objects is to serve as the backbone for ubiquitous computing, that enables smart environments to recognize and identify objects and retrieve information. A key technology in the realization of IoT systems is middleware, which is usually defined as a software system designed to be the intermediary between IoT devices and applications [7].

To understand the vision of Internet of Things, we need mechanisms for resource discovery and their services. Thus resource discovery is the fundamental requirement of any IoT platform [3].But the diverse nature of IoT devices, their capabilities & properties, communication technologies add to the complexity of effective understanding of the IoT platforms .The data generated by the Internet of Things are valuable and have the potential to bring up innovative and novel applications. The data streams that are coming from these devices will challenge the traditional approaches to data management and contribute to the emerging paradigm of big data. Discovering and configuring the IoT devices and the associated data streams is the most challenging task before collecting and processing data from these devices (e.g. sensors) [1].

The proposed IoT service search system make use of Message Queue Telemetry Transport called MQTT protocol for transporting IoT services between two points.MQTT has a simple architecture with Quality of Service (QoS) functionality. There is an ideal clustering method used which

predefines the number of clusters and group the IoT services according to their weights. The optimal distribution mechanism sorts the clusters in such a way that the service having minimum optimal distribution value is first set to output. An internal buffer queuing system is maintained to guarantee that the services requested reaches the client.
The paper is organized as follows: section II explains a review on various existing mechanisms for service discovery. The next Section III describes the proposed work. Section IV shows experimental evaluation. Finally Section V concludes with future scope.

II. RELATED WORK

This section mentions some of the existing systems for service discovery in IoT and the technologies used in them.
The authors of [1] described about a tool called SmartLink which is used to discover and configure sensor devices and their services. Despite of the heterogeneity of sensors, SmartLink is capable of discovering sensors positioned in a particular location. It has got a Context-Aware Dynamic Discovery of Things (CADDOT) model which consist of eight phases "detect, extract, identify, find, retrieve, register, reason, and configure".

Another is a scalable and self-configuring P2P based architecture which provides automated service discovery mechanism and requires no human intervention for configuring it [2].A term service discovery (SD) is defined by the constrained application protocol (CoAP) as the procedure that has been used by a client to learn about the endpoints exposed by a server. A client discovers a service by learning a Uniform Resource Identifier (URI) that reference to a resource that is in the namespace. The discovery of the resources offered by a CoAP endpoint is resource discovery. A Resource Directory (RD) is a network element containing the description of resources that are held on other servers which allows lookups to be performed for those resources. An appealing fact about the P2P network is that as the number of participating nodes increases, the overall system capacity increases as well.

Soumya Kanti Datta et al proposed a search engine based resource discovery framework [3].It has a proxy layer that includes the drivers for protocols and communication technologies. A central registry stores the configurations of the resources and index them based on the configuration parameters. A "lifetime" attribute which denotes the period through which a resource is discoverable is introduced. The resulting resources of a discovery request are ranked by the search engine and it returns an URI to directly access each resource. The functionalities of the framework are accessed by the consumers using RESTful web services. The framework allows discovery of both smart and legacy devices.

Sabriansyah Rizqika Akbar et al proposed pervasive device and service discovery protocol at the application level by creating a protocol in the smart sensor device and the smart sensor gateway [4]. By implementing the protocol, the sensor network gateway was able to find each of the sensor network device and service descriptions and request the on- demand service to the smart sensor device. The protocol is implemented in two Arduino Uno integrated with XBee transceiver as the smart sensor device and the raspberry pi as the smart sensor gateway.

Pablo Calcina Ccori et al described various device discovery strategies [5].Therefore the focus is on the topology of the network of devices: centralized, decentralized and hierarchical. In centralized networks there are two agent types: one representing the central node that acts as a directory and a second representing all the other agents that represent the devices connected to the central node. In decentralized networks we have only one uniform agent type. Finally, in hierarchical networks there are two agent types: super nodes and standard nodes, which are located randomly in the network. Based on the evaluated criteria's, hierarchical topology has featured comparatively good balance when compared with centralized and decentralized ones.

The author [6] came up with a semantic based IoT device discovery and recommendation mechanism. Service Oriented Architecture (SOA) is applied. Broker based architecture is used to discover and recommend smart devices. Here web service discovery is defined as matching the available services with the customer's requirement service. A web service clustering based approach is used for the recommendation system.

A context-aware semantics based service discovery mechanism was proposed by Juan Li et al [7]."LOCA" – a decentralized LOcation-preserving Context-Aware that effectively locate services on the context requirements. A fully distributed peer to peer architecture is used. It also supports content and path locality.

Taking inspiration from the widely studied bio-inspired Response Threshold Model (RTM), a decentralized service discovery and selection model is proposed [8]. They take advantage of the coordination mechanisms of biological societies. RTM performs emergent selection of atomic services.

Takashi Ikebe et al. brought up the importance of discovering live data [10].A distributed search architecture has been proposed. Live data here are typically sensor data or statuses produced by resources and are dynamic. The live data is pushed to nearest live data buffer. A user sends query condition to the buffer through resource resolver and the query translator creates a query filter.

For the global resource discovery of devices and sensors across several scenarios, a homogeneous and suitable mechanism has been proposed [12]. A framework called"digcovery" is defined for discovery purpose. They maximize efficiency and sustainability of deployments. Allow users to register/include their own sensors into a common infrastructure, and access/discover the available resources and services through a mobile phone. This shows how to use the smartphone capabilities, in terms of identification, geo-location and context-awareness, to access and interact with the available resources that are surrounding us. The work is based on context awareness and geo location.

Zhiyuan Li et al propose a novel resource discovery mechanism in a 3-D Cartesian coordinate system[13].It enhances the search efficiency over the Social IoT.The mechanism is based on both of preference and movement pattern similarity to achieve higher search efficiency and to reduce the system overheads of SIoT. First, the preferences of the nodes are extracted from their profile table and resources, and also movement pattern from their trajectories using the AGglomerative NESting (AGNES) clustering method. Then, the cosine similarity of the preferences and movement pattern of the nodes is generated for building sub community in a 3–D Cartesian coordinate system to improve the efficiency of resource discovery. By utilizing the similarity found among sub communities, virtual global communities are formed to improve the searching performance for the resources. Finally, a resource discovery algorithm is designed that can dynamically adjust the search radius to balance the performance and communication overhead.

Farzad Khodadadi et al. proposed a framework called Simurgh to define, discover and compose "things" and their corresponding services [14].Web API notation and API definition languages are used. They have a two phase discovery approach to find entities having certain properties, while having services that match a specific pattern of keywords and input types.

Table 1. IoT service discovery mechanisms

S.No	Mechanism	Description
1.	SmartLink tool	Discover and configure sensor devices and their services in a particular location.
2.	P2P based architecture	Automated service discovery with no human intervention.
3.	Search engine based	Resources of a discovery request are ranked by the search engine.
4.	Pervasive device and service discovery protocol	Sensor network gateway find each of the sensor network device and service descriptions and request the on- demand service to the smart sensor device.
5.	Device discovery strategies	Central, hierarchical and decentralized.
6.	Semantic based Service Oriented architecture	Broker based architecture is used to discover and recommend smart devices.
7.	Context-aware semantics based	"LOCA" – a decentralized LOcation-preserving Context-Aware that effectively locate services on the context requirements.
8.	Bio-inspired Response Threshold Model	A decentralized service discovery and selection model.
9.	Distributed search architecture	Discovering live data.
10.	A framework called "digcovery"	For the global resource discovery of devices and sensors across several scenarios.
11.	Resource discovery mechanism in a 3-D Cartesian coordinate system	Based on both of preference and movement pattern similarity.
12.	A framework called Simurgh	To define, discover and compose "things" and their corresponding services.

III. METHODOLOGY

A distributed semantic based IoT service search system is proposed. It relies mainly on MQ Telemetry Transport (MQTT) protocol. The search system also embeds ideal clustering and optimal service distribution mechanisms in order to find the optimal distribution value for request and response. (Fig.1).

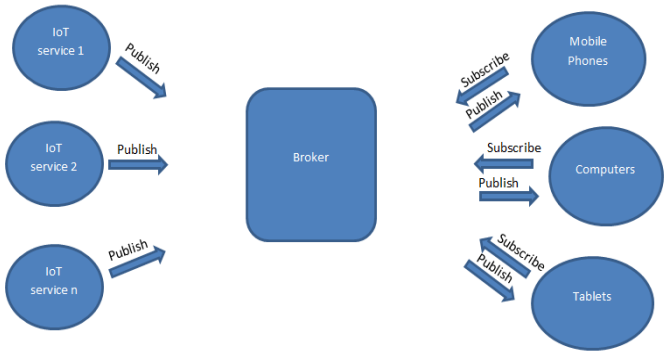

Figure 1. System Architecture

When a client(subscriber) sends an IoT service request(topics) to the broker,it informs the publisher about the topics subscribed.The publisher now encrypts the message requested using diffe-Hellman key exchange and responds with the encrypted message to the broker.The broker performs ideal clustering based on the message received and responded timing.Then optimal service distribution mechanism is carried out which sorts the clusters

making it clusters with optimal number of services and finds the optimal distribution value accordingly.The service request also undergoes clustering and optimisation.So there is a mapping between the service request and the corresponding service response.The encrypted published message enter into an internal buffer queue system.The published service response is received by the client(subscriber) and it decrypts the response with the shared secret key.

A. MQTT Protocol

A machine-to-machine lightweight protocol called MQTT protocol is used. It is a publish/subscribe messaging protocol having a client/server model. Here IoT services are the clients and they connect to a server, called a broker. Every message is a discrete chunk of data that is opaque to the broker. Every message is published to an address, known as a topic. Clients may subscribe to multiple topics. Every client subscribed to a topic receives every message published to that topic.

MQTT permits the definition of quality of service (QoS).Three levels of QoS exists in MQTT:
QoS 1: "Atmost once" delivery. Messages are not acknowledged which makes this a "fire and forget" approach.
QoS 2: "At least once" delivery. A subscriber may get the message more than once, but the receiver is allowed to acknowledge receipt.
QoS 3:"Only once" delivery. It involves four-stage handshake for delivery.

B. Ideal clustering Technique

At the broker, ideal clustering groups service requests and service responses by calculating their weights from the request time and publish time. After finding out the weights, they are clustered according to a minimum cluster value that each cluster has (Here the number of clusters are predefined).
The ideal clustering algorithm here is as follows:

Step 1: Create two array list Request [] and Response [] containing service request and service response respectively.
Step 2: Define number of clusters (Here it's four).
Step 3: Define d_{min}= {0.0d, 0.25d, 0.75d, 1.0d}.
Step 4: Find the weights for each request and response in the array list by calculating the average of the cost.
Step 5: Compare the weights with the minimum cluster value for each cluster.
Step 6: Add the corresponding request and response to the respective cluster.
The pseudo code for ideal clustering is:

```
//requestGroups.entrySet().stream().flatMap(rq-
>rq.getValue().stream()).forEach(rq->{
        double prev=0.0d;
        for(int i=0;i<dMin.length;i++) {
if((rq.getCost()>1.0d?1.0d/rq.getCost():rq.getCost())>prev&
&(rq.getCost()>1.0d?1.0d/rq.getCost():rq.getCost())<=dMin[
i]) {
            clusters.get(i).getRequestResponse().add(rq);
          }
          prev = dMin[i];
        }
      });//
```

Table 2. Vertical clusters (4x4)

	C1	C2	C3	C4
0.0	Request	Response	Request	Response
0.25	Response	Request	Response	Request
0.75	Request	Response	Request	Response
1.0	Response	Request	Response	Request

C. Optimal Service Distribution

Optimal service distribution finds the optimal distribution value for each service request and service response. First ideal clustering is performed. Then we find the optimal number of services and optimal average number of services to be arranged in a cluster.

Then optimal services distribution is calculated as follows:
First we define the terms,

Ns-The number of services distributed over a group of clusters K.
n_c-Number of clusters
The average number of services per cluster,

$$m_s = N_s/n_c$$

The average number of visited cluster per request, β is 1 when a service is matching the request and 0 when there is no service matching the request.

Next the average matching cost per request R is,

$$M_r = n_c + \beta \times m_s$$

The minimal average matching cost per request,

$$M_{rmin} = 2 \times \sqrt{\beta} \times Ns$$

The optimal number of clusters,

$$n_{cop} \approx \sqrt{(\beta i \times Ns)}$$

The optimal average number of services per cluster,

$$m_{sop} \approx \sqrt{(Ns/\beta)}$$

Finally the optimal average number of services per cluster is calculated and the optimal distribution value can be obtained for each service in the cluster.

D. Internal buffer queueing

An internal buffer queuing is performed to queue up the services requested by a client. There are three internal buffer queues namely low buffer queue, medium buffer queue and high buffer queue. The broker queues the encrypted message (service) into the low buffer queue. Then as several services gets queued up, the one in the low buffer queue is dequeued and enqueued into the medium buffer queue and then to the high buffer queue. The process continues repeatedly. Finally the requested services available in high buffer queue are dequeued and send to the subscriber. Then from the following medium and low buffer queues are dequeued.

IV. EXPERIMENTAL EVALUATION

A simulation of the broker system proposed is done in Java web application. Both the ideal clustering and cluster optimization mechanism was performed. Thus we can evaluate the proposed work according to the results obtained.

The input dataset involves services such as the following:

Table 3. Input Dataset

Topic Name	Filename	Type
Photos	image1	.jpeg
Photos	image2	.png
Songs	audioclip1	.mp3
Songs	audioclip2	.wav
Movie	videoclip1	.mp4
Movie	videoclip2	.avi
Project	doc1	.pdf
Project	doc2	.ppt
Project	doc3	.docx

On applying the clustering and optimization of clusters, the results show the optimal service distribution value.

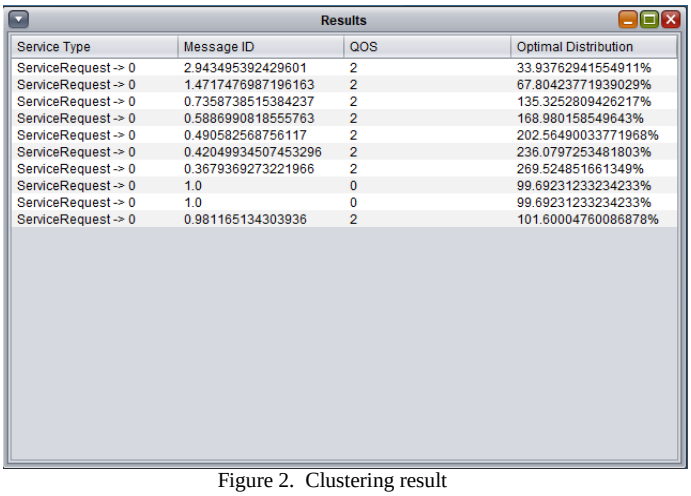

Figure 2. Clustering result

Here the service type indicates both the service request and service response. The message ID indicates it's a request or response. The QoS type indicates how the services are received by the subscriber. The percentage of optimal distribution shows the value. It also indicates the services that are delivered first to the client.

The jitter analysis graph shows the publishreceive and request time. The request time is always small and is considered zero. The publishreceive time is the differences between the requests receive time and the responses publish time. Each points indicates the publish to receive time for each service/message requested. The values grow for each request-response service.

The inputs to jitter analysis graph are: request time, publish time and receive time. From that we calculate the publishreceive time.

Table 4. Jitter graph dataset

Publish Time(ms)	Receive Time(ms)	Publishreceive Time(ms)
1000	1500	500
1200	1650	450
1350	1900	550
2000	2400	400

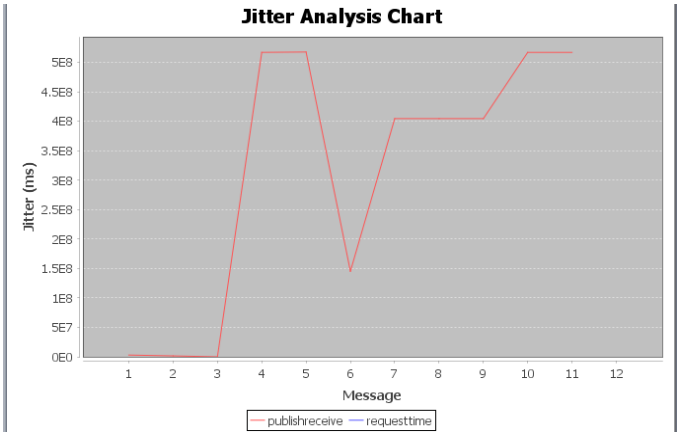

Figure 3. Jitter analysis graph

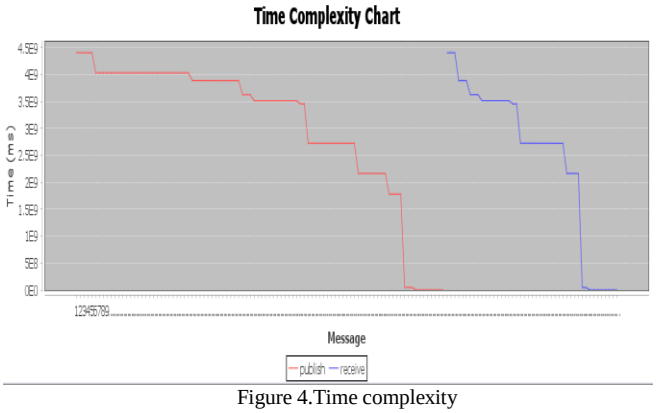

Figure 4.Time complexity

The time complexity graph depicts the publish and receive time. The difference between the current time and response time gives the publish time and the difference between the current time and request time gives the receive time.

V. CONCLUSION AND FUTURE SCOPE

The way that people interact with the world has changed. Senses have been extended with the new generation of technologies, devices and networks. The Internet of Things makes the transition from interconnecting computers to interconnecting things, describing a connected world of smart objects. The increasing number and diversity of the devices requires considerable efforts to make the communication possible. Because of this, one of the biggest challenges in the IoT era is discovering and establishing interactions with nearby objects and services. Therefore resource discovery is of very importance which includes both device discovery and service discovery. Service discovery can be performed using several mechanisms. Different mechanisms and protocols used for service discovery have been mentioned. A new semantic based IoT service discovery system that uses a MQTT protocol has been proposed. The MQTT protocol has by default QoS levels which makes it efficient to use with IoT.In addition to that there is an internal buffer queue maintained to make sure the services requested are never lost The result and the jitter analysis graph was analyzed and found better performance. For future work, we can extend the system to support semantic reasoning which results in more precise matching of request with response.

REFERENCES

[1] Charith Perera, Prem Prakash Jayaraman, Arkady Zaslavsky, Dimitrios Georgakopoulos, Peter Christen, "Sensor Discovery and Configuration Framework for the Internet of Things Paradigm", 2014 IEEE World Forum on Internet of Things (WF-IoT).

[2] Simone Cirani, Luca Davoli, Gianluigi Ferrari, Rémy Léone, Paolo Medagliani, Marco Picone, and Luca Veltri, "A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things", IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 5, OCTOBER 2014.

[3] Soumya Kanti Datta, Rui Pedro Ferreira Da Costa, Christian Bonnet," Resource Discovery in Internet of Things: Current Trends and Future Standardization Aspects", IEEE 2015.

[4] Sabriansyah Rizqika Akbar, Wijaya Kurniawan, Mochammad Hannats Hanafi Ichsan, Issa Arwani, Maystya Tri Handono ,"Pervasive Device and Service Discovery Protocol In XBeeSensor Network",IEEE 2016.

[5] Pablo Calcina Ccori. Laisa Caroline Costa De Biase Marcelo Knorich Zuffo Fl´avio Soares Corrˆea da Silva,"Device discovery strategies for the IoT", 2016 IEEE International Symposium on Consumer Electronics.

[6] Stefana Chirila, Camelia Lemnaru and Mihaela Dinsoreanu,"Semantic-based IoT device discovery and recommendation mechanism", 2016 IEEE.

[7] Juan Li,Nazia Zaman, Honghui Li,"A decentralized Locality-preserving Context-aware Service Discovery Framework for the Internet of Things",2015 IEEE International Conference on Services Computing.

[8] Elli Rapti, Catherine Houstis, Elias Houstis, Antony Kargeorgos.,"A Bio-inspired Service Discovery and Selection Approach for IoT applications, 2016 IEEE International Conference on Services Computing.

[9] Anne H.Ngu, Marion Gutierrez, Vangelis Metsis, Surya Nepal and Quan Z.Sheng,"IoT Middleware: A survey on Issues and Enabling Technologies", 2017 IEE Internet of Things Journal, Vol 4, NO.1.

[10] Takashi Ikebe, Hirofumi Noguchi, Naoto Hoshikawa,"Distributed Live Data Search Architecture for Resource Discovery on Internet of Things", 2016 IEEE.

[11] Pedro R. J. Pêgo, Luís Nunes,"Automatic Discovery and Classifications of IoT Devices".

[12] Antonio J. Jara, Pablo Lopez, David Fernandez, Jose F. Castillo, Miguel A. Zamora and Antonio F. Skarmeta,"Mobile Digcovery: A Global Service Discovery for the Internet of Things", 2013 27th International Conference on Advanced Information Networking and Applications Workshops.

[13] Zhiyuan Li, Rulong Chen, Lu Liu, Geyong Min," Dynamic Resource Discovery Based on Preference and Movement Pattern Similarity for Large-Scale Social Internet of Things", IEEE Internet of Things Journal, VOL. 3, NO. 4, AUGUST 2016.

[14] Farzad Khodadadi, Amir Vahid Dastjerdi, Rajkumar Buyya," Simurgh: A Framework for Effective Discovery, Programming, and Integration of Services Exposed in IoT", 2015 International Conference on Recent Advances in Internet of Things (RioT) Singapore, 7-9 April 2015.

