State Level Conference on Engineering Innovations (SCEI-17)
International Journal of Advanced Engineering Research and Science (IJAERS) *Special Issue (SCEI-17)*
https://dx.doi.org/10.22161/ijaers/scei.2017.1 *ISSN: 2349-6495(P) | 2456-1908(O)*

Instant Push Notification Mechanism

Prof. Mrs. S. A. Deshpande[1], Harshal Kanani[2], Tushar Aherkar[3], Ashutosh Hundekar[4], Ajay Bali[5]

[1]Assistant Professor, Department of Computer and IT Engineering, PVG's COET, Pune, Maharashtra, India
Email: sad_comp@pvgcoet.ac.in
[2]Student, Department of Computer and IT Engineering, PVG's COET, Pune, Maharashtra, India
Email: harshalkanani@gmail.com
[3]Student, Department of Computer and IT Engineering, PVG's COET, Pune, Maharashtra, India
Email: tsaherkar@gmail.com
[4]Student, Department of Computer and IT Engineering, PVG's COET, Pune, Maharashtra, India
Email: ashutosh.hundekar@gmail.com
[5]Student, Department of Computer and IT Engineering, PVG's COET, Pune, Maharashtra, India
Email: ajaybali9@gmail.com

Abstract**—***Sending push notification is one of the most trending marketing strategies. The notification system can be implemented using various approaches. Currently, HTTP protocol is favored for notifications. The protocols for loT and Machine to Machine (M2M) environments, namely MQTT, CoAP or LWM2M, are directly dependent on the TCP/IP protocol suite. So the reliability of this protocol suite can be used for sending notifications. To improve delivery speed and provide full tracking, we use MQTT protocol by means of a broker. We implement a MQTT based publisher subscriber model and a HTTP based notification system and experiment the usages of the two systems to prove efficiency of one over another. Bringing in the use of MQTT protocol to develop a system, which will behave like traditional system, yet be better than it, providing abstraction for the system and handling the mobility of applications, forms the basis of our proposed system.*

Keywords— API, HTTP, IoT, MQTT, TCP

I. INTRODUCTION

In 21st century, we are highly dependent on IoT devices such as smart phones given their rising popularity. To provide interactivity, instant push notification service is a must in case of mobile clients. Currently for such scenarios, HTTP protocol is preferred. Despite its reliability, there are various demer- its surrounding the system, based on mechanisms using this protocol. The service comes at the price of battery usage and bandwidth usage. HTTP protocol is based on the concept of HTTP streaming. In applications, the server sends notifications if and only if client requests for it. Consider a client server application in which a client will first send request to server for some specific task to perform. Server will hold request of the client until data is available at the server side. As there is some data available, that data will be sent to client. Client will then, again request for new data and this process goes on. In this way, client and server communicate with each other. To formulate a system in which the protocol uses lesser resources, provide higher efficiency and uses a model in which clients publish and subscribe to topics forms the basis of our idea. To do the same, we use MQTT protocol and publisher-subscriber mechanism.

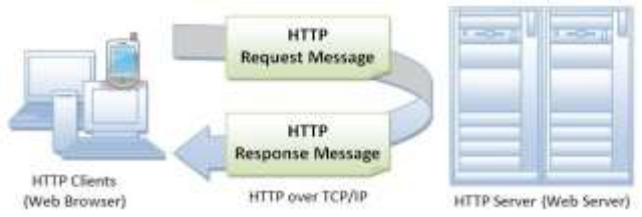

Fig. 1: HTTP-based system

II. LITERATURE SURVEY

The current notification mechanisms and services give a fair idea on how traditionally the push service is implemented.

A. Present Work

1) Pushover: Pushover makes it easy to get real-time notifications on your Android, iPhone, iPad, and Desktop. It is based on HTTP protocol and use request-response mechanism. Its mechanism depends on registering with the server and using http streaming.

2) GCM: GCM (Google Cloud Messaging) is a mobile notification service developed by Google that enables third- party application developers to send notification data or information from developer-run servers to applications that target the Google Android Operating System, as well as applications or extensions developed for the Google Chrome internet browser. Google Cloud Messaging functions using server APIs and SDKs, both maintained by Google. GCM has the ability to send push notifications, deep-linking commands, and application data using HTTP POST requests.

3) Existing Morphological System: The currently existing systems are mostly based on HTTP protocol. They use the request response paradigm. The clients have to register first. The server pushes a notification to the respective client phone (node) accordingly. The existing systems have wide range of examples like Times of India, Instagram, etc. They use the basic HTTP push notification concept. At developer scale, GCM or its blends are the favourite choice for many. Various applications such as Times of India, Yahoo Digest etc. depend on such libraries. WhatsApp, the famous chat messaging app, using XMPP blended with HTTP to send and receive messages and notify the users.

B. Proposed Work

We propose to develop a system which uses MQTT protocol with a publisher-subscriber mechanism to deliver notifications and send data across the nodes. We intend to do the same with HTTP based notifications. Ultimately, we will compare the results. System proposed will be able to allow users to get notified with the help of MQTT protocol in a similar fashion like the traditional but with increased battery life and lesser power consumption. We propose to develop abstract classes to implement publisher subscriber model based on MQTT protocol so that developers who desire to use this lightweight model in their applications do not have to study the details of underlying protocol suite.

III. MQTT PROTOCOL OVERVIEW

MQTT is a lightweight message queuing and transport protocol. It is suitable for Machine to Machine (M2M), WSN (Wireless Sensor networks) and ultimately for Internet of Things (IoT) where sensors and actors nodes communicate with each other through MQTT message broker. MQTT was initially developed by IBM and Eurotech. In 2014, it was adopted and published as an official standard by OASIS. This protocol is directly dependent on the underlying TCP suite. To avoid continuous request-response strategy used in HTTP, MQTT makes use of *long polling*. In *long polling*, even if there is no data available on server side, server will accept the connection from client and hold the connection for a period of time till keep alive and sends data over it whenever data is available with the server. MQTT provides three quality of service levels which indicates the level of assurance of a sent message. 0 ”at most once ” delivery in which message follows fire and forget policy. 1 ”at least once” delivery which is acknowledged delivery and 2 ”at most once” delivery which is assured delivery.

IV. OVERALL SYSTEM DESIGN

System architecture mainly consists of publishers, sub-scribers and a broker.

1) Publisher: A publisher is usually a node or device which is interested in sending data. Data is sent to specific other nodes or devices. The publisher will send specific message over a topic to the broker. A publisher is that component which is responsible for generating the appropriate message which can be sent and delivered to those intended.

2) Subscriber: A subscriber is usually a node or device which is interested in receiving data. Data from other nodes or devices is received by the subscribed device or node. The only condition here is that the subscriber needs to be subscribed to that particular topic on which the publisher is sending data. The subscriber gets the message after it is published by the publisher.

3) Broker: It acts a central connecting unit. It basically does the job of actually transferring data from publishers to subscribers. It does all matching, routing, storing, QoS management and filtering work. It helps all publishers or subscribers build a connection with it first, before they can start sending or receiving data.

V. SYSTEM FEATURES

A. Notification Service

The system allows devices to get connected to the broker. A device may be a publisher or a subscriber. If it is interested in sending data, it is a publisher. If it is interested in receiving data, it is subscriber. It has to subscribe to the interested topic. The service is such that it allows publisher to publish data to a certain topic made on the broker. The broker would be instrumental in actually delivering the notification from the publisher to the receiver on the subscriber side. The subscriber must have been subscribed to the topics before.

B. API(Application Programming Interface)

The application programming interface must be able to have direct connection mechanisms. It should provide bridging classes which will help developers directly connect without much hustle. It must allow direct publisher and subscriber mechanism in such a way that it behaves like a traditional notification mechanism. It should provide abstraction to the complete details of MQTT protocol so that developers need not have knowledge of implementation of the same. Developers just have to include the proposed package into their product to do function of sending notifications by using MQTT protocol. The proposed system betters the traditional system and can be as a product for products ahead.

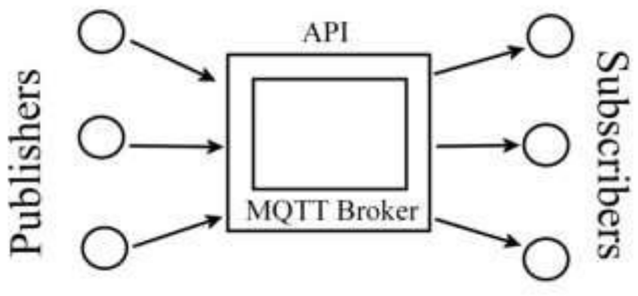

Fig. 2: MQTT-based system model

VI. BENEFITS

- Notification delivery using lesser bandwidth and power.
- Reliable.
- Scalable.
- User-friendly, as it is not different than traditional system for users.
- Developer-friendly, as it can be further used by other developers.

VII. APPLICATIONS

- It can be used in implementing a notification system for any further applications, providing benefits of lesser usage of resources for similar system.
- It can be implemented on various IoT devices, be it handheld phones or sensor nodes.
- It can be scaled to support large number of subscribers.

VIII. CONCLUSION

On the basis of the survey carried out, we observe that use of HTTP based notification system dominates the market given its reliability. We propose to implement a system based on MQTT protocol which will help to do the same task with lesser resources. It will also cover a wider variety of IoT devices, given that it is directly just dependent to TCP suite. To provide abstraction to the work to be carried out, so that other developers need not go to the exact networking basics and details, and begin work over proposed system immediately with the help of classes, forms the crux of the idea.

REFERENCES

[1] Arvind Kumar, Suchi Jobari, “Push Notification as a Business Enhancement Technique for E-commerce”, 2015 Third International Conference on Image Infonnation Processing 2

[2] Konglong Tang, Yong Wang, Hao Liu, Yanxiu Sheng, Xi Wang and Zhiqiang Wei, “Design and Implementation of Push Notification System Based on the MQTT Protocol”, International Conference on Information Science and Computer Applications (ISCA 2013), pp. 116-119,2013.