Abstract—Message queues are tools used for inter-process communication. The communication can be between softwares, services or devices on the network.

This paper covers the aspects of various standards and protocol families that can be used to implement message queues. The following protocols are being discussed here: MQTT (Message Queuing Telemetry Transport), AMQP (Advanced Message Queuing Protocol) and STOMP (Streaming Text Oriented Messaging Protocol).

This paper also describes the middlewares that implement the above mentioned protocols. The middlewares that are part of this paper are: Apache Kafka, Apache ActiveMQ, ZeroMQ, RabbitMQ.

I. INTRODUCTION

Message queues are software tools that are used for inter-process communication. [1] These tools are used to send and receive messages between one or many services or devices. Message queues are asynchronous mode of communication. As message queues are asynchronous mode of communication, the senders and receivers don't need to be active on the message queue at the same time.

Messaging Patters are protocols or methods that define how different parts of a message queue communicate with each other. It defines the kind of relations or roles, services or devices could play in a messaging system. Messaging patterns can be divided into three major types:

- Client / Server
- Publish / Subscribe
- Push / Pull

II. MESSAGING PATTERNS

A. Client / Server

Client Server mode of messaging pattern is when a lot of clients i.e. devices need to be connected to a single server and request data from the server. In this kind of messaging pattern, the server has the possibility of letting a lot of clients connect to it. The clients then request data from the server and the server sends the data in return.

Insert image for Client Server here.

B. Publish / Subscribe

A publish subscribe messaging pattern is similar to a notice board or a RSS feed kind of implementation. Multiple devices can subscribe to channels or topics and one of those devices can publish data to the topic or channel. This is a centralised kind of messaging pattern as all the devices need to be connected to a single device that act as a broker.

Insert image for Publish subscribe.

III. STANDARDS AND PROTOCOL FAMILIES

A. MQTT

MQTT (Message Queuing Telemetry Transport) [2] is a publish subscribe message queueing protocol. It was approved as an ISO standard in 2015.

B. AMQP

AMQP (Advanced Message Queuing Protocol) [3] also implements a topic based publish subscribe system. It has a range of features that it implements.

C. STOMP

STOMP (Streaming Text Oriented Messaging Protocol) [4] is a text based protocol. It is a very simple protocol and hence can be connected from any simple client, example a telnet client.

IV. MESSAGING MIDDLEWARE

A. Apache Kafka

Kafka [5]
- Pub/Sub

B. Apache ActiveMQ

ActiveMQ [6]
- MQTT
- AMQP
- STOMP

C. ZeroMQ

ZeroMQ [7]
- Pub/Sub
- Client/Server
- Push/Pull
- PAIR

D. RabbitMQ

RabbitMQ [8]
- AMQP
- MQTT
- STOMP
V. Others

Other papers that I would be referring to are:

[9]

REFERENCES


