MQTT is a very light weight protocol designed for IoT which supports lots of features with consistent accuracy. It might be a question why we need MQTT protocol when HTTP is still exists and working fine. The clarification of that question is that HTTP is great for doing request and response like a client may ask for information to a server and server will response according to that request but it doesn't really have a good solution when a source of information should push a change to many clients and there is no built in support for quality of service. The text based format of HTTP requires more bandwidth and any device that act as host must have a server installed and keep the server live to answer incoming request consumes a lot of battery life but MQTT solves many of the problems of HTTP. First of all it introduce a publish and subscribe messaging pattern which means that any source of information such as a sensor can publish its data and then any client can subscribe that data. All these is happening in a broker like keep track of all subscription and publications so when a publisher sense and update with new data it publishes a message and the broker takes care of sending the new data to all subscriber. The built in support for quality of services meaning that the broker can guarantee the delivery of the message e.g. that is delivered at least one time or exactly once. MQTT is a binary format that requires a minimum bandwidth. The fixed header is actually only 2 bytes and it also has a small implementation footprint that requires less battery. So it may said that MQTT can handle smaller devices in low bandwidth because MQTT control packet headers are kept as small as possible. Each MQTT control packet consist of three parts, a fixed header, variable header and payload. MQTT has a small header overload that makes it appropriate for IoT by lowering the amount of data transmitted over week network .Both of these efficient aspects make MQTT scale very well and it is very important when we handle a million of devices.

1.3 MQTT Functionality

So as explained MQTT is a connectivity protocol designed for M2M. It is an extremely lightweight publish/subscribe messaging transport that is ideal for connecting small devices connected on networks with minimal bandwidth. Now I want to explain how actually it works. Below is a simple figure of how it works. So it follows publish/subscribe messaging protocol which allows a message to be published once and multiple subscribers to receive the message providing decoupling between the publisher and subscriber. A publisher publish a message (Publication) in the Topics.

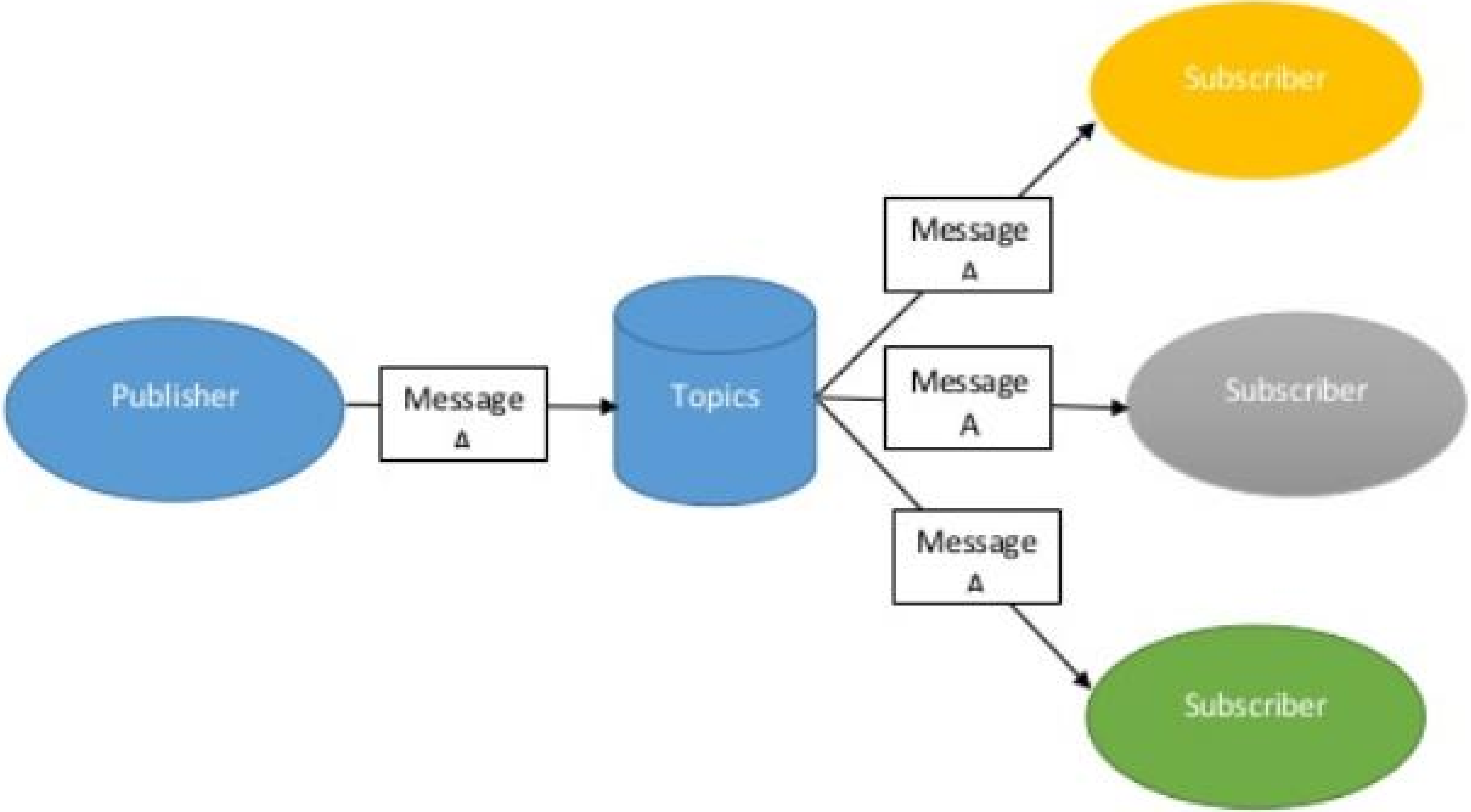

Figure: Simple MQTT functions

The subscribers subscribe to the Topics for message (publications). A message server or Broker matches publications to subscriptions. If it found no matches the message will be discarded. If one or more matches the message the broker will deliver the message to each subscriber. Thus it works. Now come to the technical mechanism. The MQTT messages are delivered asynchronously ("PUSH") through publish subscribe architecture. It works by exchanging some set of control packets where each control packets has a specific functions. The MQTT topology has a MQTT server and MQTT client and they are communicate by those control packets. Every steps of communication has a verification e.g. CONNECT packet is responsible to send the request to the server informing that client wants to connect and CONNACK packet is generated by the server to acknowledge the client about the connections. Similarly PUBLISH and PUBACK packet work like same process.

1.5 Goals

MQTT (Message Queuing Telemetry Transport) is one of the protocols supported by the IBM Message Broker products as a communicating data to and from the Broker. The protocol was designed specifically for remote telemetry applications, with three specific design goals:

1. The protocol should be as lightweight as possible across the "wire" (or other communication medium) most remote telemetry is done over low bandwidth, high cost networks, and so minimizing the overhead of each message is highly desirable.
2. The protocol should be very easy to implement on embedded devices such as sensors and gateways.
3. Can handle lots of devices whether it is a lower configured device or higher and with a minimum bandwidth.
4. This protocol maintains consistency to lower latency.

1.6 Conclusion

MQTT provides a lots of functions for the Internet of Things. It can help providing a great performance and create new area for messaging and can handle billion of things connected through the internet. It is a very light weight protocol that can work with every types of devices and work using a minimum bandwidth. Now-a-days facebook.com is using MQTT protocol for their messenger which working great in our messaging in social network.