

Introduction to Internet of Things
Prof. Sudip Misra
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 06
Basics of IoT Networking-Part-II

So, we are now going to continue our discussions on the basic issues, basic aspects of the networking in internet of things. So, we have already seen the different fundamental issues that are out there. Now, we are going to look at the different protocols that are there that can be used for something different purposes in IoT.

(Refer Slide Time: 00:44)

Functionality-based IoT Protocol Organization

- ✓ **Connectivity** (6LoWPAN, RPL)
- ✓ **Identification** (EPC, uCode, IPv6, URIs)
- ✓ **Communication / Transport** (WiFi, Bluetooth, LPWAN)
- ✓ **Discovery** (Physical Web, mDNS, DNS-SD)
- ✓ **Data Protocols** (MQTT, CoAP, AMQP, Websocket, Node)
- ✓ **Device Management** (TR-069, OMA-DM)
- ✓ **Semantic** (JSON-LD, Web Thing Model)
- ✓ **Multi-layer Frameworks** (Alljoyn, IoTivity, Weave, Homekit)

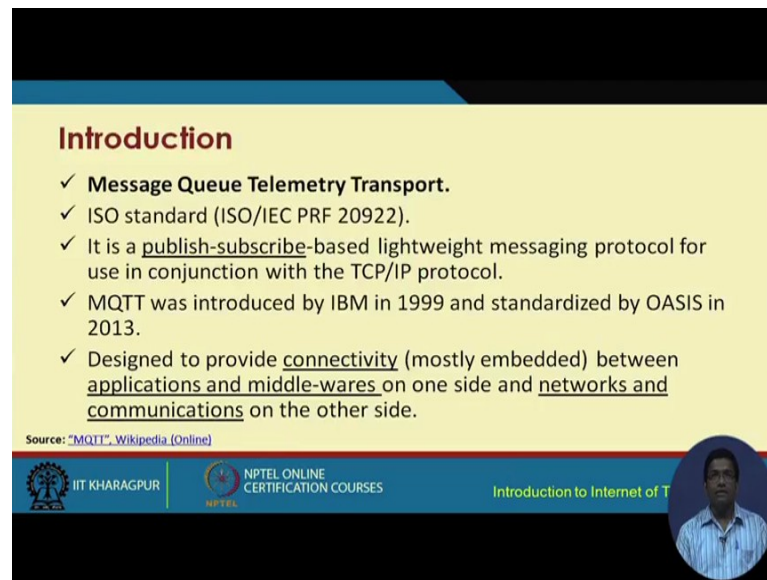
Source: Internet of Things Protocols (Online)

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things

Now, based on the different functionalities, there are actually large numbers of protocols that are proposed for use in IoT. So, based on the functionality, these protocols are classified in this in this manner. So, this is just a classification that has been shown, but this is not unique classification in anyway and it should be constituted in that manner. So, this is just a classification attempt under different categories that is shown and here as we can see the different protocols for something.

These classifiers you know a different issues are also mentioned in brackets like this. So, it is basically not possible and also not required to go through all these different protocols. So, we have selected only a few protocols from this you know red coloured category and this is what we are going to discuss in this particular course.

(Refer Slide Time: 01:55)




Introduction

- ✓ **Message Queue Telemetry Transport.**
- ✓ ISO standard (ISO/IEC PRF 20922).
- ✓ It is a publish-subscribe-based lightweight messaging protocol for use in conjunction with the TCP/IP protocol.
- ✓ MQTT was introduced by IBM in 1999 and standardized by OASIS in 2013.
- ✓ Designed to provide connectivity (mostly embedded) between applications and middle-wares on one side and networks and communications on the other side.

Source: "MQTT", Wikipedia (Online)

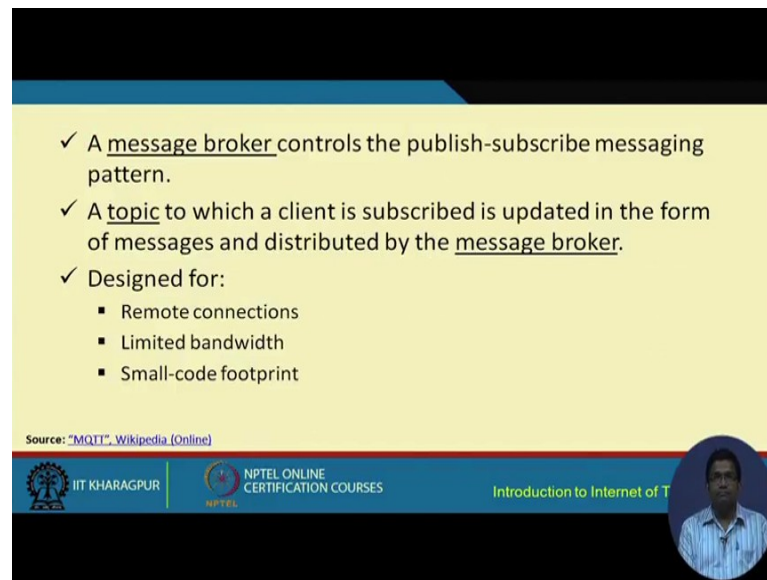
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of T



So, we will start with the MQTT protocol first and this MQTT, the full form of this is Message Queue Telemetry Transport. So, it is an ISO standard which is based on publish subscribe model. So, basically you know what happens is there is some kind of publishing of the data and then, the fetching of the data by the subscribers. So, this is how this publish subscribe model works and MQTT basically what it has done is, this publish subscribe model, it has been made lightweight through the use of this protocol, so that this lightweight protocol can be used in conjunction with the TCP IP protocol suit. This is what MQTT supports.

MQTT going back to the history was introduced in 1999 by IBM and is standardized in 2013 by Oasis. This is a standardization organization oasis. So, it has standardized in the year 2013. So, this particular protocol does couple of things. One is offering connectivity between different embedded devices between the applications and then, middle ware of one device and network and communication on the other side of the device. So, we have connectivity between applications and middle ware for one side and the networks and communication on the other. This is what MQTT does.

(Refer Slide Time: 03:32)



✓ A message broker controls the publish-subscribe messaging pattern.


✓ A topic to which a client is subscribed is updated in the form of messages and distributed by the message broker.

✓ Designed for:

- Remote connections
- Limited bandwidth
- Small-code footprint

Source: "MQTT", Wikipedia (Online)

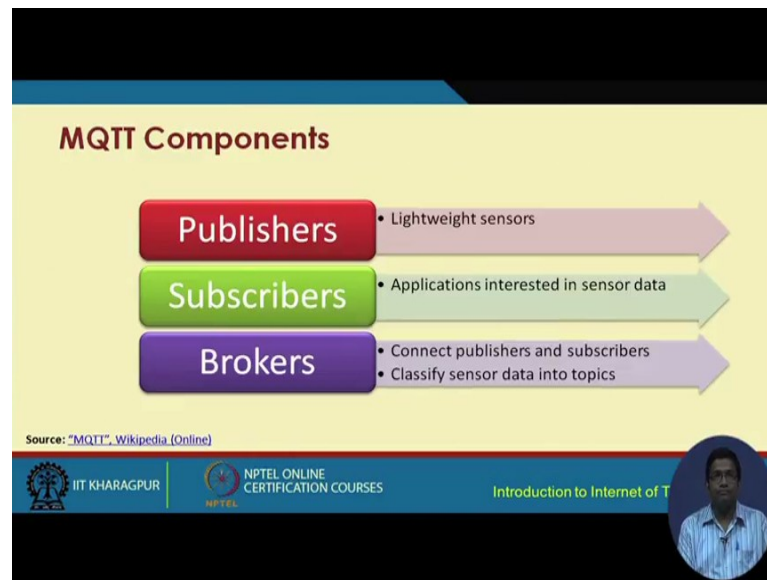
IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Introduction to Internet of T



So, in MQTT there are three concepts that are involved. The first we are going to go through is the concept of a message broker and first, we are going to go through some of these concepts and there after I am going to show you pictorially how MQTT functions. So, we have a message broker; the concept of a message broker that basically serves like a broker which takes control of publishing of the messages and subscription of the messages. So, publish subscribe is basically controlled by the message broker, number 1. Number 2 is there is a concept of topic and this is what the client is subscribed and based on the updates. The data are sent to the clients by the message broker, this data are distributed by the message broker to the clients who have subscribed to the services.

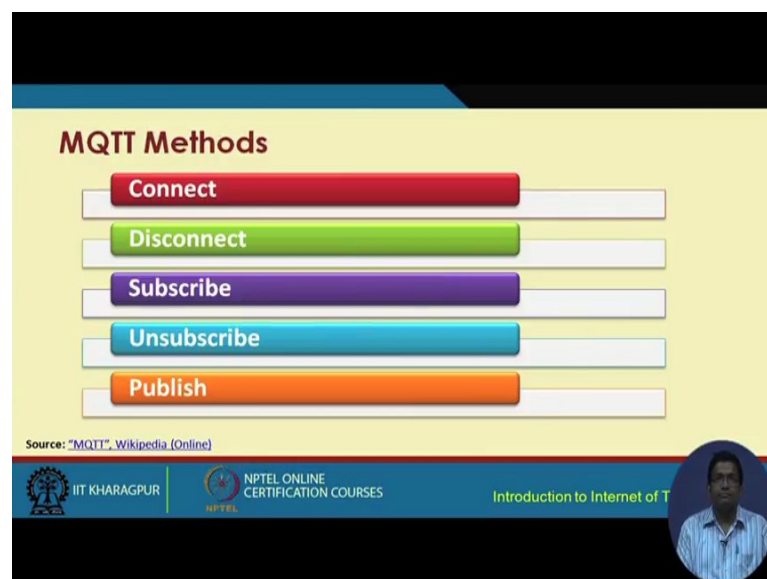
So, this is design for remote connections limited bandwidth environments and MQTT, basically the advantage is that it provides every small code foot print. So, basically you know by writing only a small piece of code, one would be able to achieve all these different functions that I have just mentioned.

(Refer Slide Time: 04:48)



The different components of MQTT are as follows. We have three principle components. The publishers which involve the different sensors, the subscribers and that means, those entities, those applications, those units that are interested in the data that is published by the sensors. Number 3 is the broker in between which helps the publishers and the subscribers connect to one another and also help in classifying the sensor data into different topics.

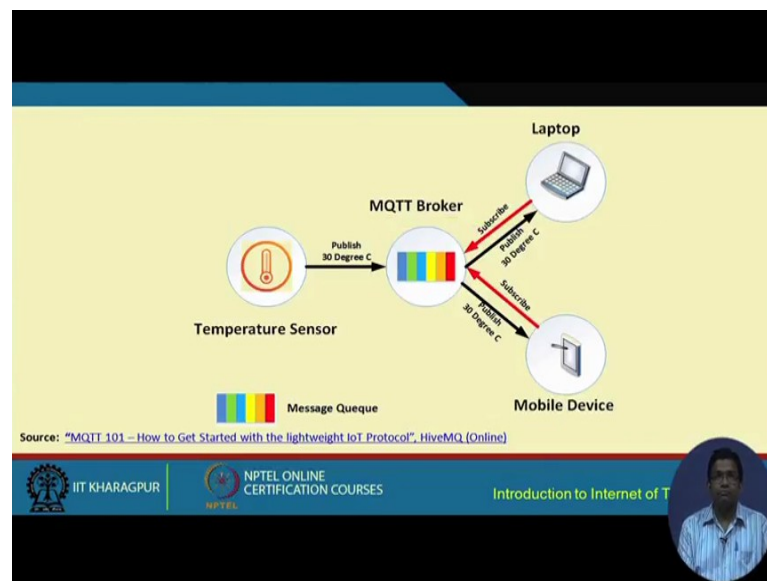
(Refer Slide Time: 05:29)



In MQTT there are a few different methods. One is connect, the second is disconnect, subscribe, unsubscribe and publish. So, basically the connect method helps to connect with the server, helps to connect this device with the server. Then, disconnect is the opposite. Whenever it is no longer required to be to remain connected, the disconnect method helps in disconnecting from the server from TCP IP service offerings and so on. And then comes the subscribe which is basically subscribing to the services and unsubscribe is the opposite that whenever it is no longer required to continue with getting the different data offerings, the data services and so on.

The unsubscribe method can be executed and then, we have the publish method which is basically publishing data for maybe you know publishing the data from these different sensors or these different devices to the broker for it to be fetched by the different application clients.

(Refer Slide Time: 06:49)

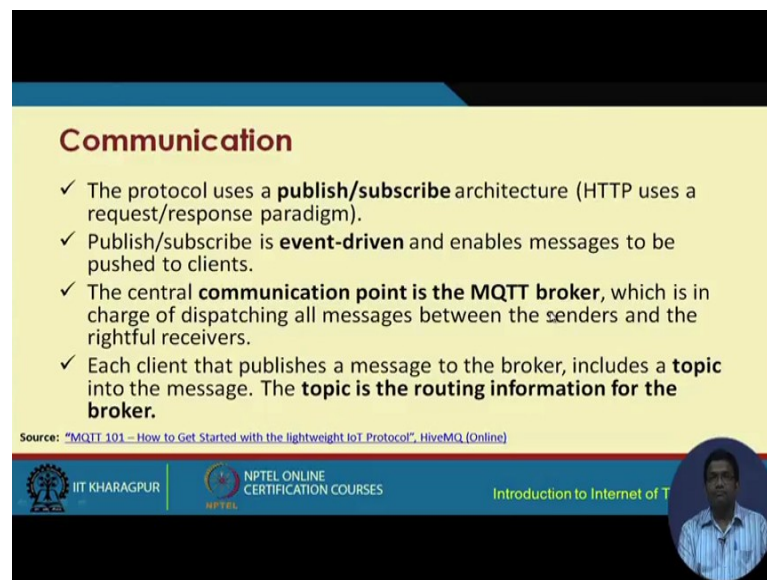


So, this is what I was referring to just a short while back. So, what we have is these pictorial depictions of how these publish subscribe model works in the case of MQTT. So, we have a temperature sensor, we have laptops, we have mobile devices. This temperature sensor in this example publishes the temperature which is basically brokered at this MQTT broker. So, what this fellow does is, it broker you know because it is a broker between the different clients and different other application serving devices which

required the data which can subscribe to the data that is published by these different sensors.

So, what it is going to do is, first it is going to get the subscription requests from these different clients, the mobile device, the laptop and so on and so forth. So, first is the subscribed and there after it is published. So, what is published? The temperature is published. Where is the temperature residing? It is fetched from the sensor and it is being brokered at the MQTT broker. So, from those devices, those clients which have basically subscribed to the services, they are going to get this sensing updates, the sensor data updates.

(Refer Slide Time: 08:27)




Communication

- ✓ The protocol uses a **publish/subscribe** architecture (HTTP uses a request/response paradigm).
- ✓ Publish/subscribe is **event-driven** and enables messages to be pushed to clients.
- ✓ The central **communication point is the MQTT broker**, which is in charge of dispatching all messages between the senders and the rightful receivers.
- ✓ Each client that publishes a message to the broker, includes a **topic** into the message. The **topic is the routing information for the broker**.

Source: "MQTT 101 – How to Get Started with the lightweight IoT Protocol", HiveMQ (Online)

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things

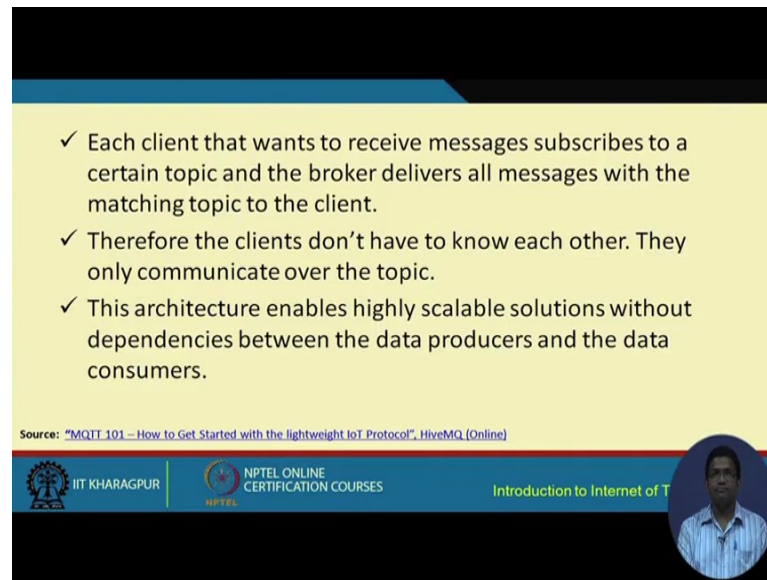


So, when we look at the communication, the architecture that is followed is publish subscribe publish subscribe architecture and not the request response architecture that is typically followed by traditional http which is used for the internet, this publish subscribe model is event driven. That means, whenever there is an event, whenever there is a fire, when the temperature increases, whenever a camera may be an IoT camera basically observes some kind of change in the environment or whatever maybe there is an intruder or whatever it is, so the central communication point.

So, it is sorry, I am sorry. So, what we have is it is event driven. So, whenever there is some kind of an event and those data are basically pushed to the clients, so these messages basically are pushed to the clients and then, we have this broker which is in-

charge of dispatching all the messages between the senders and receivers and each client that publishes a message to the broker includes a topic into the message and this is the topic which is very much important in the case of MQTT and this is the topic which is of interest to these application clients.

(Refer Slide Time: 10:02)




✓ Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client.

✓ Therefore the clients don't have to know each other. They only communicate over the topic.

✓ This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers.

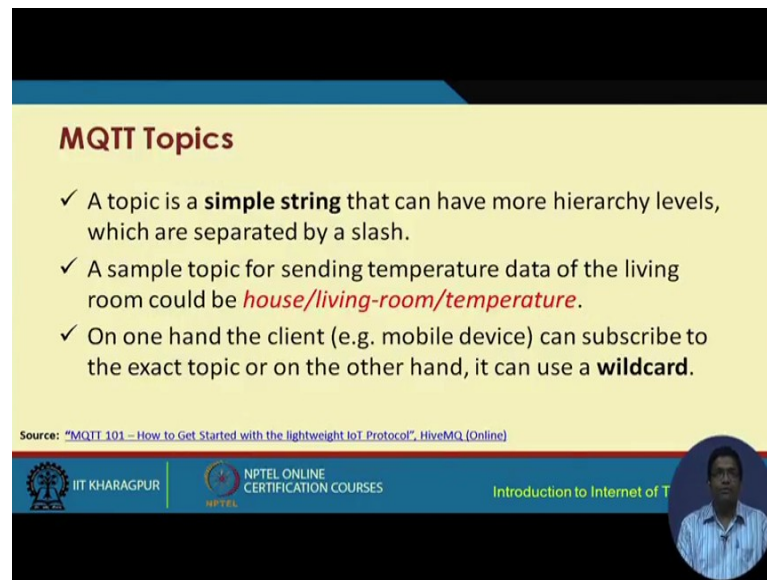
Source: "MQTT 101 – How to Get Started with the lightweight IoT Protocol", HiveMQ (Online)

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things



So, each client that wants to receive the messages subscribes to a certain topic and the broker delivers all the messages with in the matching topic to the client. So, essentially what is happening is the clients, they do not have to know each other and what is required is they only need to communicate with each other over the topic. So, they do not know about each other and this architecture basically appears to be a scalable architecture with a scalable solution. There is not much dependency between the producers and the consumers of the data and that is where MQTT is very popular.

(Refer Slide Time: 10:43)


A presentation slide titled "MQTT Topics" with a yellow background. It contains three bullet points explaining MQTT topics. The slide also includes a source reference, logos for IIT Kharagpur and NPTEL, and a small video inset of a speaker.

MQTT Topics

- ✓ A topic is a **simple string** that can have more hierarchy levels, which are separated by a slash.
- ✓ A sample topic for sending temperature data of the living room could be *house/living-room/temperature*.
- ✓ On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a **wildcard**.

Source: "MQTT 101 – How to Get Started with the lightweight IoT Protocol", HiveMQ (Online)

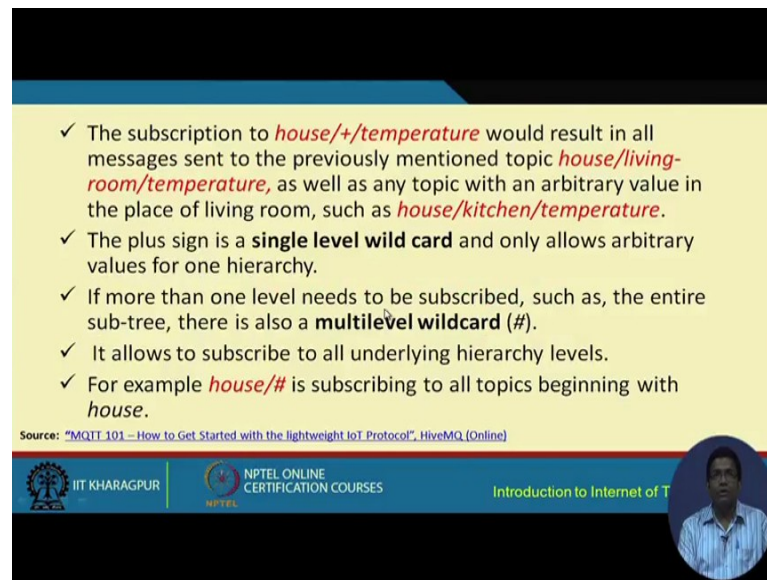
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things



In MQTT we have the concept of the topic as I just mentioned a minute back and this topic is nothing, but a simple string that can have more hierarchical levels which are separated by a slash. In this manner a sample topic for sending the temperature data of the living room could be marked in this manner, could be named in this manner. So, we have house, a smart house kind of scenario. We have house and within the house we have living room and the temperature that is collected from this particular living room is of interest. So, this topic that is of interest and this is a sample, this is an example of how the topic for sending temperature data of a living room looks like.

So, on one hand the client like a mobile device, a laptop or whatever can subscribe to the exact topic or on the other hand, it can also use a wildcard. So, this is an example of an exact topic. Wildcard could also be used.

(Refer Slide Time: 12:01)



✓ The subscription to *house/+temperature* would result in all messages sent to the previously mentioned topic *house/living-room/temperature*, as well as any topic with an arbitrary value in the place of living room, such as *house/kitchen/temperature*.

✓ The plus sign is a **single level wild card** and only allows arbitrary values for one hierarchy.


✓ If more than one level needs to be subscribed, such as, the entire sub-tree, there is also a **multilevel wildcard (#)**.

✓ It allows to subscribe to all underlying hierarchy levels.

✓ For example *house/#* is subscribing to all topics beginning with *house*.

Source: "MQTT 101 – How to Get Started with the lightweight IoT Protocol", HiveMQ (Online)

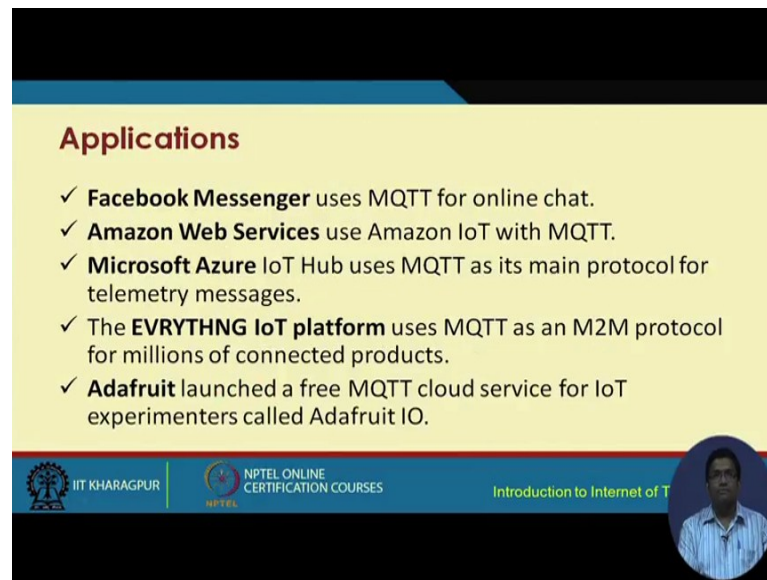
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of T



Wildcards can be used in different ways based on the different levels the subscription to *house/+temperature* results in all messages sent to the previously mentioned topic, *house/living room/temperature* as well as any topic with an arbitrary value in the place of a living room such as *house/kitchen/temperature*.

So, basically you know this plus sign is a wildcard character which only allows arbitrary values for one hierarchy. If more than one hierarchical level is required, the multilevel wildcard is used. So, this is single level wildcard. The plus one is a single level wildcard and this hash sign is as multilevel wide wildcard. It allows to subscribe to all underlying hierarchical levels. For example, *house/#* is for subscribing to all topics beginning with *house*.


(Refer Slide Time: 13:06)



Applications

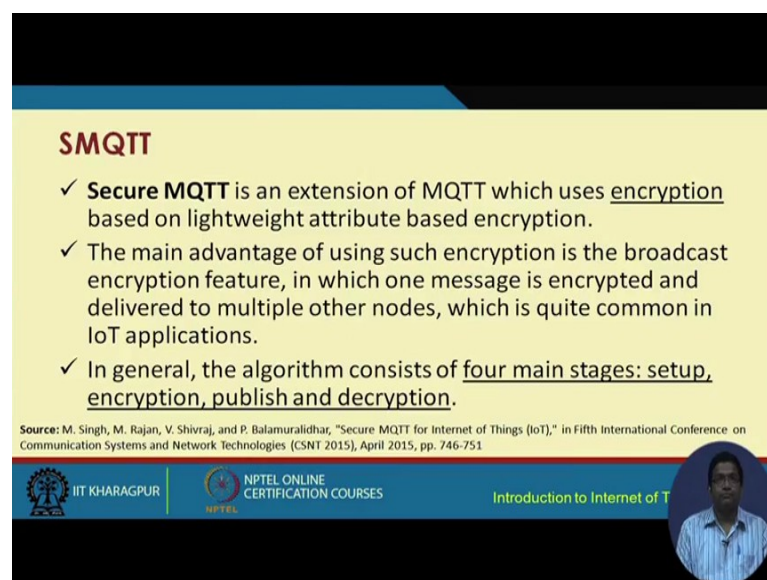
- ✓ **Facebook Messenger** uses MQTT for online chat.
- ✓ **Amazon Web Services** use Amazon IoT with MQTT.
- ✓ **Microsoft Azure** IoT Hub uses MQTT as its main protocol for telemetry messages.
- ✓ The **EVERYTHING IoT platform** uses MQTT as an M2M protocol for millions of connected products.
- ✓ **Adafruit** launched a free MQTT cloud service for IoT experimenters called Adafruit IO.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things



Different applications that use MQTT, Facebook messenger for online chat, Amazon website service use Amazon IoT with MQTT, Microsoft Azure IoT hub uses MQTT as its main protocol. For telemetry messages, the EVERYTHING IoT platform uses MQTT as an M2M protocol for connecting several products and devices. Adafruit uses MT MQTT cloud service for IoT experimenter's caller Adafruit IO.

(Refer Slide Time: 13:45)




SMQTT

- ✓ **Secure MQTT** is an extension of MQTT which uses encryption based on lightweight attribute based encryption.
- ✓ The main advantage of using such encryption is the broadcast encryption feature, in which one message is encrypted and delivered to multiple other nodes, which is quite common in IoT applications.
- ✓ In general, the algorithm consists of four main stages: setup, encryption, publish and decryption.

Source: M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," in Fifth International Conference on Communication Systems and Network Technologies (CSNT 2015), April 2015, pp. 746-751

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things

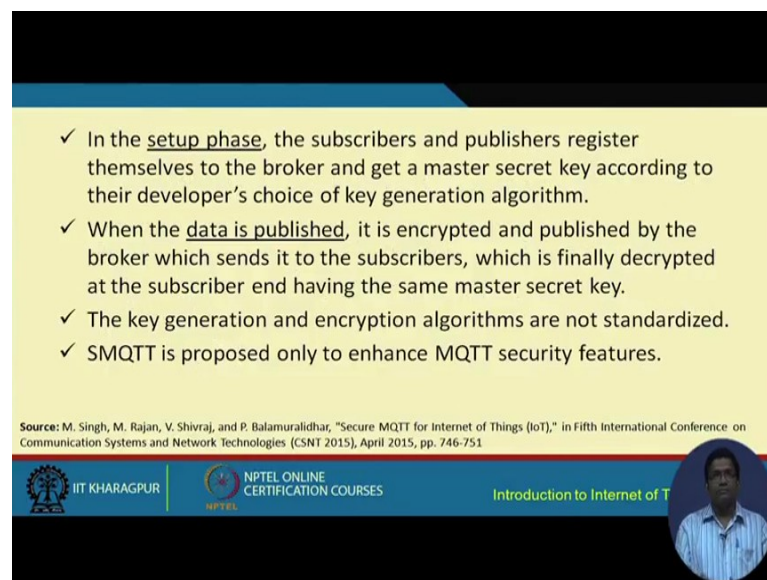


Finally after understanding the overall philosophy behind MQTT, let us now quickly review the secured version of MQTT which is called the secure MTTM MQTT, the

secure MQTT or SMQTT in short. So, this is known in both these ways and this actually to me is quite similar in notion to http and https, the secure http. So, we have http secure http, we have MQTT secure MQTT.

So, secure MQTT is an extension of MQTT. So, basically it is an extension of the MQTT that we just discussed by using different security features such as encryption and so on. The advantage of such encryption is the broadcast encryption feature in which one message is encrypted and delivered to multiple other nodes which is quite common in IoT applications. In general, the algorithm consists of four main stages i.e. the setup stage, the encryption stage, the publish stage and the decryption stage. So, setup encryption publish and decryption.

(Refer Slide Time: 15:03)



✓ In the setup phase, the subscribers and publishers register themselves to the broker and get a master secret key according to their developer's choice of key generation algorithm.


✓ When the data is published, it is encrypted and published by the broker which sends it to the subscribers, which is finally decrypted at the subscriber end having the same master secret key.

✓ The key generation and encryption algorithms are not standardized.

✓ SMQTT is proposed only to enhance MQTT security features.

Source: M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," in Fifth International Conference on Communication Systems and Network Technologies (CSNT 2015), April 2015, pp. 746-751

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of T



In setup phase, the subscribers and publishers register themselves to the broker and get a master secret key according to the developer's choice of key generation algorithm. So, depending on what key generation algorithm is used, the subscribers and the publishers register themselves to the broker and get a master secret key according to that particular algorithm.

So, when the data is published, it is encrypted and published by the broker which sends it to the subscribers which is finally decrypted at the subscriber end having the same master secret key. The key generation and encryption algorithms are not standardized SMQ. SMQTT is proposed only to enhance the security aspects of MQTT.

So, in this part of the lecture on basic topics on IoT networks, what we have done is primarily we have gone through two protocols. We have seen that there is an assortment of different protocols that they have to support the networking of IoT, but in this particular part of the lecture, we have focussed mostly on two protocols. One is the SMTT and the other one is, sorry MQTT and the other one is SMQTT which is basically the secure version of MQTT.

So, we are going to go through few other protocols in the next lecture and from that we can use them finally to establish to build different platforms for IoT in a small scale or even in the larger scale.

Thank you.